

As 10 Áreas da Engenharia de Software, Conforme o SWEBOK

Jorge H C Fernandes
2004

Indicações de que a engenharia de software está se tornando uma atividade de engenharia (SWEBOK)

- Cursos de graduação em universidades dos EUA, Reino Unido, Austrália, Canadá e Brasil
- ISO e CMM/CMMI usados para certificar empresas de desenvolvimento de software
- Código de ética
- **Registro e Certificação da profissão!?**

Características de uma Engenharia (SWEBOK, 1999)

- Educação de iniciação profissional validada e legitimada pela sociedade
- Registro da adequação à prática através de certificação voluntária ou licenciamento compulsório
- Desenvolvimento de habilidades especializadas e educação profissional continuada
- Suporte através de sociedades profissionais
- Compromisso a normas de conduta freqüentemente estabelecidas em um código de ética

Origens do Corpo de Conhecimentos da Engenharia de Software

- Matemática
- Ciência da Computação
- Administração de Projetos
- Ciências Cognitivas e Fatores Humanos
- Engenharia de Computadores
- Administração e Ciências Administrativas
- Engenharia de Sistemas

Conceitos Fundamentais de Engenharia de Software

- Abstração
- Métodos e Notações de Análise e Design
- Prototipação de Interface com Usuário
- Modularidade e Arquitetura
- Ciclo de Vida e Processo de Software
- Reuso
- Métricas
- Suporte Automatizado

10 Áreas da Engenharia de Software

Gerência de Configuração de Software

Identifica a configuração do sistema (características documentadas do hardware e software que o compõem) em pontos discretos no tempo, de modo a controlar sistematicamente suas mudanças e manter sua integridade e rastreabilidade durante o ciclo de vida do sistema.

Gerência de Engenharia de Software

Gerencia projetos de desenvolvimento de software.

Processo de Engenharia de Software

Define, implementa, mede, gerencia, modifica e aperfeiçoa o processo de desenvolvimento de software

Ferramentas e Métodos

Ferramentas de software automatizam o processo de engenharia de software

Métodos impõem estrutura sobre a atividade de desenvolvimento e manutenção de software com o objetivo de torna-la sistemática e mais propensa ao sucesso.

Qualidade de Software

Conjunto de atividades relacionadas com garantia de qualidade de software, entre estas as atividades de verificação e validação.

Requisitos de software

Aquisição, análise, especificação e gestão de requisitos de software.

Design de software

Transformação de requisitos (de software), tipicamente estabelecidos em termos relevantes ao domínio do problema, em uma descrição explicando como solucionar os aspectos do problema relacionados com software

Construção de Software

Construção de programas funcionais e coerentes através da codificação, auto-validação, e teste unitário.

Teste de Software

Verificação dinâmica do comportamento do programa através do uso de um conjunto finito de casos de teste - adequadamente selecionados de um domínio de execuções usualmente infinito - contra o comportamento esperado deste

Manutenção de Software

Atividades de suporte custo-efetivo a um sistema de software, que pode ocorrer antes e após a entrega do software.

Após a entrega do software são feitas modificações com o objetivo de corrigir falhas, melhorar seu desempenho ou adapta-lo a um ambiente modificado.

Antes da entrega do software são feitas atividades de planejamento.

Requisitos de Software

- Aquisição, análise, especificação e gestão de requisitos de software
- Dividido em seis tarefas interativas
 - Engenharia de Requisitos
 - Elicitação (captura, descoberta, aquisição) de requisitos
 - Análise de Requisitos
 - detecção e resolução de conflitos, descoberta dos limites e interações do sistema com o ambiente (mapeamento dos requisitos do sistema para requisitos do software)
 - Especificação de requisitos (estrutura, qualidade e verificação do documento de requisitos)
 - Validação de requisitos (verificação de omissões, conflitos e ambiguidades) + adequação às normas de qualidade.
 - Gestão de requisitos (gestão de mudanças, manutenção da consistência com as fases posteriores)

Design de Software

- Transforma requisitos em uma descrição de solução sob o ponto de vista do software
- Decomposição e organização em componentes, com descrição de interfaces entre estes componentes
- Refina a descrição em nível de detalhe adequado à construção

Sub-áreas do Design de Software

- Arquitetura de software
 - estruturas e pontos de vista
 - estilos arquiteturais
 - padrões de design
 - famílias de programas e frameworks
 - Análise e avaliação de qualidade de design
 - atributos de qualidade, análise de qualidade
 - ferramentas de avaliação
 - métricas
- Notações de design (estruturais e comportamentais)
- Estratégias e métodos de design
 - estratégias gerais
 - design orientado a funções
 - design orientado a objetos
 - design centrado em estruturas de dados
 - outros métodos

Descrever Estruturas de uma aplicação

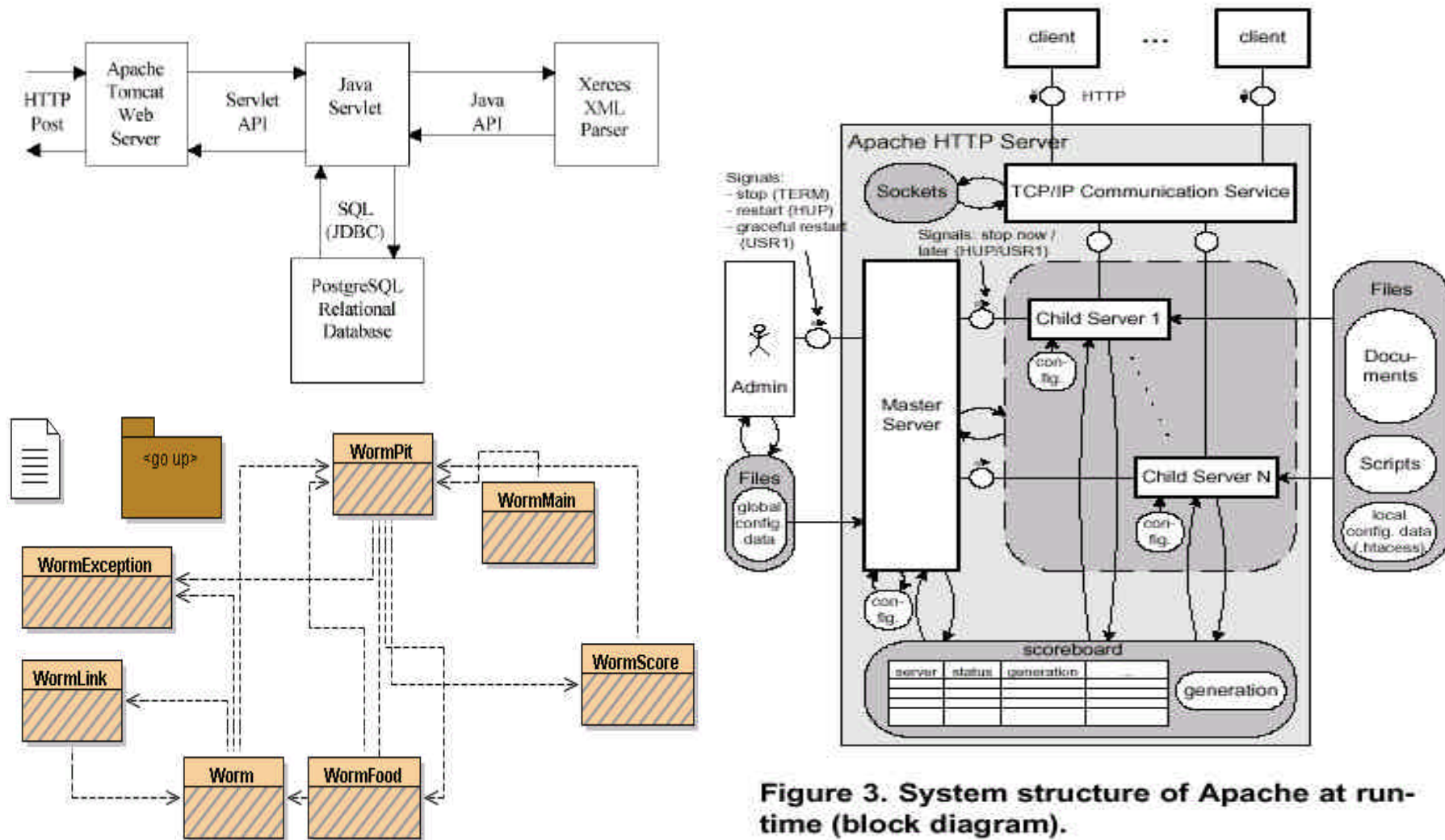


Figure 3. System structure of Apache at run-time (block diagram).

