

A Importância da Disciplina de Análise & Design para uma Aplicação

Ivan Zilotti Alencar – 01/06/2006
ialencar@gmail.com

Apresentações

- ◆ Sua empresa/instituição
- ◆ Seu papel
- ◆ Sua experiência
 - Em requisitos
 - Processo de desenvolvimento
 - RUP (Rational Unified Process)
 - XP
 - Orientação a objetos
 - Programação
 - UML



Objetivos

- ◆ Enfatizar as melhores práticas da engenharia de software voltadas para a arquitetura da aplicação.
- ◆ Mostrar a importância da disciplina de Análise & Design do RUP.
- ◆ Expor como os casos de uso fornecem suporte à arquitetura de um sistema de software.

Definição de Processo

- ◆ Um processo define **Quem** está fazendo **O Que, Quando e Como**, para atingir uma certa meta.
- ◆ Um processo de desenvolvimento de software é o conjunto de atividades necessárias para transformar um requisito do usuário em um sistema de software.
[Jacobson]



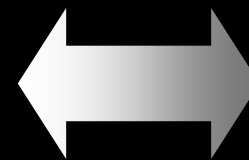
O que é o RUP (Rational Unified Process)?

- ◆ O RUP® é um processo de engenharia de software.
- ◆ Oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento.
- ◆ Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis.



RUP Implementa as Melhores Práticas

Um conjunto de princípios, métodos e processos, organizados e documentados, que aumentam a qualidade e produtividade do desenvolvimento de software.



Desenvolver Iterativamente

Gerenciar Requisitos

Utilizar Arquiteturas de Componentes

Modelar Visualmente

Verificar a Qualidade Continuamente

Gerenciar Mudanças

Prática 3: Utilizar Arquiteturas de Componentes



O que é Arquitetura?

- ◆ A estrutura organizacional de um sistema.
- ◆ Uma arquitetura pode ser repetidamente decomposta em partes que interagem através de interfaces, relações que conectam partes e restrições para associar partes.
 - As partes que interagem através de interfaces incluem classes, componentes e subsistemas.



Benefícios do Foco na Arquitetura

- ◆ Base para a reutilização
 - de componentes
 - de arquitetura
- ◆ Bases para o gerenciamento de projeto
 - Planejamento
 - Recrutamento
 - Entrega
- ◆ Controle
 - Gerenciar complexidade
 - Manter integridade



