

# Introdução a **Arquitetura orientada a Serviços** e **Desenvolvimento orientado a modelos**

Vítor Braga – [vtb@cin.ufpe.br](mailto:vtb@cin.ufpe.br)

# Objetivos

- Mostrar os principais problemas nos processos de desenvolvimento de software
- Apresentar os principais conceitos, motivações e mitos de *Model Driven Engineering* (MDE) e *Service Oriented Architecture* (SOA)



Qual o principal problema no desenvolvimento de Software?

**Complexidade !!**

# Por que projetos de software são complicados?



# Algumas razões

- Software é:
  - Complexo
  - Abstrato
- Requisitos incompletos
- Tecnologia muda muito rápido
- Mudança é inevitável

# Evolução ...

- Linhas de Produtos de Software
- Programação Orientada a Aspectos
- Desenvolvimento Baseado em Componentes
- **Desenvolvimento baseado em modelos**
- **Arquitetura Orientada a Serviços**

# Desenvolvimento Orientado a Modelos



# Desenvolvimento Orientado a Modelos

- A principal motivação é aumentar a produtividade:
  - Reutilização
  - Independência de tecnologia
  - Automação
- Aumentar o nível de abstração
  - Foco no modelo (modelo), não no código
- “The model is the code”

# Por que modelar é importante?

- A modelagem é essencial para a atividade humana, porque cada ação é precedida pela construção (implícita ou explícita) de um modelo.
- A modelagem é usada em vários domínios e permite que diversas pessoas (com perfis diferentes) participem da concepção do sistema

Vídeo: Modeling Through the Ages

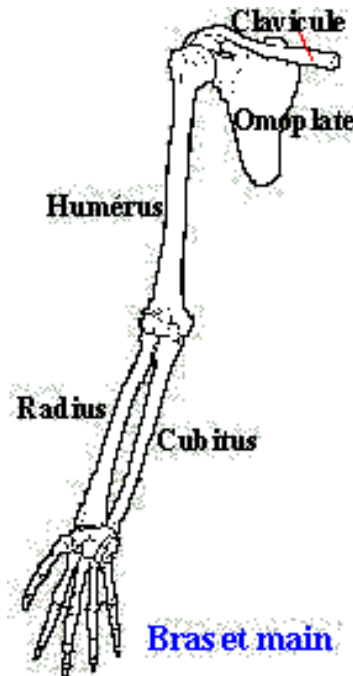
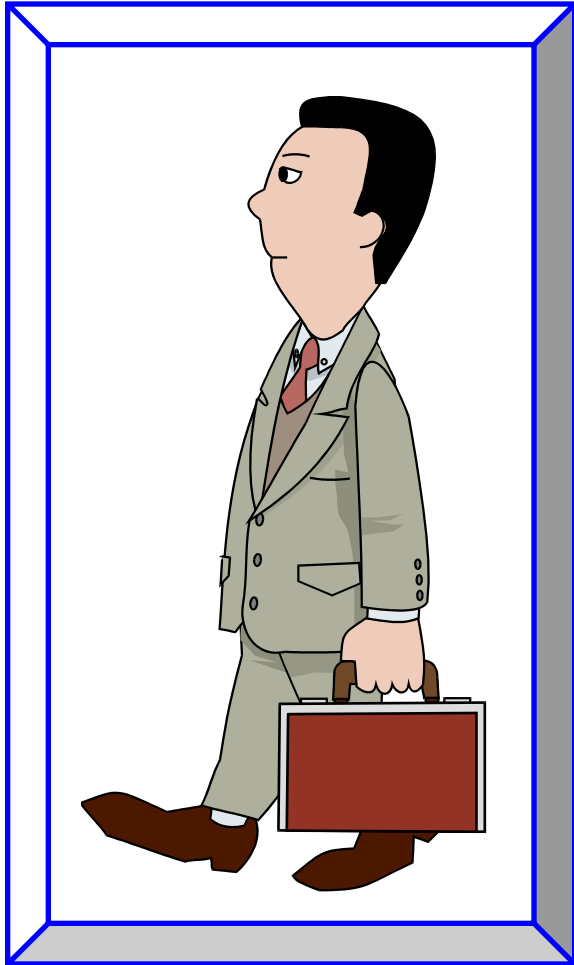


O que é um modelo?