Descrever Comportamentos...

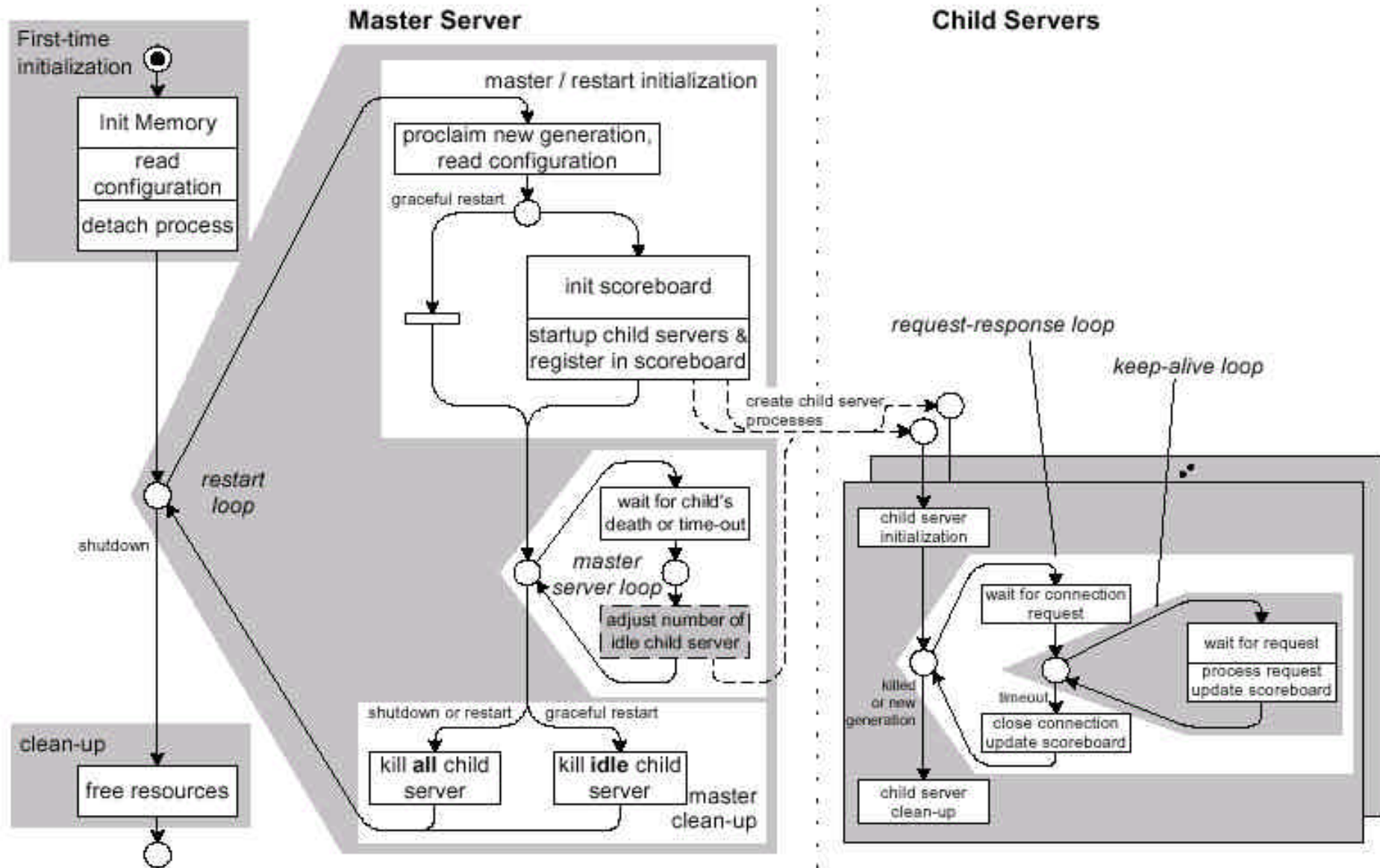


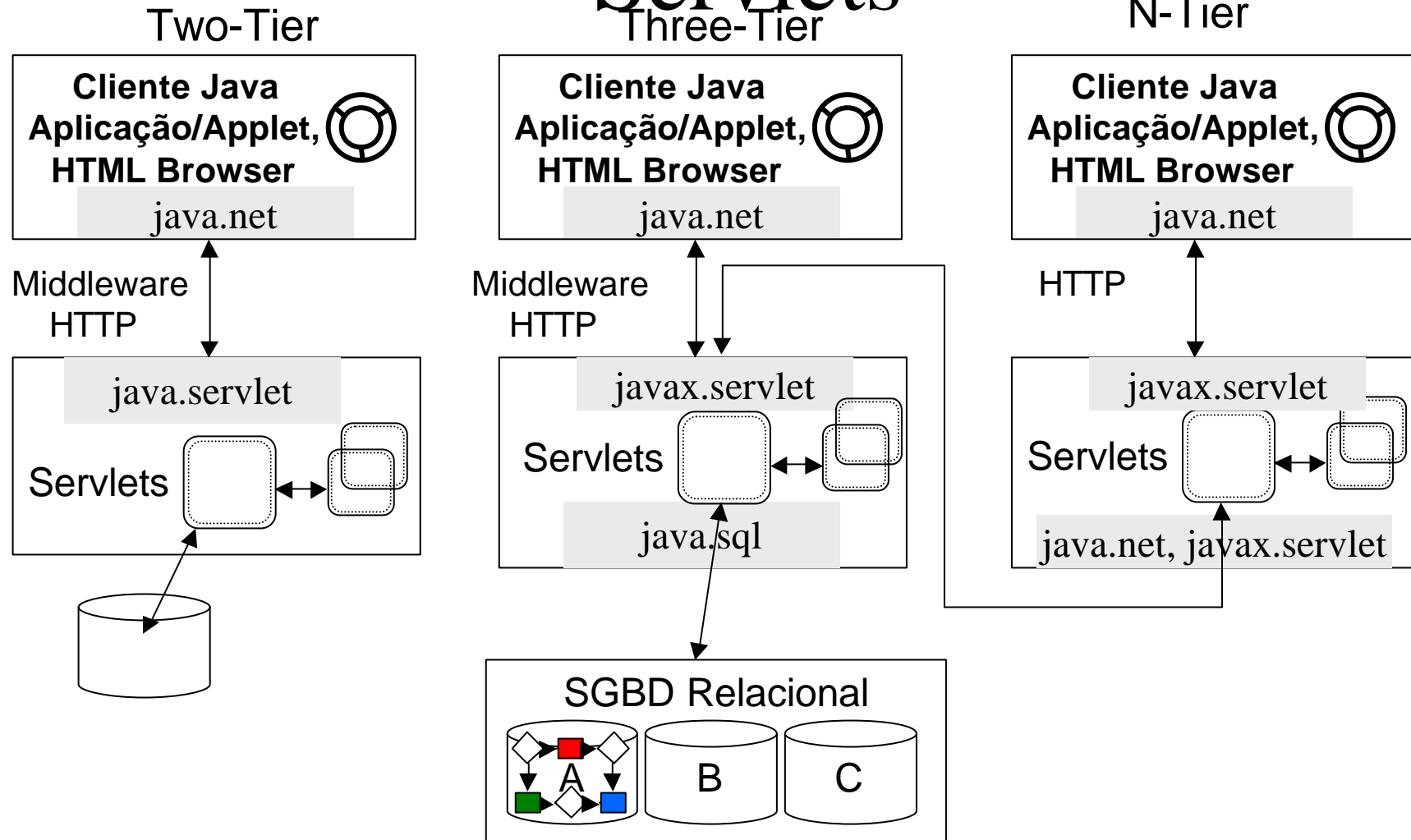
Figure 4. Overall behavior of the Apache HTTP server (petri net).