Arquiteturas Resilientes e Componentizadas

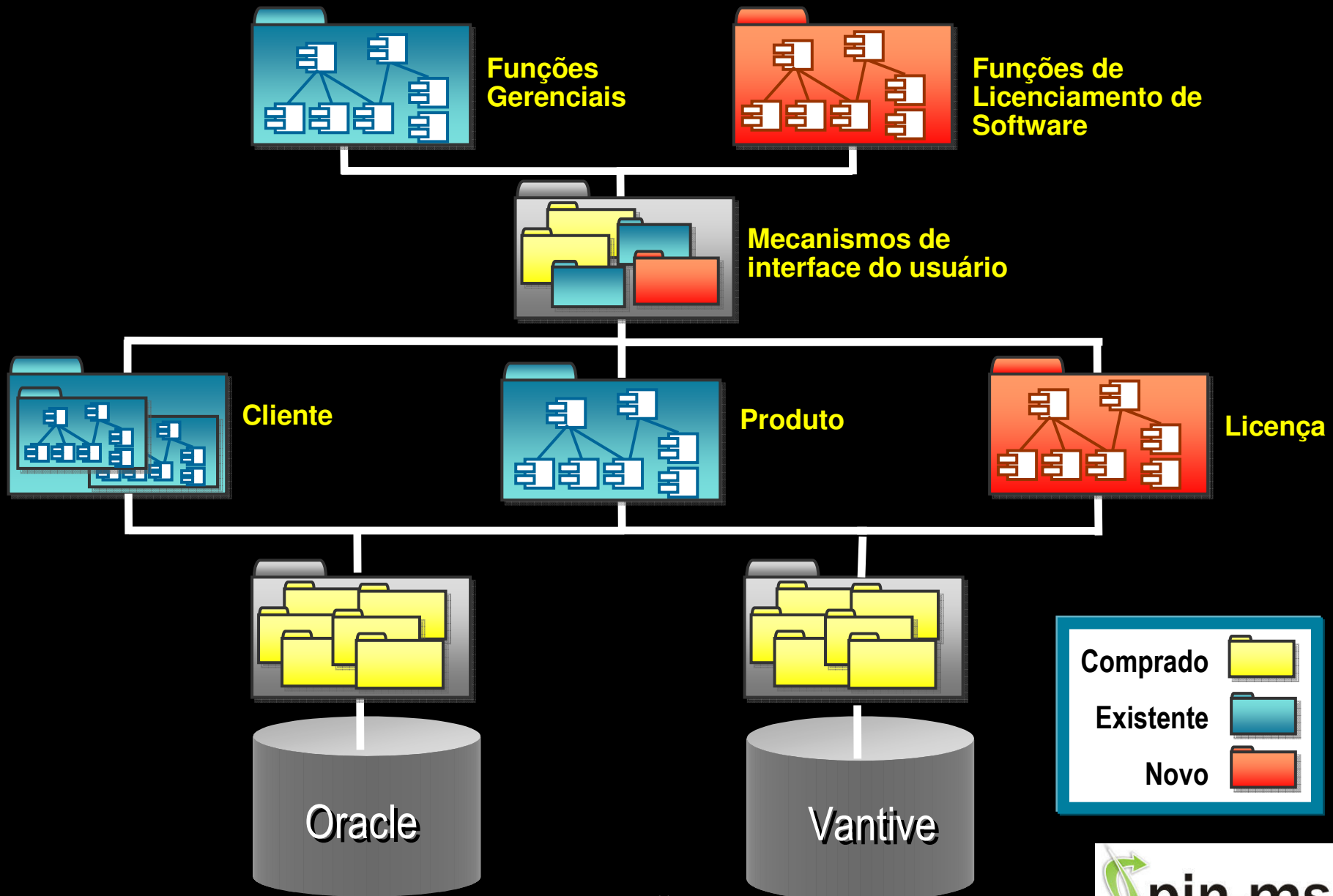
◆ Resilientes

- Satisfaz requisitos atuais e futuros
- Melhora a extensibilidade
- Proporciona reutilização
- Encapsula as dependências do sistema