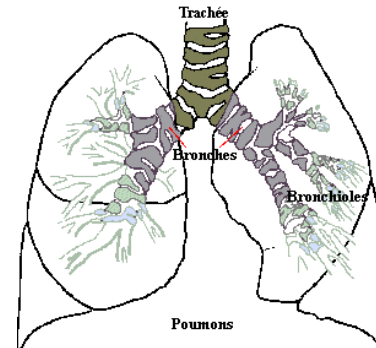
# O modelo é uma visão parcial do sistema

O Sistema ← repOf

Vários modelos do sistema  
(partial views)



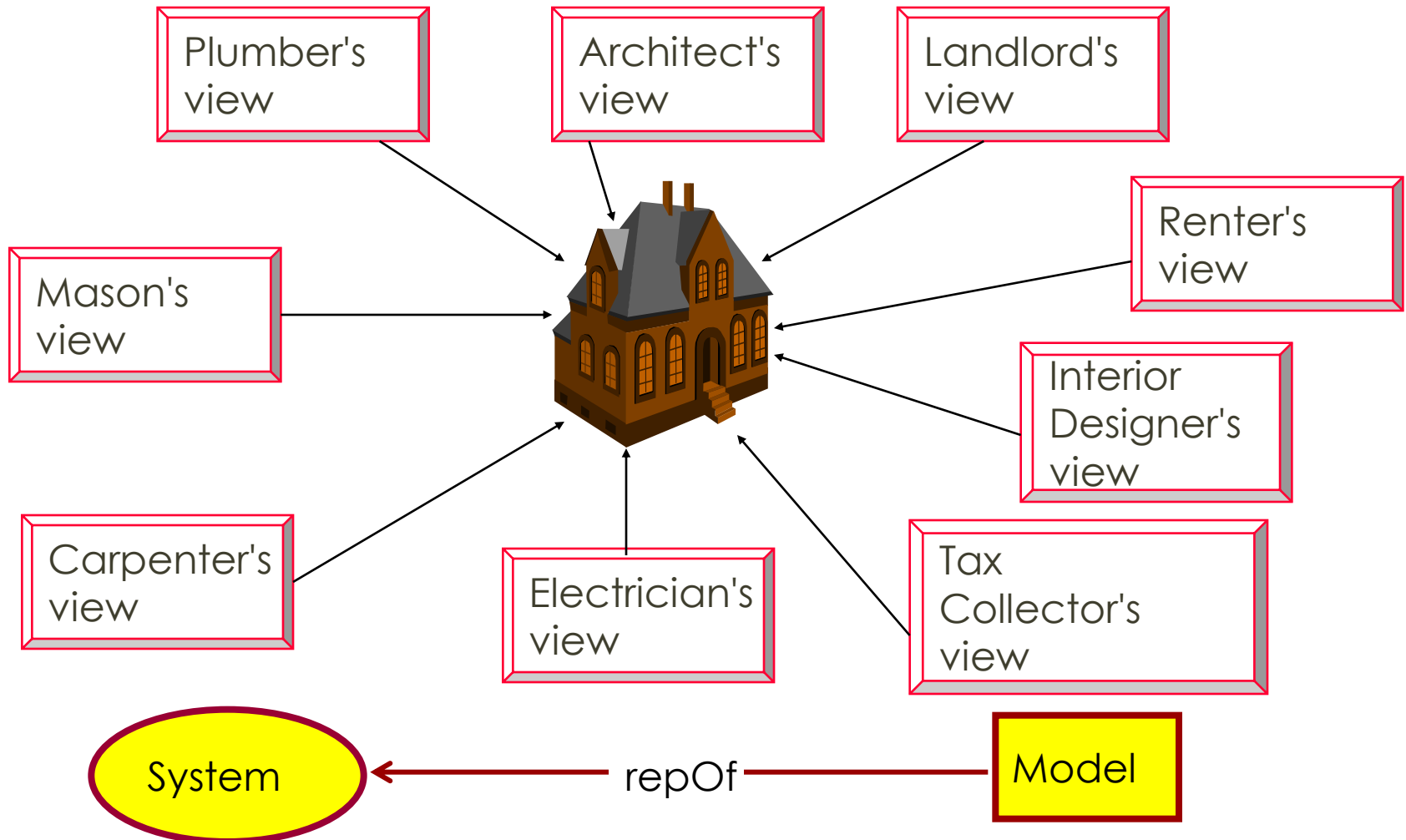
Skeleton model



Respiratory model

Other Models  
muscular,  
nervous,  
circulatory,  
digestive,  
endocrinous,  
etc.