Edições Java 2 e Seus Dispositivos

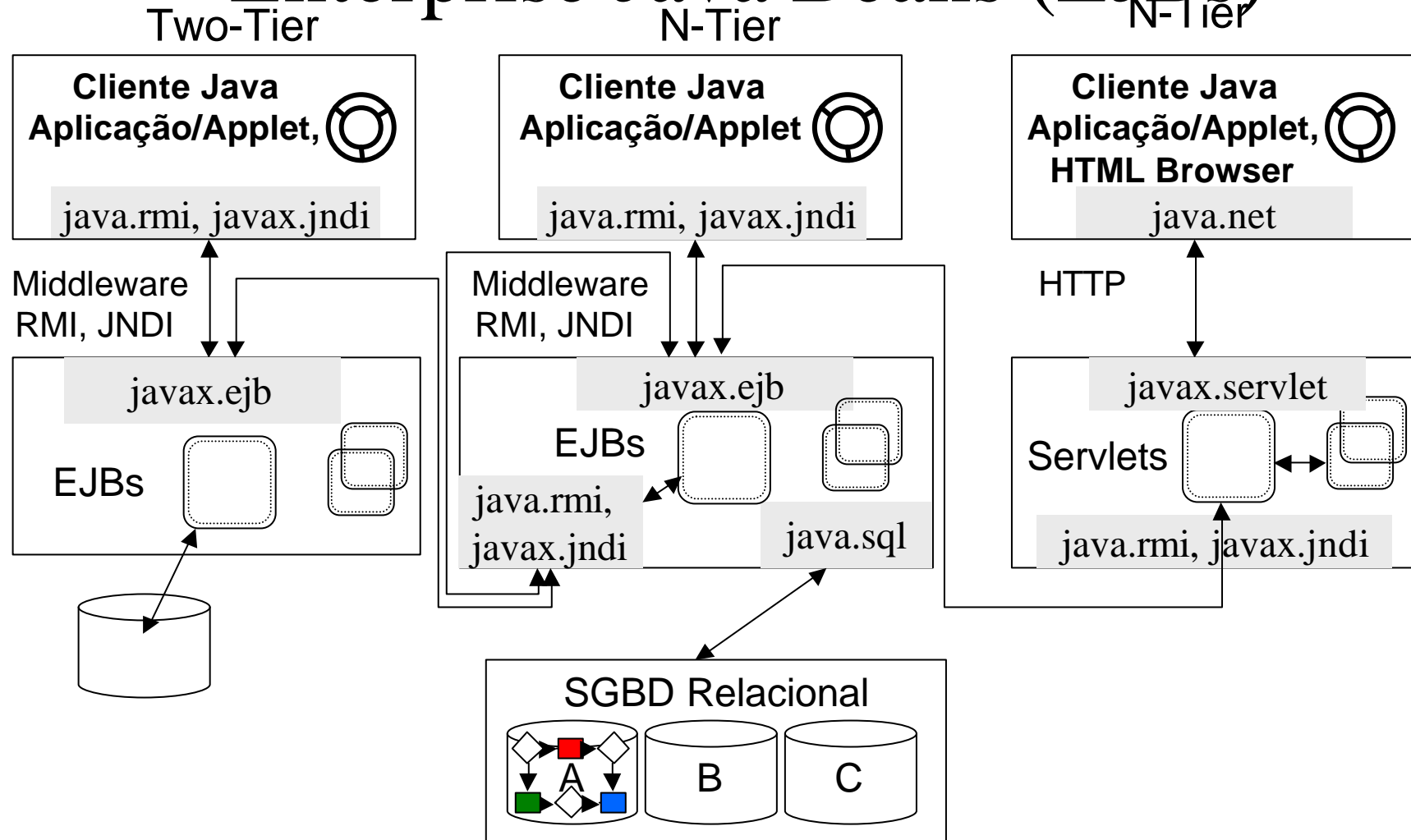


Arquiteturas Multi-Camadas com

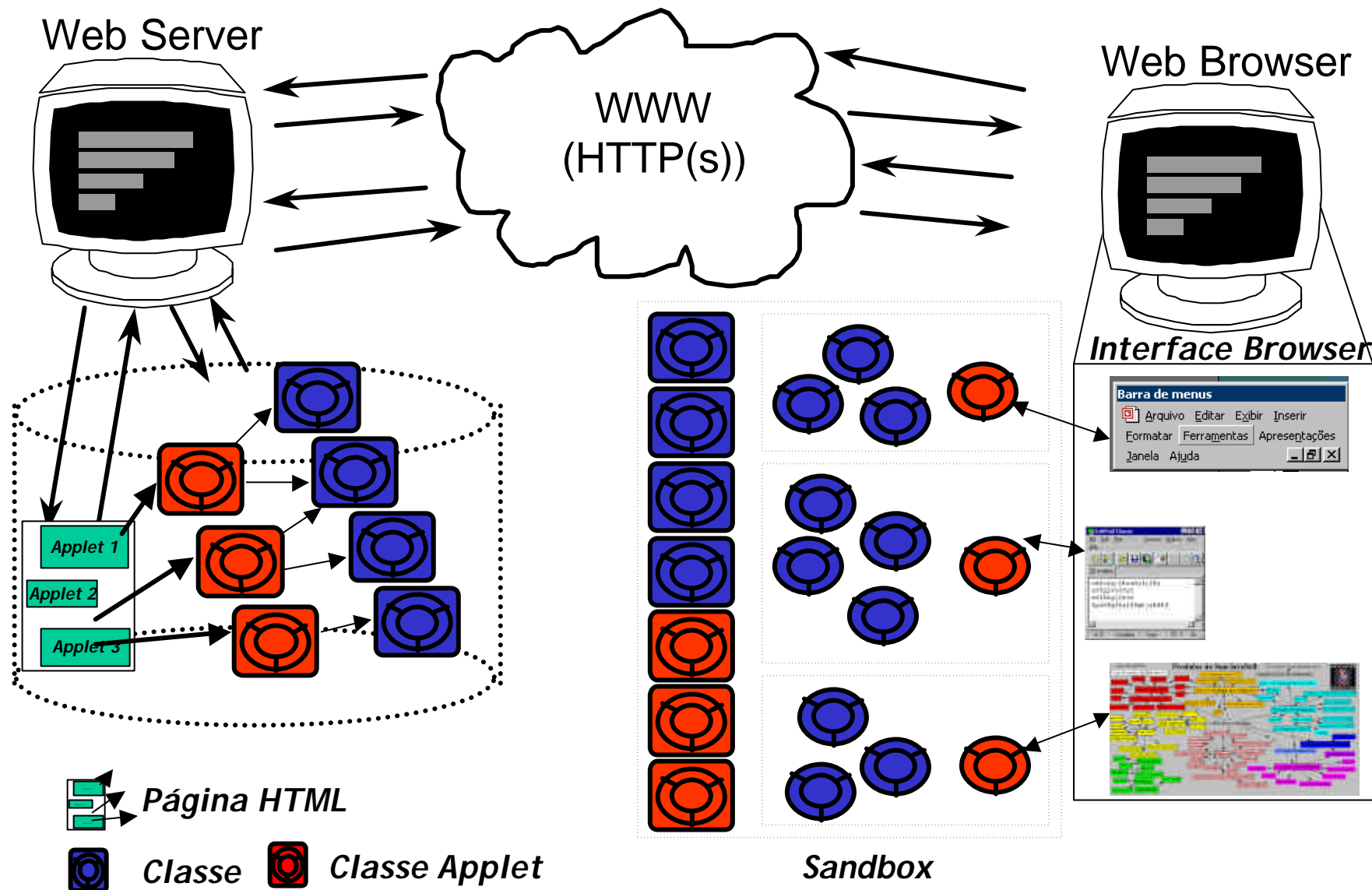
Servlets



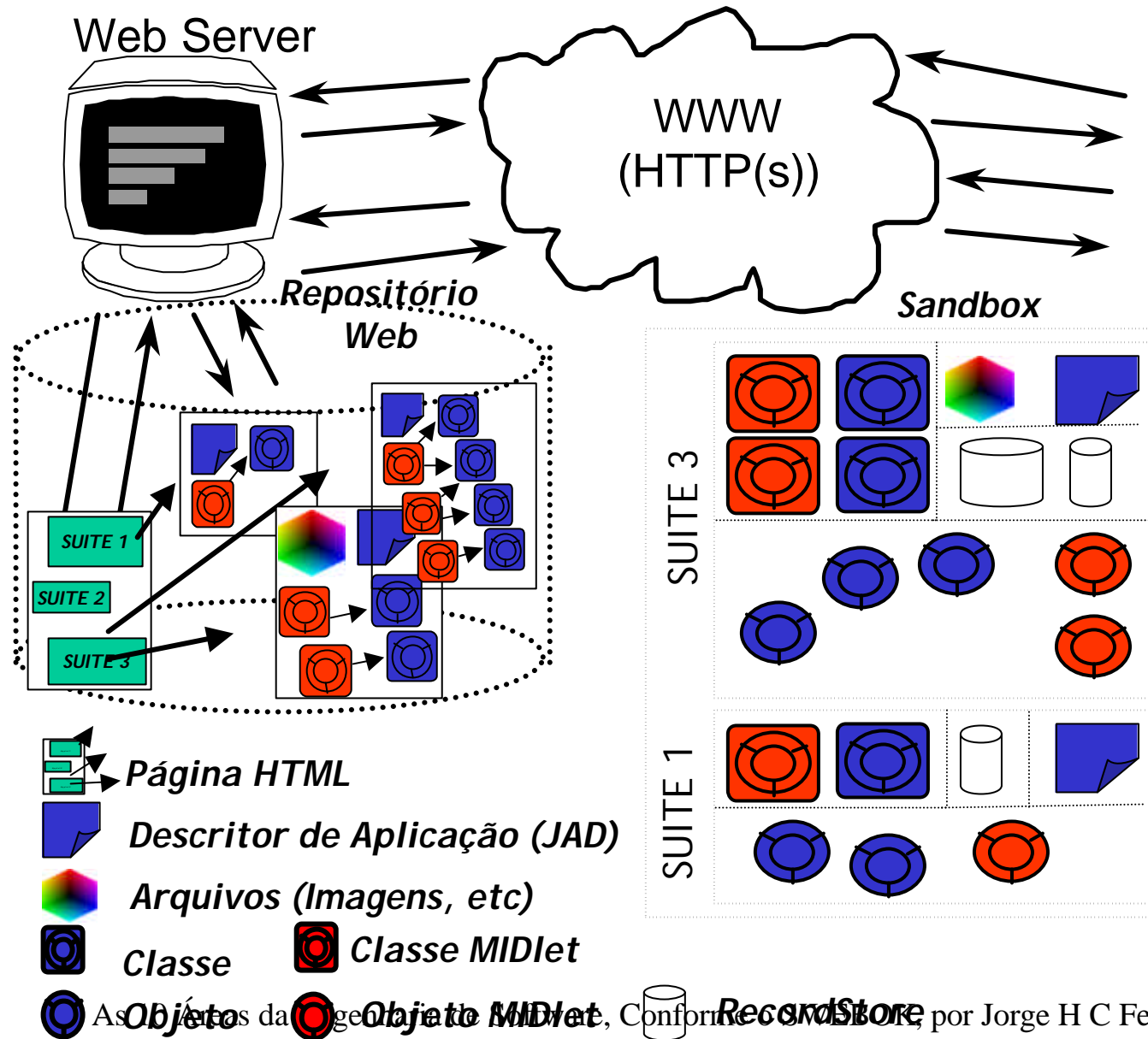
Arquiteturas Multi-Camadas com Enterprise Java Beans (EJBs)