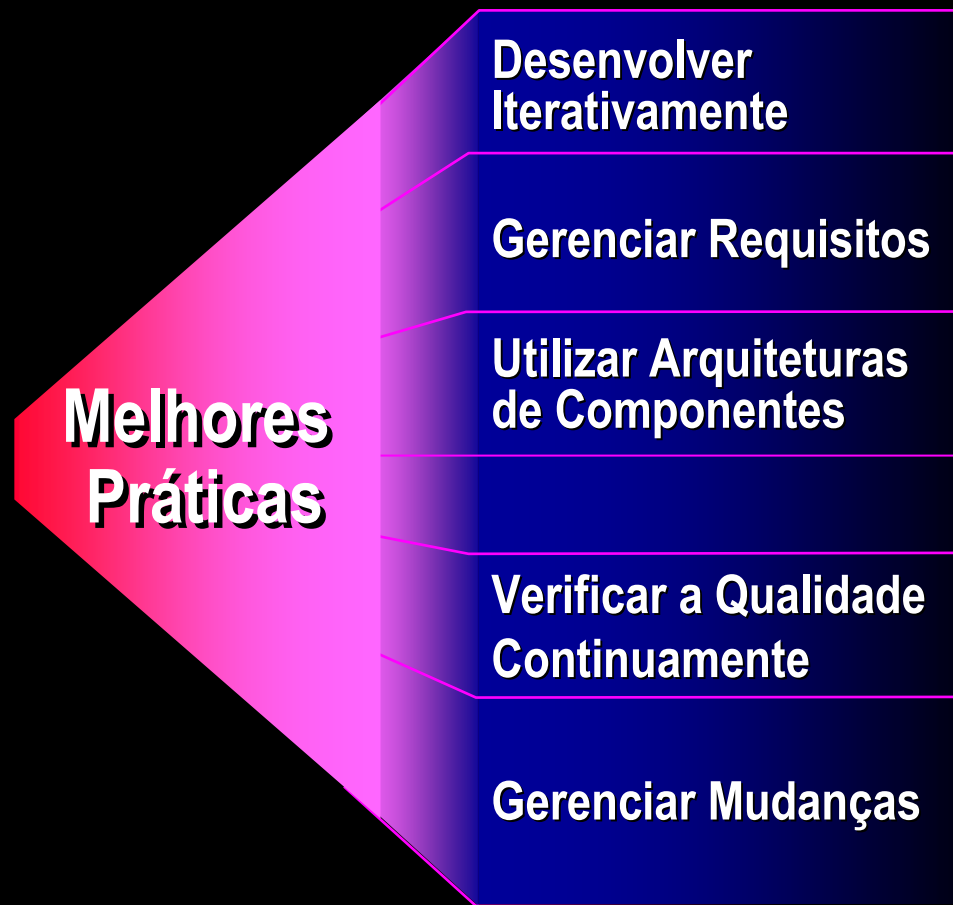
◆ Baseadas em Componentes

- Reutilização ou customização de componentes
- Escolher dentre os componentes comerciais
- Evoluir software existente incrementalmente

Exemplo: Arquitetura baseada em componentes



Prática 4: Modelar o Software Visualmente



Modelar Visualmente (UML)

O que é a Linguagem Unificada de Modelagem?

- ◆ Uma linguagem para...
 - Visualizar
 - Especificar
 - Construir
 - e Documentar os artefatos de um sistema intensivo de software [BOO98].

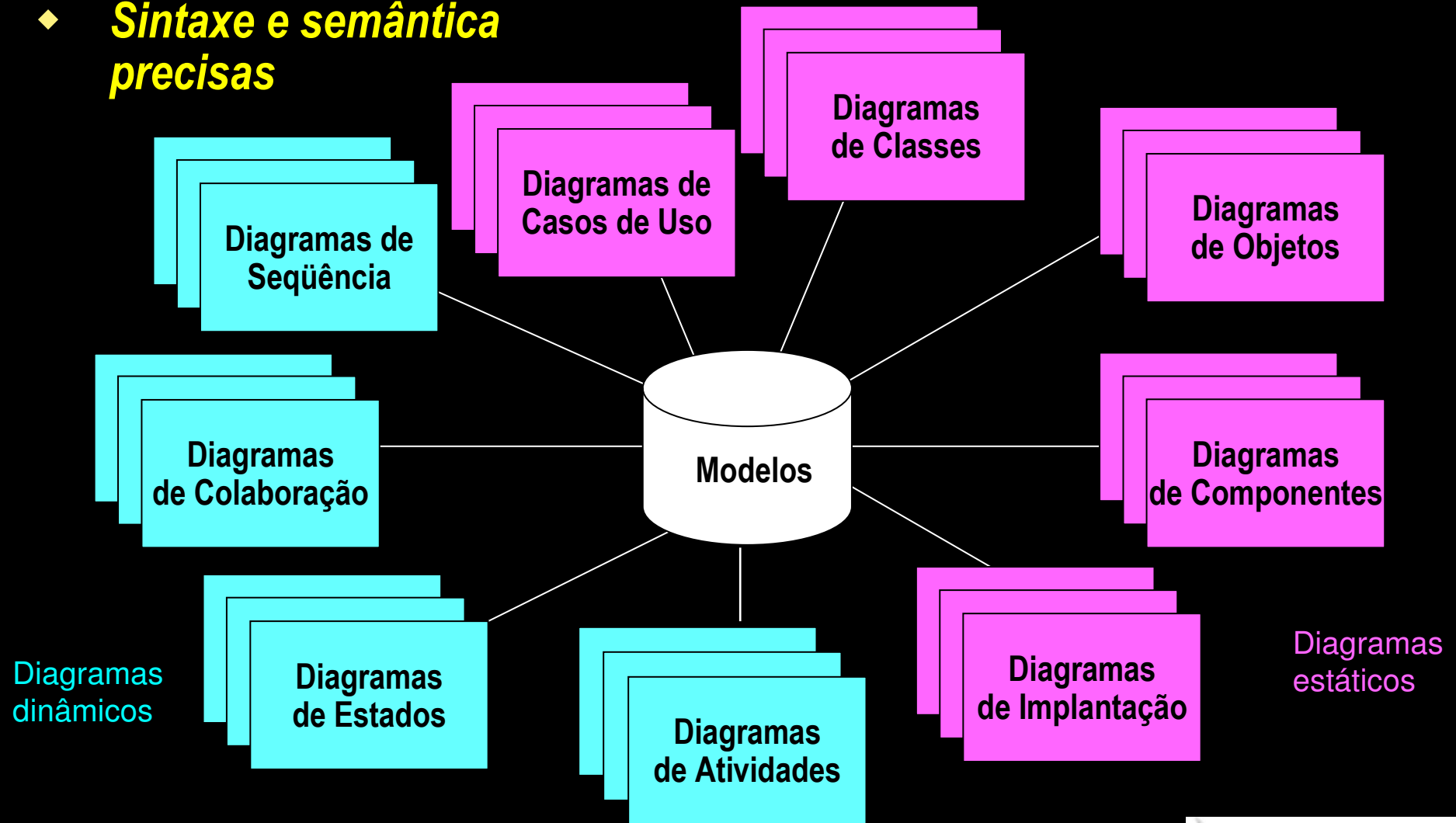
Por que Modelar Visualmente?