# Multiples visões



# Princípios, Padrões e Ferramentas

Princípios

Model-Driven Engineering (MDE)

Padrões

MDA™  
Model-Driven  
Architecture  
(OMG)

MIC  
Model  
Integrated  
Computing

Software  
Factories  
(MS)

Other  
Standards

Ferramentas

Eclipse  
EMF  
GMF

GME

Microsoft  
Visual Studio  
Team system  
DSL Tools

Other  
Tools

# Nomes, Termos e Siglas

- MDE Model Driven Engineering
- ME Model Engineering
- MDA Model Driven Architecture
- MDD Model Driven Development
- MDSD Model Driven Software Development
- MDSE Model Driven Software Engineering
- MM Model Management
- MDDE Model Driven Data Engineering
- ADM Architecture Driven Modernization
- MDRE Model Driven Reverse Engineering
- DSL Domain Specific Language
- DSM Domain Specific Modeling
- MDE is a generic term; ME ~ MDE
- MDA™ and MDD™ are OMG trademarks; MDD is a protection trademark (no use as of today/just reserved by OMG for future use)
- MDSD like MDSE is sometimes used instead of MDD when one does not wish to be associated to OMG-only technology, vocabulary and vision.
- ADM is another standard intended to be the reverse of MDA: MDA covering forward engineering while ADM covers backward engineering. ADM ~ MDRE
- MM ~ MDDE
- DSM is more Microsoft marked but of increasing use by the academic and research community



# Some OMG Trademarks

- **OMG's Registered Trademarks Include:**

- MDA®
- Model Driven Architecture®
- UML®
- CORBA®
- CORBA Academy®
- XMI®

- **OMG's Trademarks Include:**

- CWM™
- Model Based Application Development™
- MDD™
- Model Based Development™
- Model Based Management™
- Model Based Programming™
- Model Driven Application Development™
- Model Driven Development™
- Model Driven Programming™
- Model Driven Systems™
- OMG Interface Definition Language (IDL)™
- Unified Modeling Language™
- <<MOF



# Model Driven Architecture (MDA)

- MDA é uma abordagem de desenvolvimento de sistemas que se baseia na ideia da separação da especificação das funcionalidades do sistema dos detalhes de implementação [OMG]
- Principais Objetivos
  - Portabilidade
  - Interoperabilidade
  - Reutilização

# Model Driven Architecture (MDA)

- MDA define mecanismos para:
  - Especificar o sistema independente da plataforma que o suporta
  - Especificar plataformas de execução
  - Escolher uma plataforma para um sistema
  - Transformar a especificação do sistema em uma plataforma específica
- Esta abordagem ajuda a focar nos aspectos de **negócio** da solução ao invés dos aspectos **técnicos**

# MDA

- MDA sugere um desenvolvimento baseado na utilização de modelos para direcionar o entendimento, projeto, construção, instalação e manutenção dos sistemas
- Esta abordagem recomenda a criação de modelos em três níveis de abstração:
  - Modelo Independente de Computação
  - Modelo Independente de Plataforma
  - Modelo Específico de Plataforma

# MDA

Abordagem **orientada negócio** para a construção de sistemas de software onde **modelos são refinados** a partir das necessidades do negócio até a sua realização **em soluções de software**