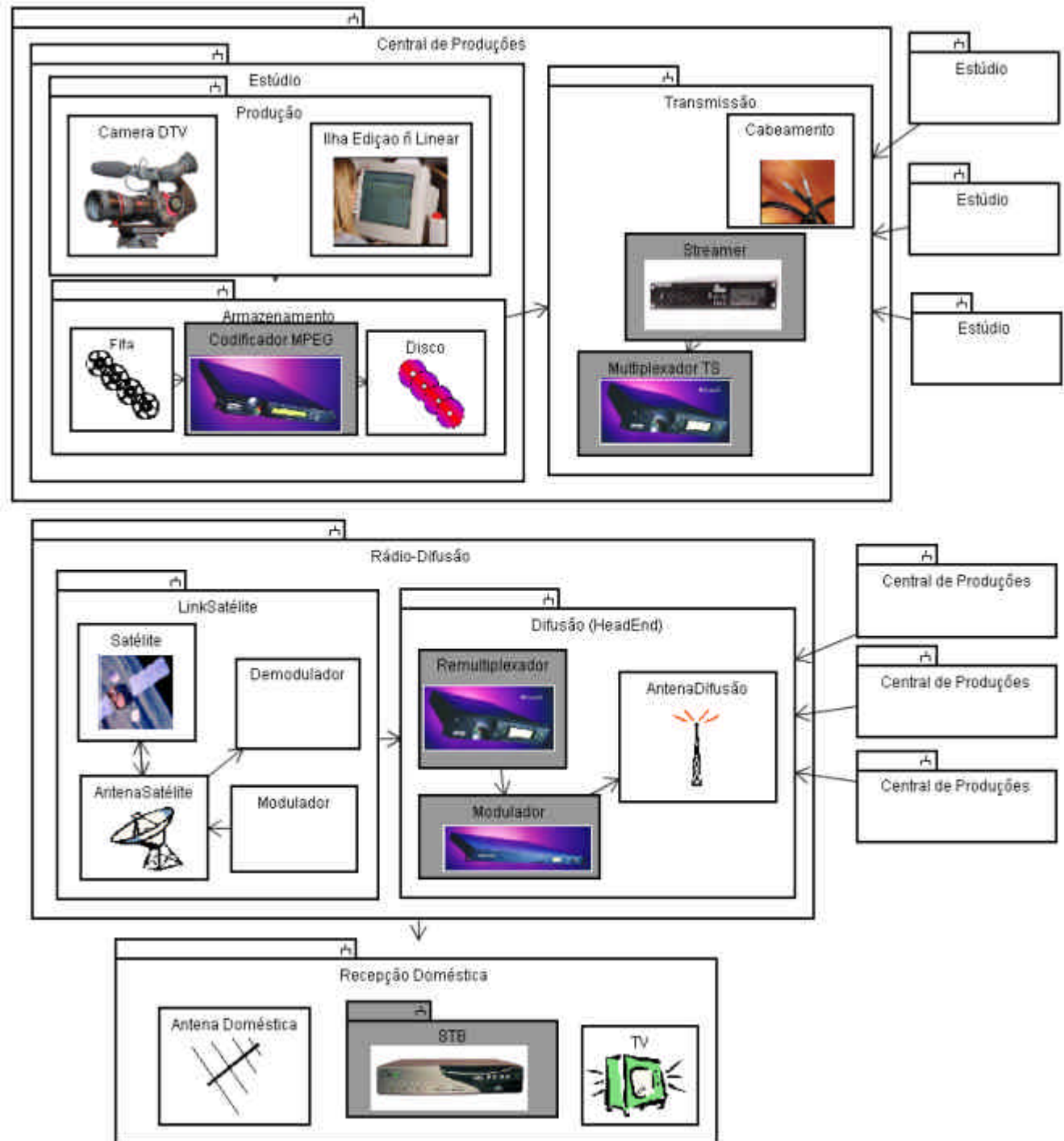
Funcionamento dos Applets



Funcionamento dos MIDlets



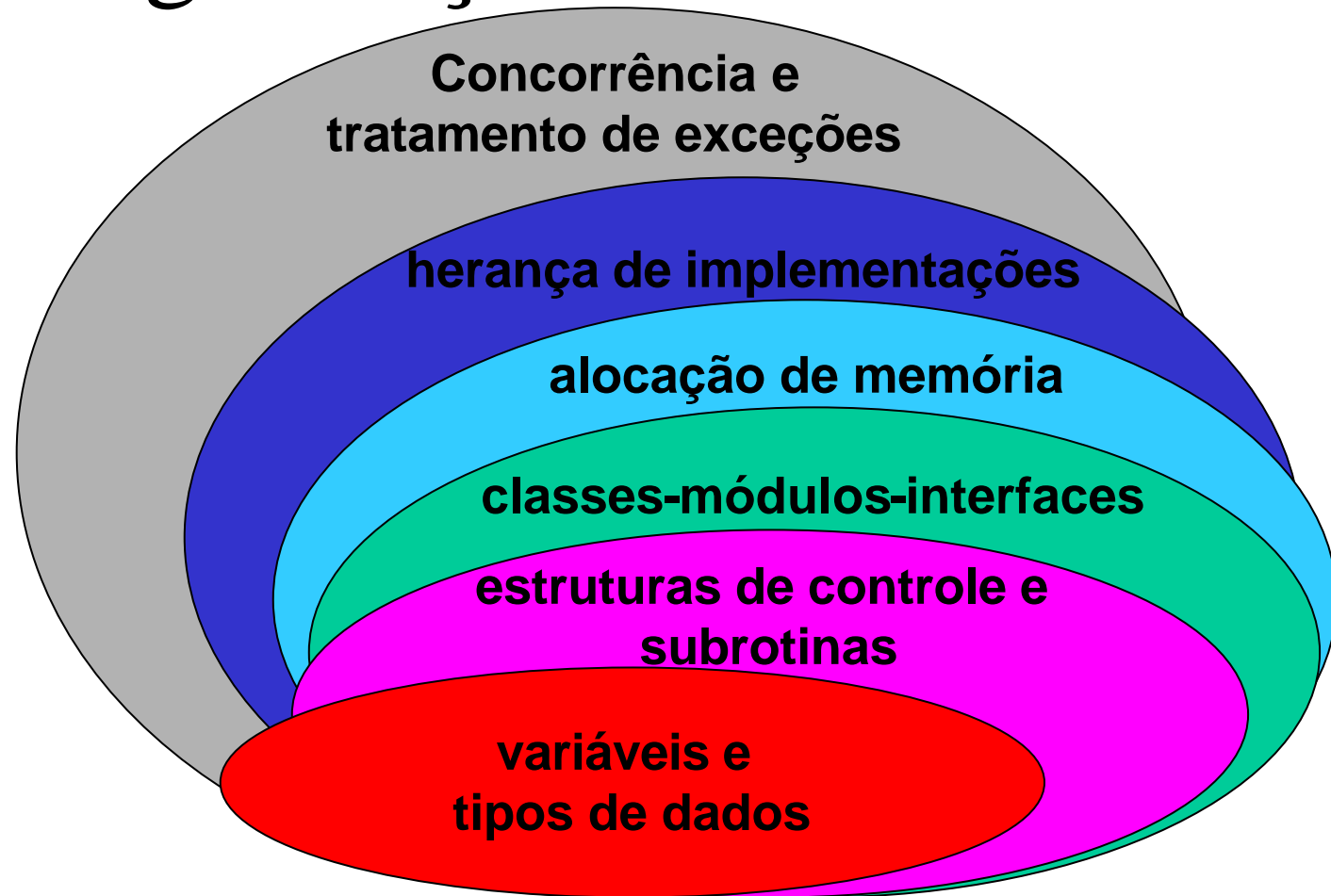
Arquitetura de Sistemas de TV Digital



Construção (implementação) de Software

- Implementação de software funcional e completo através da codificação, auto-validação e auto-teste (teste unitário).
- Duas visões complementares
 - Estilos de construção de interfaces
 - linguístico, formal e visual
 - Formalismos: Linguagem de Especificação Formal Z, linguagem de programação funcional Haskell, ML, LISP-Scheme.
 - princípios de organização
 - redução de complexidade, antecipação à diversidade, estruturação para validação e uso de normas

Java: Um Modelo de Organização Semântica



Teste de Software

- Verificação dinâmica do comportamento de um programa através de uma seqüência finita de casos de tese, adequadamente retirada de um domínio usualmente infinito de execuções.
- Técnicas de teste
 - baseados na intuição, baseados em especificação, baseado em código, baseado em falhas, baseado no uso ou conforme a natureza da aplicação.
 - Black-box e white-box

Plano de Teste

Plano de Teste: <APLICACAO> <VERSÃO-DO-PLANO>

Autor: <AUTOR-PLANO>

Configuração sob teste: <DESCRIÇÃO-CONFIG><VERSÃO-CONFIG>

Histórico de Alterações: <DT-CRIAÇÃO> <ALT-1> <ALT-2>, ..., <ALT-N>

Caso de Teste

CASO 1	Objetivo: <OBJETIVO DO CASO DE TESTE-1>
Passos:	Resultado Esperado
<PASSO-1-1>	<RESULTADO-1-1>
<PASSO-1-2>	<RESULTADO-1-2>
<PASSO-1-n>	<RESULTADO-1-n>
CASO 2	Objetivo: <OBJETIVO DO CASO DE TESTE-2>
Passos:	Resultado Esperado
<PASSO-1-1>	<RESULTADO-1-1>
<PASSO-1-2>	<RESULTADO-1-2>
<PASSO-1-n>	<RESULTADO-1-n>

Log de Teste

Log de Teste: <APLICACAO> <VERSÃO-DO-PLANO>			
Testador: <TESTADOR>			
Configuração sob teste: <DESCRIÇÃO-CONFIG><VERSÃO-CONFIG>			
Data: <DATA DE REALIZACAO DO TESTE>			
Casos de Teste			
CASO 1	Objetivo:	<OBJETIVO DO CASO DE TESTE-1>	
	Passos:	Resultado Esperado	Observações
	<PASSO-1-1>	<RESULTADO-OBTIDO>	<ERRO? Ou ACERTO?>