Para:

- ◆ Capturar estrutura e comportamento
- ◆ Mostrar como os elementos do sistema se relacionam
- ◆ Ocultar ou expor detalhes quando apropriado
- ◆ Manter design e implementação consistentes
- ◆ Fornecer uma base sólida para a implementação
- ◆ Promover comunicação sem ambigüidade
 - A UML fornece uma linguagem para todos do time

Modelando Visualmente com a UML

- ◆ *Permite múltiplas visões*
- ◆ *Sintaxe e semântica precisas*



Modele Visualmente usando os Diagramas da UML

Diagrama de casos de uso

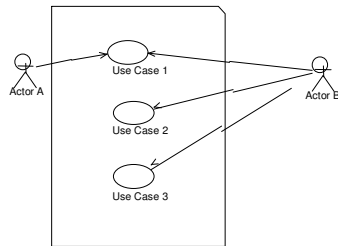


Diagrama de classes

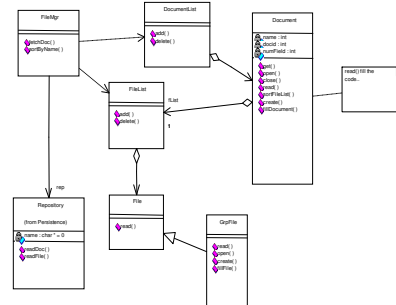


Diagrama de estados

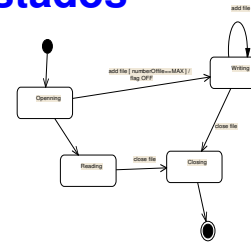


Diagrama de implantação

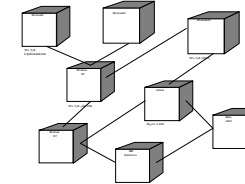


Diagrama de colaboração

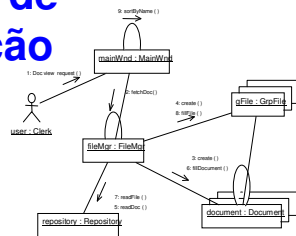
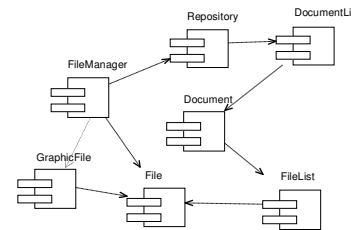


Diagrama de componentes



Sistema alvo

Engenharia avante e Reversa

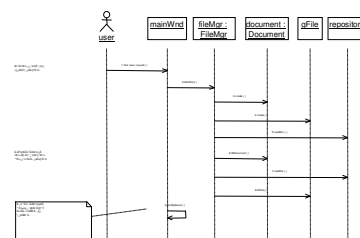
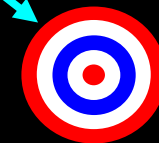
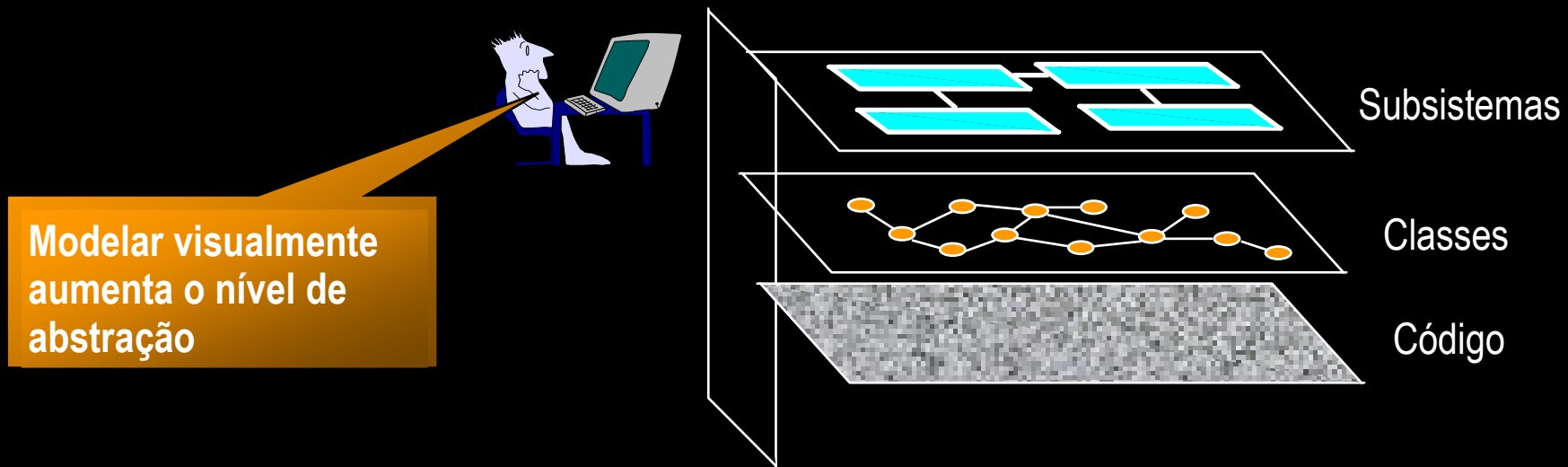