## CIM

Computation Independent modelo

Captura o contexto e requisitos de negócio



Transformações Automatizadas

## PIM

Platform Independent modelo

Serviços do sistema independentes de Tecnologia



Transformações Automatizadas

## PSM

Platform Specific modelo

Realização dos serviços em uma plataforma



Transformações Automatizadas

## Código



# CIM

Computation Independent modelo

Captura o contexto e requisitos de negócio



Transformações Automatizadas

# PIM

Platform Independent modelo

Serviços do sistema independentes de Tecnologia



Transformações Automatizadas

# PSM

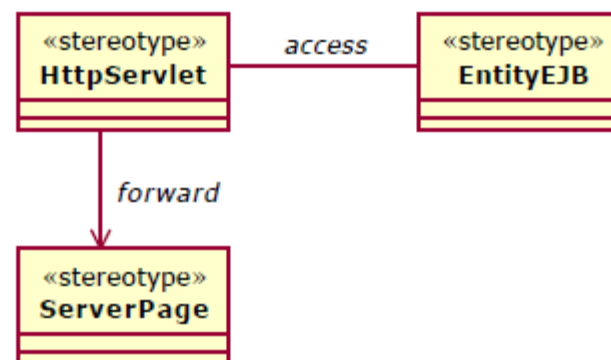
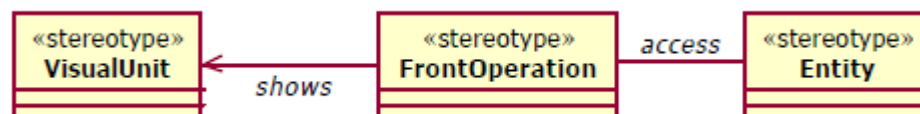
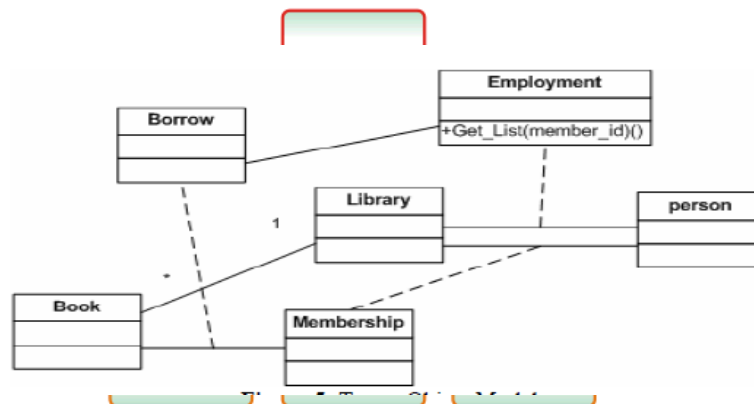
Platform Specific modelo