	<PASSO-1-2>	<RESULTADO-OBTIDO>	<ERRO? Ou ACERTO?>
	<PASSO-1-n>	<RESULTADO-OBTIDO>	<ERRO? Ou ACERTO?>
CASO 2	Objetivo:	<OBJETIVO DO CASO DE TESTE-2>	
	Passos:	Resultado Esperado	
	<PASSO-1-1>	<RESULTADO-OBTIDO>	<ERRO? Ou ACERTO?>
	<PASSO-1-2>	<RESULTADO-OBTIDO>	<ERRO? Ou ACERTO?>
	<PASSO-1-n>	<RESULTADO-OBTIDO>	<ERRO? Ou ACERTO?>

Teste de software

- Análise da satisfação dos requisitos do sistema, definidos por cliente e usuário, feitas por uma máquina construída sob coordenação de um software
 - Se qualquer parte da máquina muda a análise deve ser repetida
 - Software
 - Hardware
 - SGBD
 - Rede
 - Pessoal
 - Se os requisitos mudam, a máquina mudará para atendê-los, e deve ser testada novamente

Grandes Limitações do Teste de Software

- 1 – Você não pode testar o software em todos os sistemas nos quais ele será carregado

Solução: testar em (configurações) de sistemas padronizadas

- 2 – Você não pode dialogar com a máquina até a exaustão, porque a linguagem é finita, mas os diálogos são infinitos

Solução: técnicas de cobertura de testes

Manutenção de Software

- Totalidade das atividades requeridas para prover suporte custo-efetivo a um sistema de software, que pode ocorrer antes ou após a entrega do software.
- Após a entrega do software são feitas modificações com o objetivo de corrigir falhas, melhorar seu desempenho ou adapta-lo a um ambiente modificado.
- Antes da entrega do software são feitas atividades de planejamento.
- Atividades de Manutenção de Software
 - Implementação do processo de manutenção
 - Análise do problema e modificações
 - Implementação da Modificação
 - Revisão e aceitação da manutenção
 - Migração ou Descontinuação

Leis de Lehman Aplicáveis à Manutenção de Software e Sistemas de Grande Porte

- Lei da Mudança Contínua
- Lei da Complexidade Crescente
- Leis da Auto-Regulação
- Lei Conservação da Estabilidade Organizacional
- Lei da Conservação da Familiaridade
- Lei Crescimento Contínuo
- Lei da Qualidade em Declínio