Diagrama de seqüência

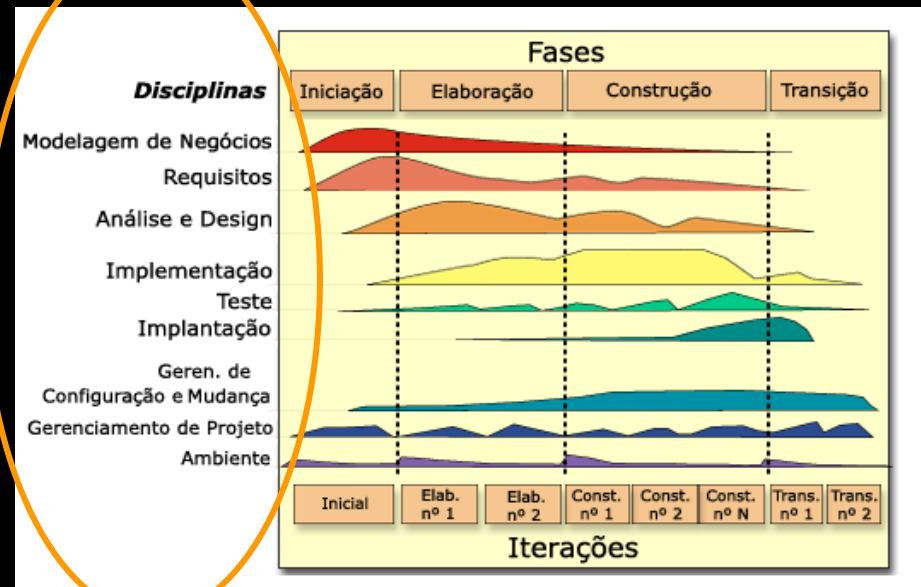
Modelar Mantém a Integridade Arquitetural

- ◆ Detecta e avalia mudanças arquiteturais
- ◆ Comunica mudanças arquiteturais aceitas
- ◆ Sincroniza seu modelo e código fonte durante cada iteração