Realização dos serviços em uma plataforma



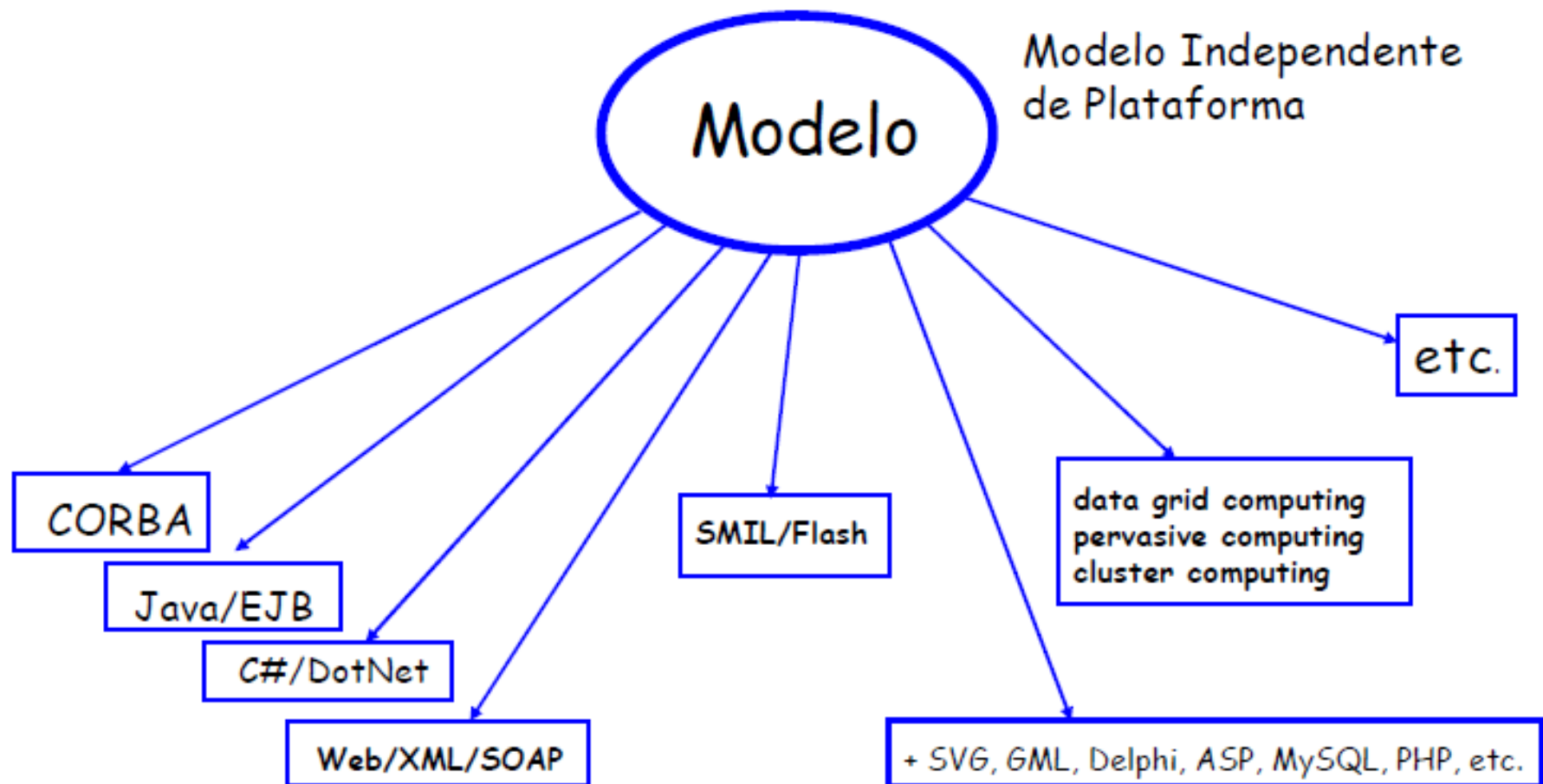
Transformações Automatizadas

# Código

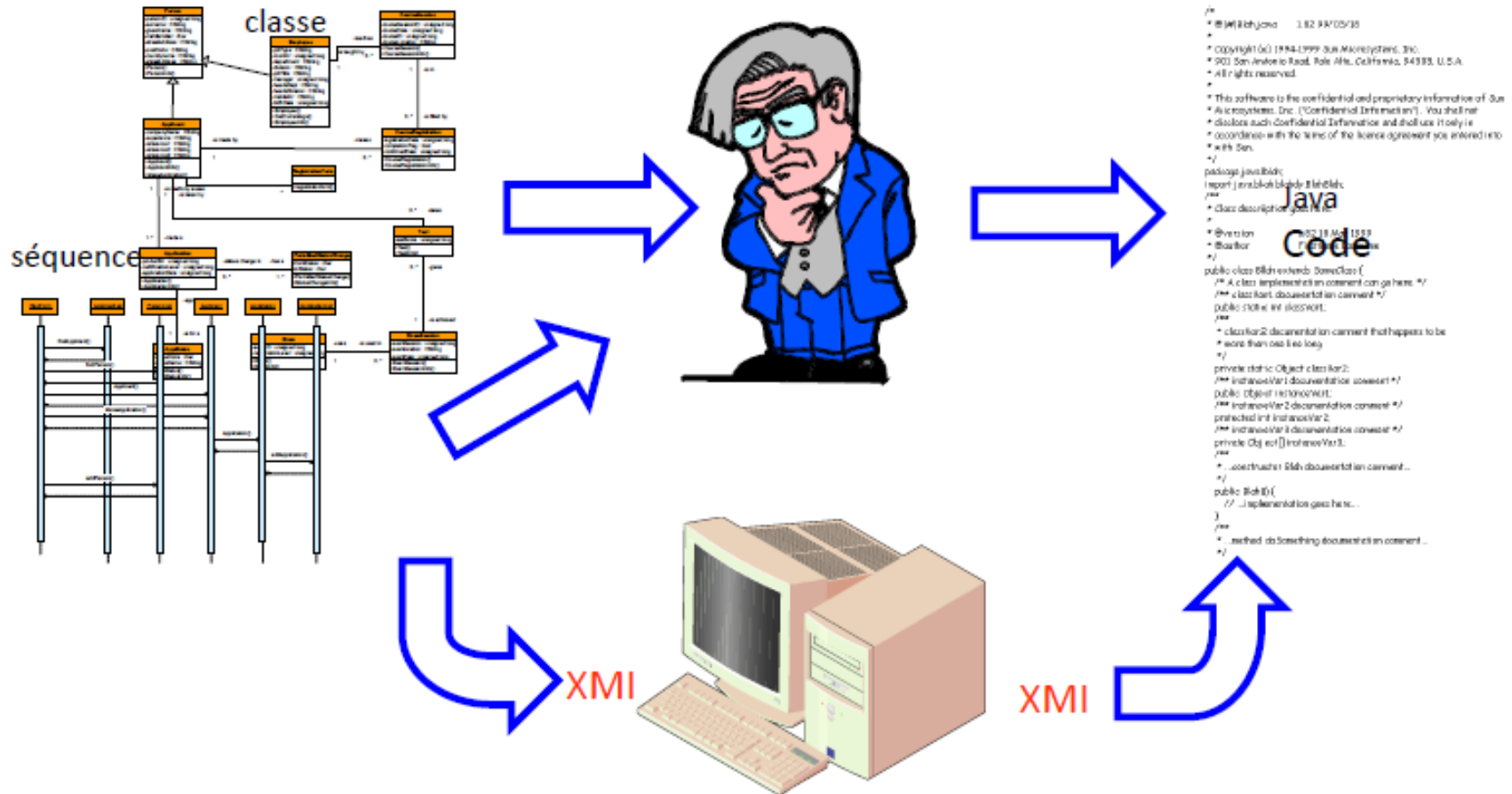


# O sonho

Write Once, Run Anywhere → **Model Once, Generate Anywhere**



# Desenvolvimento de Sw: Visão MDA





# Outras Abordagens

- MODA-TEL
- MASTER
- MIDAS
- C3
- MDDP
- Model Based Software Development Process for Production Applications
- General framework for model-driven development based on Model-Driven Architecture
- MDD Process Framework
- Agile Model Driven Development
- Process, product, conceptual and system models
- SDL-MDD
- MDA effects on the RUP
- Telelogic Harmony-SE.
- RUP SE
- Vitech Model-Based System Engineering (MBSE) methodology
- UMDD: User Model Driven Software Development
- A Model Driven Approach for the Alignment of Business and Information Systems Models