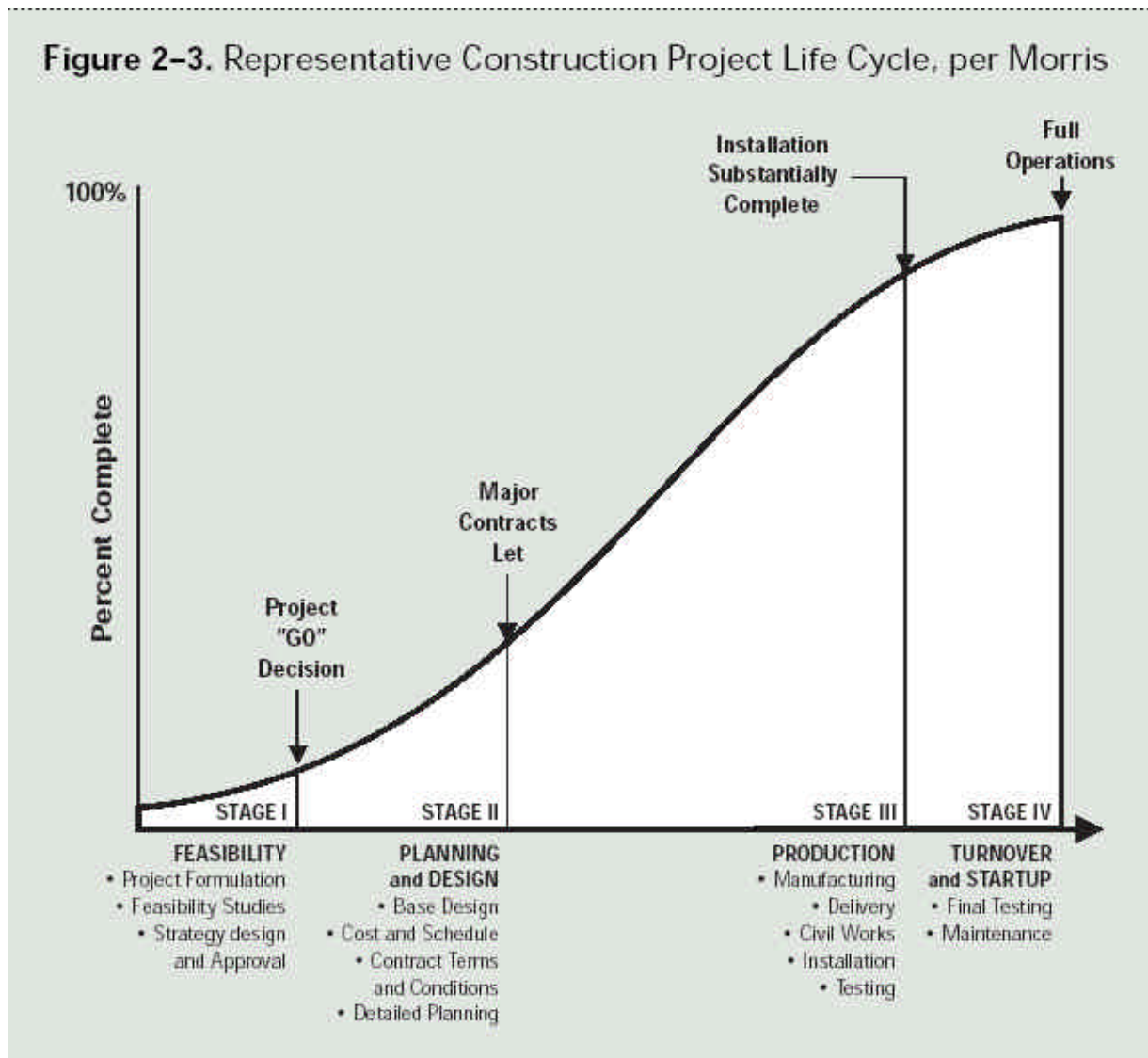
Gerência de Configuração de Software

- Identifica a configuração do sistema (características documentadas do hardware e software que o compõem) em pontos discretos no tempo, de modo a controlar sistematicamente suas mudanças e manter sua integridade e rastreabilidade durante o ciclo de vida do sistema.

Gerência de Engenharia de Software

- Gerencia projetos de desenvolvimento de software.

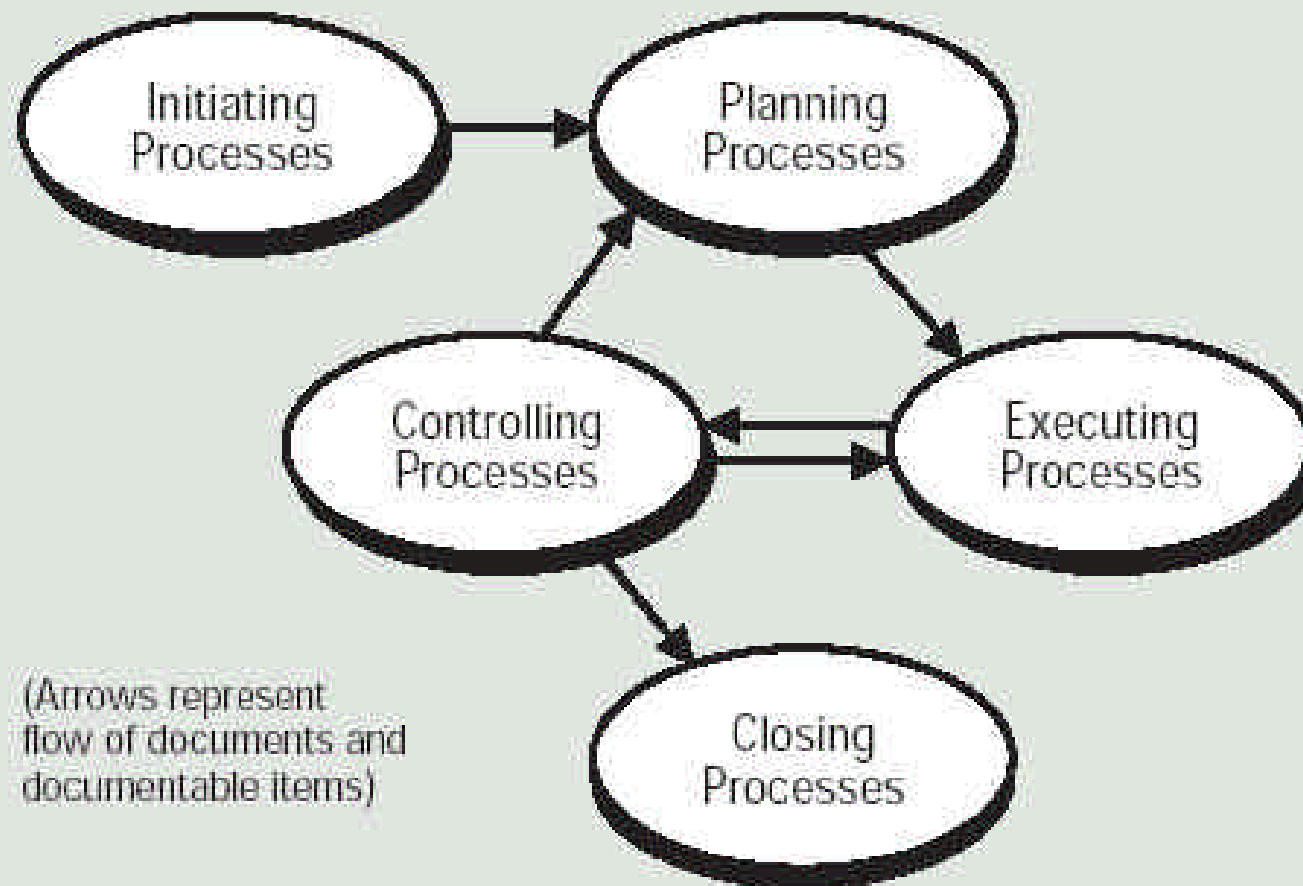
Ciclo de Vida do Projeto [PMBOK, 1996]



As 10 Áreas da Engenharia de Software, Conforme o SWEBOK, por Jorge H C Fernandes, Dez/2004