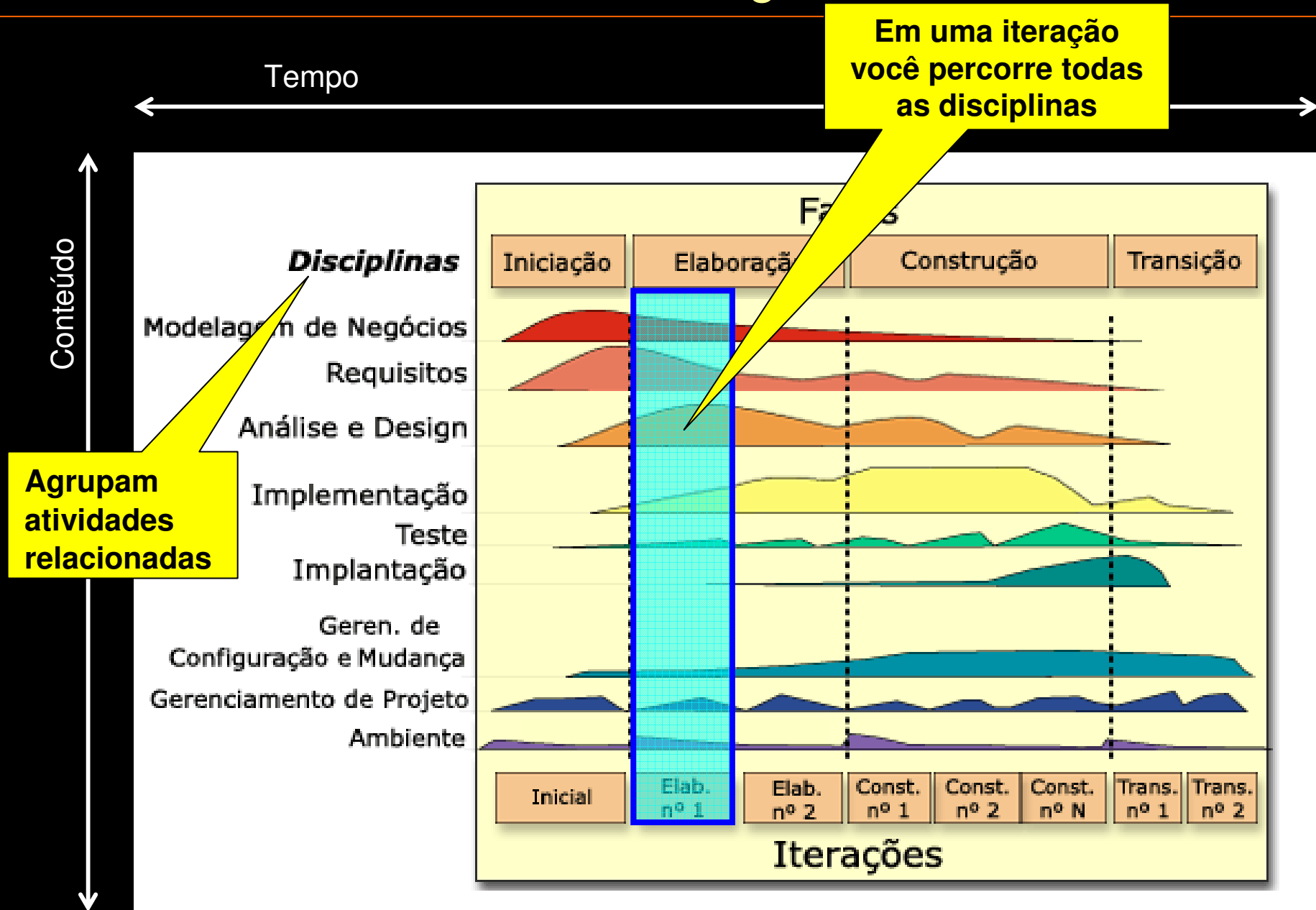


Obtendo as Melhores Práticas através do Processo

- ◆ Uma abordagem iterativa
- ◆ Guias para atividades e produtos de trabalho (artefatos)
- ◆ Processo focado na arquitetura
- ◆ Casos de uso que dirigem design e implementação
- ◆ Modelos que abstraem o sistema



Juntando Tudo: Uma Abordagem Iterativa



Por que Modelar?

“A modelagem é a parte central de todas as atividades que levam à implantação de um bom sistema”

Jacobson, Ivar; BOOCH, Grady;
RUMBAUGH, James; UML Guia do
Usuário.

Cenário Atual

- ◆ Ainda hoje, muitos sistemas são desenvolvidos com base em metodologias estruturadas.
 - Dificuldade para gerenciar e estruturar
 - Requisitos variáveis.
 - Aumento da complexidade.
 - Mudanças.

Por quê?

Vantagens de uma Metodologia OO

- ◆ Grande aceitação pelos desenvolvedores, porque:
 - Possibilita a criação de sistemas de todos os tipos de domínios de problemas.
 - Abrange todos os graus de tamanho e complexidade.

O Foco na Arquitetura

- ◆ Permite que o sistema evolua, mesmo após a entrega do produto.
- ◆ Concentra-se nas funcionalidades chaves do sistema (5 a 10% do total de casos de uso).
- ◆ Gera o mínimo possível de dependências entre módulos e subsistemas.

Conclusão

Um processo iterativo aliado à preocupação com a arquitetura guiam o time para o sucesso!