- MDSOFA: A MODEL-DRIVEN SOFTWARE FACTORY

# Desafios

# Tecnologia

- Maior padronização
- Processos ágeis para projetos de pequeno e médio porte
- Gerencia de configuração
- Reuso de sistemas legados (engenharia reversa)
- Ferramentas para transformações de modelos

# O que é Arquitetura Orientada a Serviços ?

# Vídeo: What is Service Oriented Architecture?



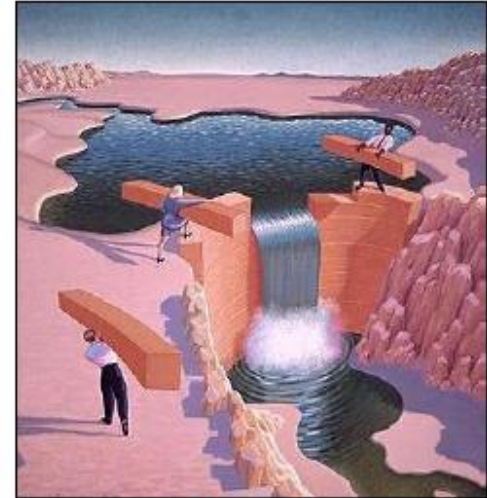
### **Melhorar os processos**

melhora a qualidade  
dos serviços e produtos



### **Redução de Custos**

da operação de TI



### **Agilidade**

para atender as novas  
demandas e oportunidades



### **Flexibilidade**

para responder as mudanças



### **Interoperabilidade**

entre os sistemas legados e novos



### **Integração**

entre o Negócio e a TI

- Diferentes interpretações, dependendo do interlocutor...