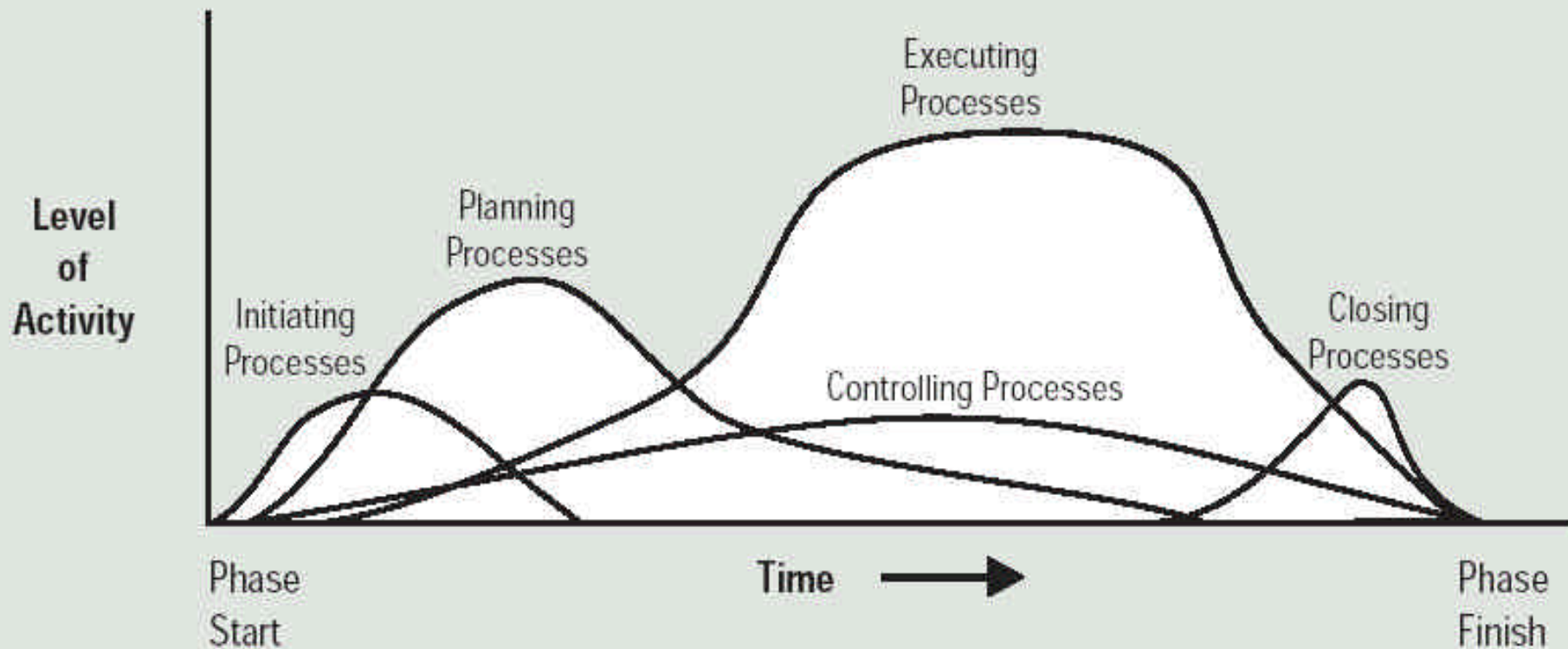
Categorias de Processos [PMBOK, 1996]

Figure 3-1. Links Among Process Groups in a Phase



Consumo de Recursos por Grupo de Processo [PMBOK, 1996]

Figure 3-2. Overlap of Process Groups in a Phase

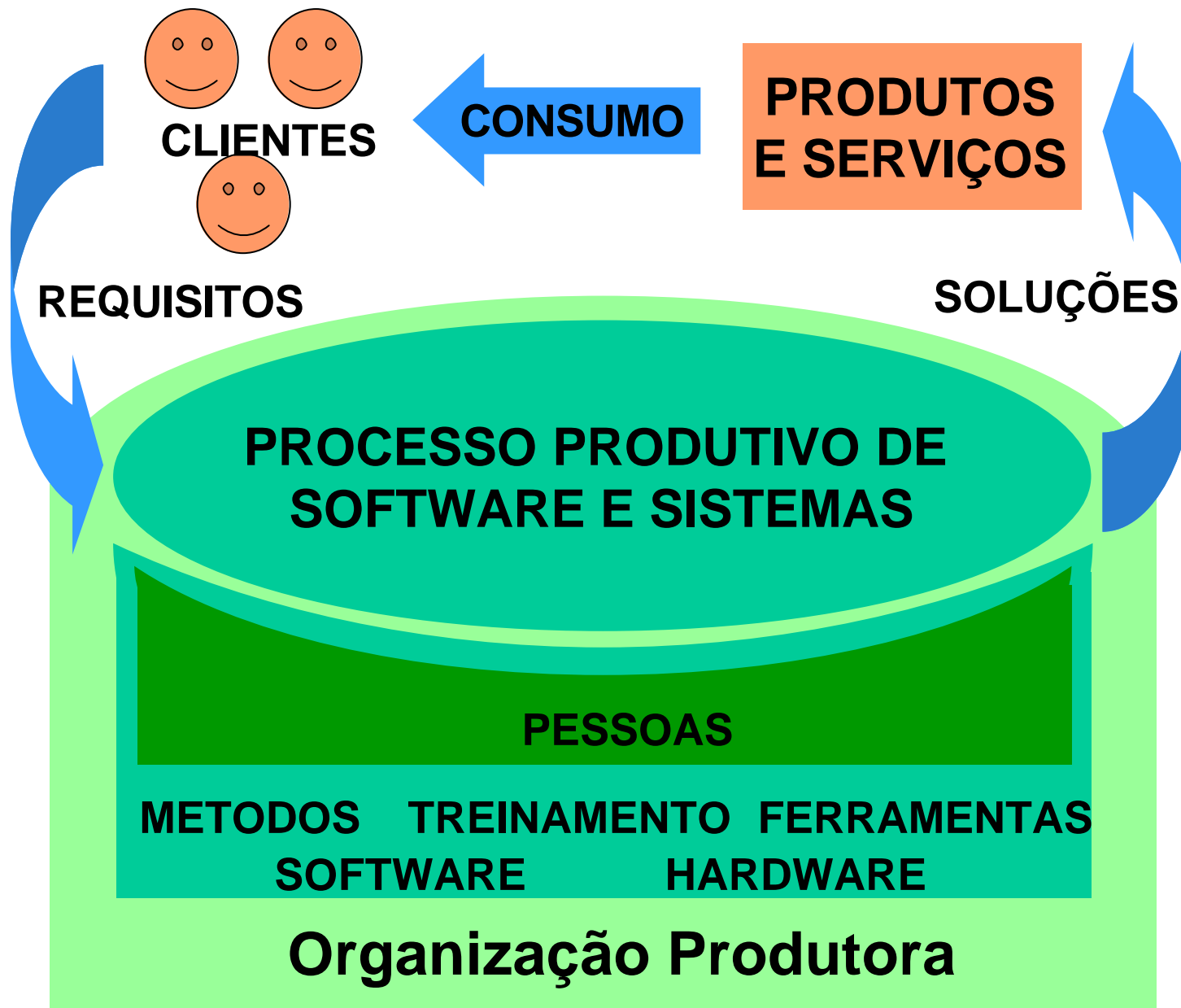


Áreas da Gestão de Projetos

- Gestão de Integração
- Gestão de Escopo
- Gestão de Tempo
- Gestão de Custos
- Gestão de Qualidade
- Gestão de Recursos Humanos
- Gestão de Comunicações
- Gestão de Riscos
- Gestão de Aquisições

Processo de Engenharia de Software

- Define, implementa, mede, gerencia, modifica e aperfeiçoa o processo de desenvolvimento de software