**Diretor Negócios**

“SOA é uma tecnologia que cria um ambiente de negócio ágil e provê vantagem competitiva ou maior valor.”



**Gerente TI**

“SOA é conjunto de processo, estrutura e diretrizes de governança que permite alinhar TI às necessidades do negócio.”



**Arquiteto SW**

“SOA é uma arquitetura de software baseada em padrões abertos que permite integrar aplicações novas e existentes.”



**Desenvolvedor**

“SOA é um *framework* baseado em *webservices* que permite invocar objetos remotamente utilizando protocolo SOAP, baseado em XML.”



## O que **não** é a SOA ?

SOA **não** é uma tecnologia

SOA **não** é um produto

SOA **não** é um projeto de TI

SOA **não** é um software

SOA **não** é um “framework”

SOA **não** é uma metodologia

SOA **não** é uma solução de negócio

SOA **não** é um middleware

SOA **não** é um padrão (norma)

SOA **não** é um serviço

SOA **não** é uma ferramenta de produtividade





# SOA

- Confusão com o termo  
“*Service oriented computing*”



*“SOA é, basicamente, um modelo de arquitetura de software que beneficia a eficiência, agilidade e produtividade no desenvolvimento de aplicações e está alinhado com os objetivos de “service oriented computing” Thomas Erl*

# O que é SOA?

- Estilo de arquitetura onde as funcionalidades de aplicações existentes são disponibilizadas na forma de serviços

# O que são serviços

- Serviço é um componente que atende a uma função de negócio (business function). Ele pode receber e responder requisições ocultando os detalhes de sua implementação.
  - Desacoplados em relação ao cliente/consumidor
  - Descritos através de contratos de operações

# Como Projetar os Serviços?

# Princípios de Design de Serviços

Para fornecer o valor para o negócio, os serviços tem conjunto de critérios. Estes critérios são atributos ou características:

**Serviços são Reutilizáveis**  
**Serviços compartilham um Contrato formal**  
**Serviços possuem um Baixo Acoplamento**  
**Serviços Abstraem a lógica**  
**Serviços são capazes de se Comporem**  
**Serviços são Autônomos**  
**Serviços evitam Alocação de Recursos por longos períodos**  
**Serviços são capazes de serem Descobertos**

Fonte: Thomas Erl



# Princípios de Design: Baixo Acoplamento

## Baixo acoplamento (Loose Coupling)

### Como minimizar dependências e maximizar o reuso?

O acoplamento é uma medida de quão fortemente um serviço está conectada, possui conhecimento ou depende de outro serviço e/ou componente.

Com fraco acoplamento, um serviço não é dependente de outros serviços e/ou componentes

### Um serviço que possui forte acoplamento, temos os seguintes problemas:

- Mudanças em um serviço força mudanças em outros serviços e/ou componentes
- O serviço é mais difícil de ser reusado, já que depende da presença de outros serviços e/ou componentes

### Solução:

- > Desenhar e modelar serviços com **baixo acoplamento**.
- > Contudo o baixo acoplamento leva complexidade de implementação

Acoplamento refere-se a dependência ou relacionamento entre duas coisas. A medida do acoplamento é comparada ao nível de dependência.

- **Forte:** Forte grau dependência ... Exemplo: Significa que o um serviço depende de uma determinada tecnologia para ser implementado, por exemplo: Web Services.
- **Fraco:** Fraco grau de dependência fraco. Exemplo: Significa que um serviço não depende diretamente de uma determinada tecnologia. O Serviço tem sua implementação encapsulada.