Processos de Software

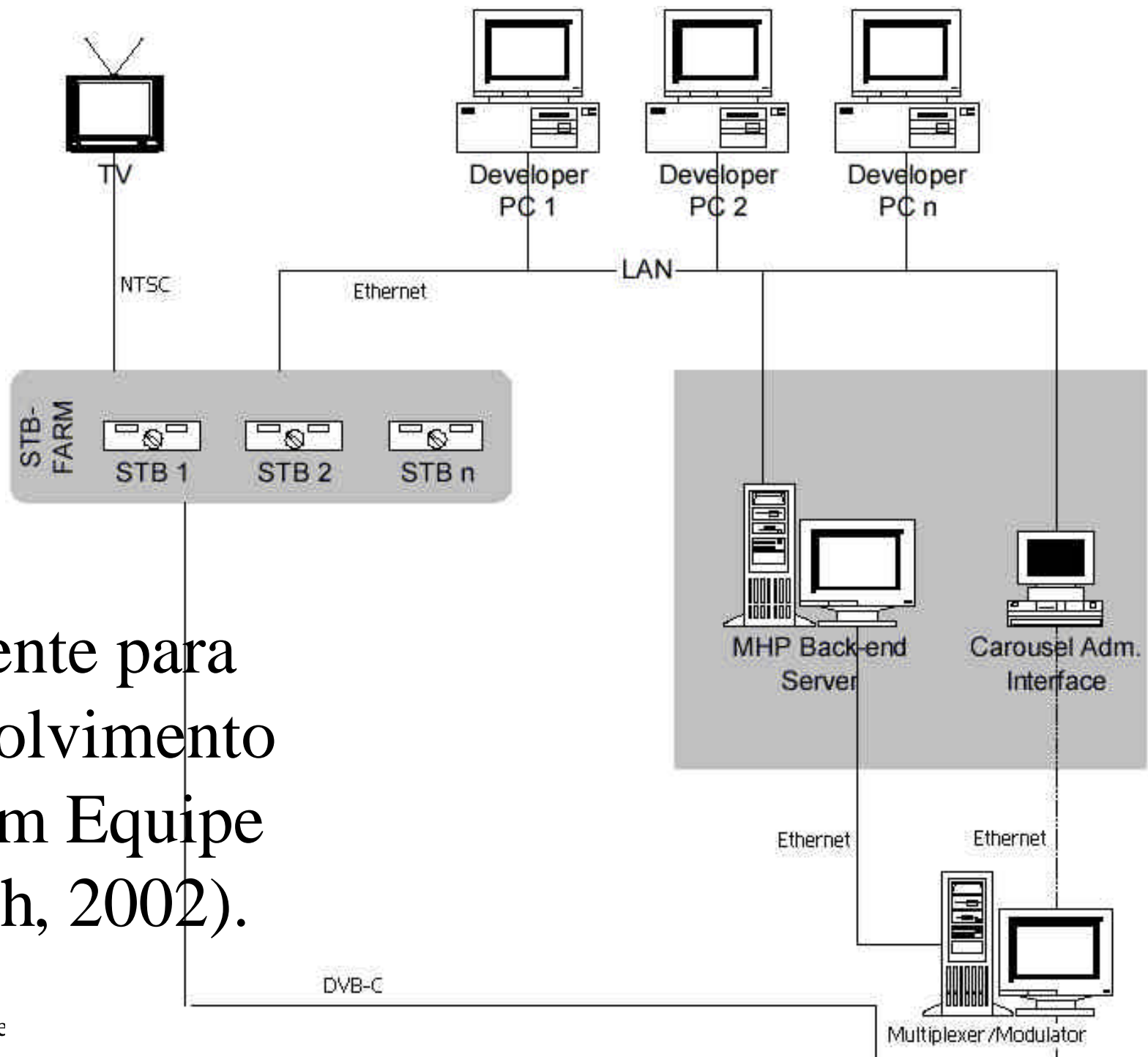
- Processo é disciplina
 - Individual
 - Grupal
 - Organizacional
- Abordagens
 - Qualidade de Processo
 - Estrutura de Processo
 - Automação de Processo
- Por onde começar?

Ferramentas de Engenharia de Software

- Ferramentas
 - Automatizam as tarefas do processo de engenharia de software
- Ambientes de desenvolvimento de software
 - Ferramentas computadorizadas que auxiliam ao processo de desenvolvimento de software
 - Automatizam tarefas repetitivas e bem definidas

Ambiente para Desenvolvimento MHP em Equipe (Frolich, 2002).

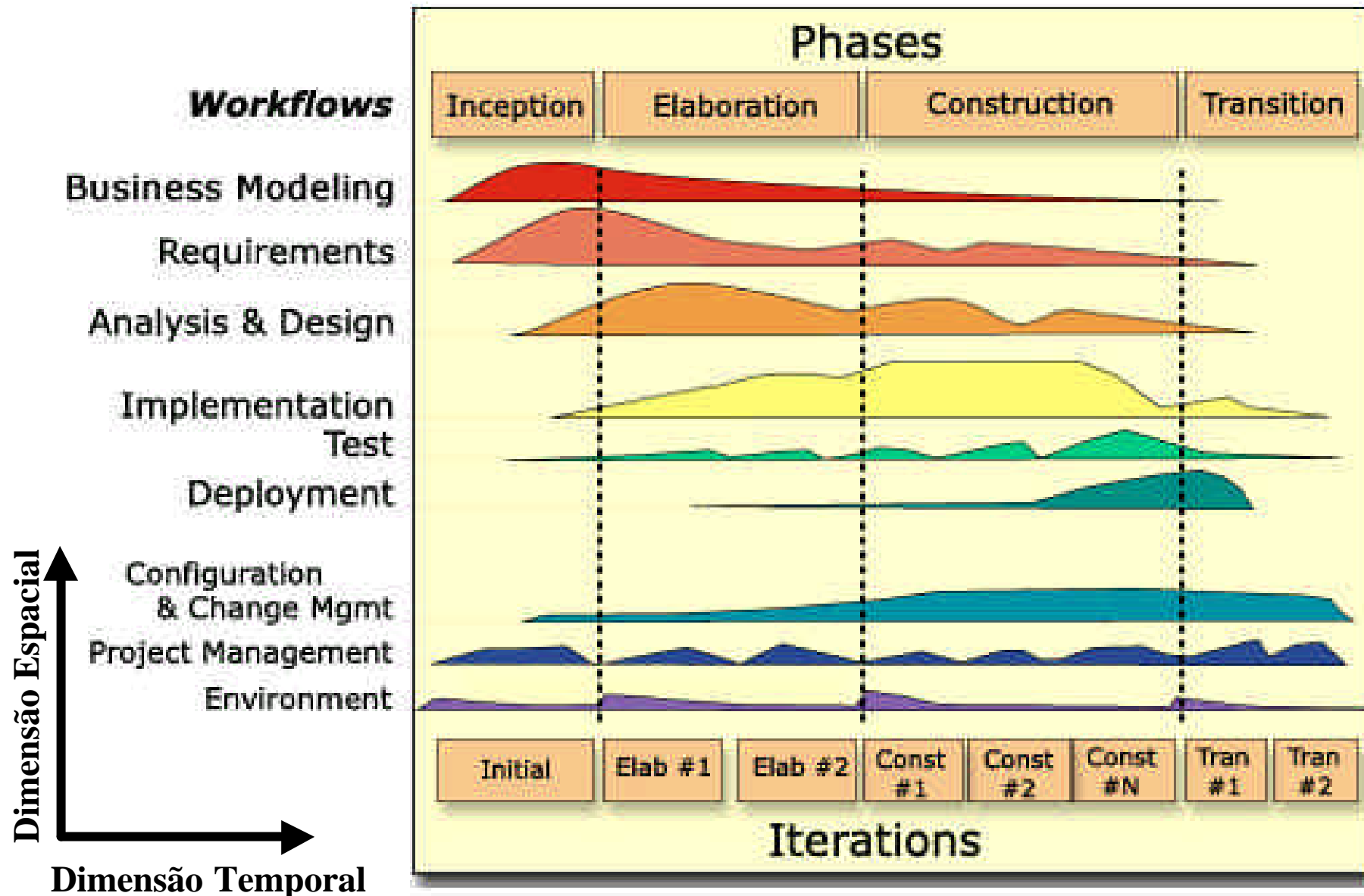
As 10 Áre



Métodos

- Impõem estrutura sobre a atividade de desenvolvimento e manutenção de software com o objetivo de torna-la sistemática e mais propensa ao sucesso.
- Componentes de um Método
 - Notação + vocabulário
 - Lista de atividades e procedimentos para realizá-las
 - Guias para avaliar produto e processo.

Organização do RUP



Qualidade de Software

- Conjunto de atividades relacionadas com garantia de qualidade de software, entre estas as atividades de verificação e validação.

As 10 Áreas da Engenharia de Software, Conforme o SWEBOK

Jorge H C Fernandes
2004