O SOA, requer que os serviços tenham **acoplamento fraco**, pois isto facilita a manutenção e favorece o reuso

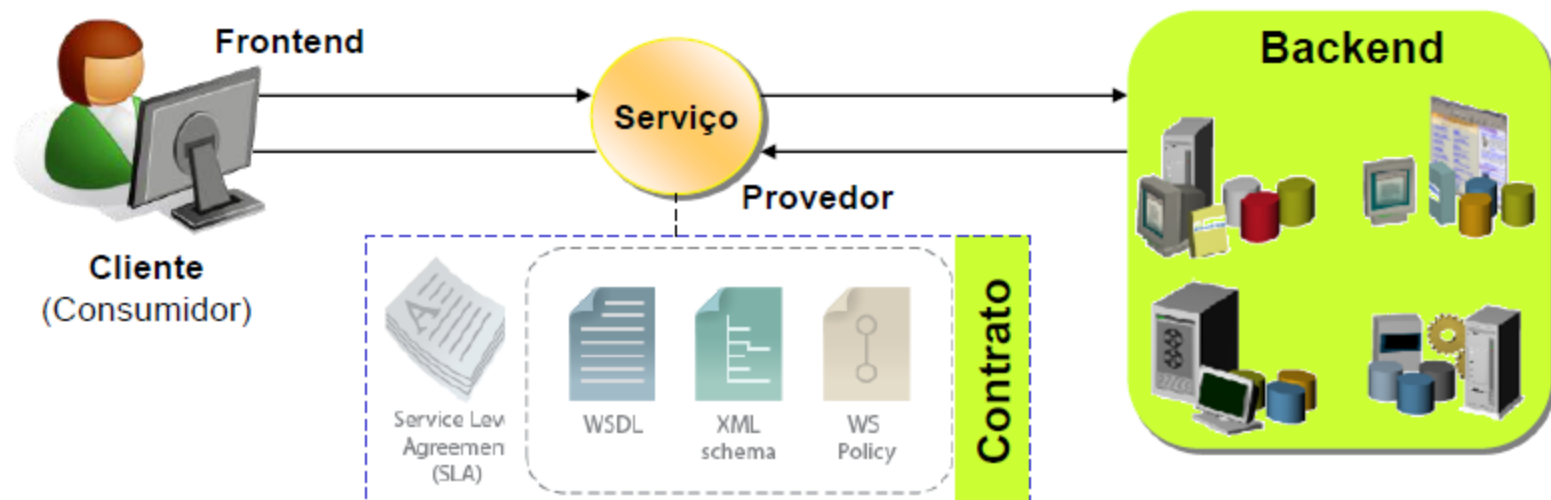
# Princípios de Design: Contratos

## Contrato bem definido (Well-defined service contracts) ou Interface bem Definida

Serviços expressam seu objetivo e as capacidades através de um Contrato de Serviço.

A descrição completa de uma interface (pública) de serviço entre o consumidor (cliente) e o provedor de serviço. O contrato contém a interface técnica (assinatura), as semânticas e os aspectos não funcionais (atributos de qualidade, como: disponibilidade, segurança, performance, confiabilidade e etc) com o Acordo de Nível de Serviço (SLA).

Devemos garantir que os contratos de serviços sejam bem definidos, concisos, otimizados, devidamente granular, padronizado (para garantir que os parâmetros estabelecidos pelos serviços são coerentes), confiáveis e governáveis.



O Contrato descreve as funcionalidades, tipos de dados, modelo dados definidos e como as políticas são definidas e efetivadas.

# Princípios de Design: Granularidade

## Granularidade

**Granularidade** nada mais é do que o nível de detalhe de cada serviço ou componente, existem dois tipos de granularidade:

- **Grossa:** Quanto menor for o nível de detalhes, maior ou mais grossa será o nível de granularidade, ou seja, sem detalhes.
- **Fina:** Quanto maior for o nível de detalhes, menor ou mais fina será o nível de granularidade, ou seja, com detalhes

O SOA, que é arquitetura orientada a serviços, requer que os serviços tenham (obrigatoriamente) o nível de granularidade grossa (sem detalhes). **O nível de granularidade grossa favorece o reúso.**

### Exemplos:

Exemplo de Serviço com **Granularidade Grossa**:

- Confirmar Pedido
- Confirmar Reserva
- Verificar disponibilidade de voo
- Realizar venda de produtos
- Criar Plano Tarifário
- Validar uma transação de pagamento (cartão de crédito)



Exemplos **incorretos** (no contexto SOA) de Serviços com **Granularidade Fina**:

- Alterar o Número do Pedido
  - Alterar o Código do Produto
  - Fazer crédito da conta-corrente do cliente
-



# Exemplo de projeto Software + Serviços

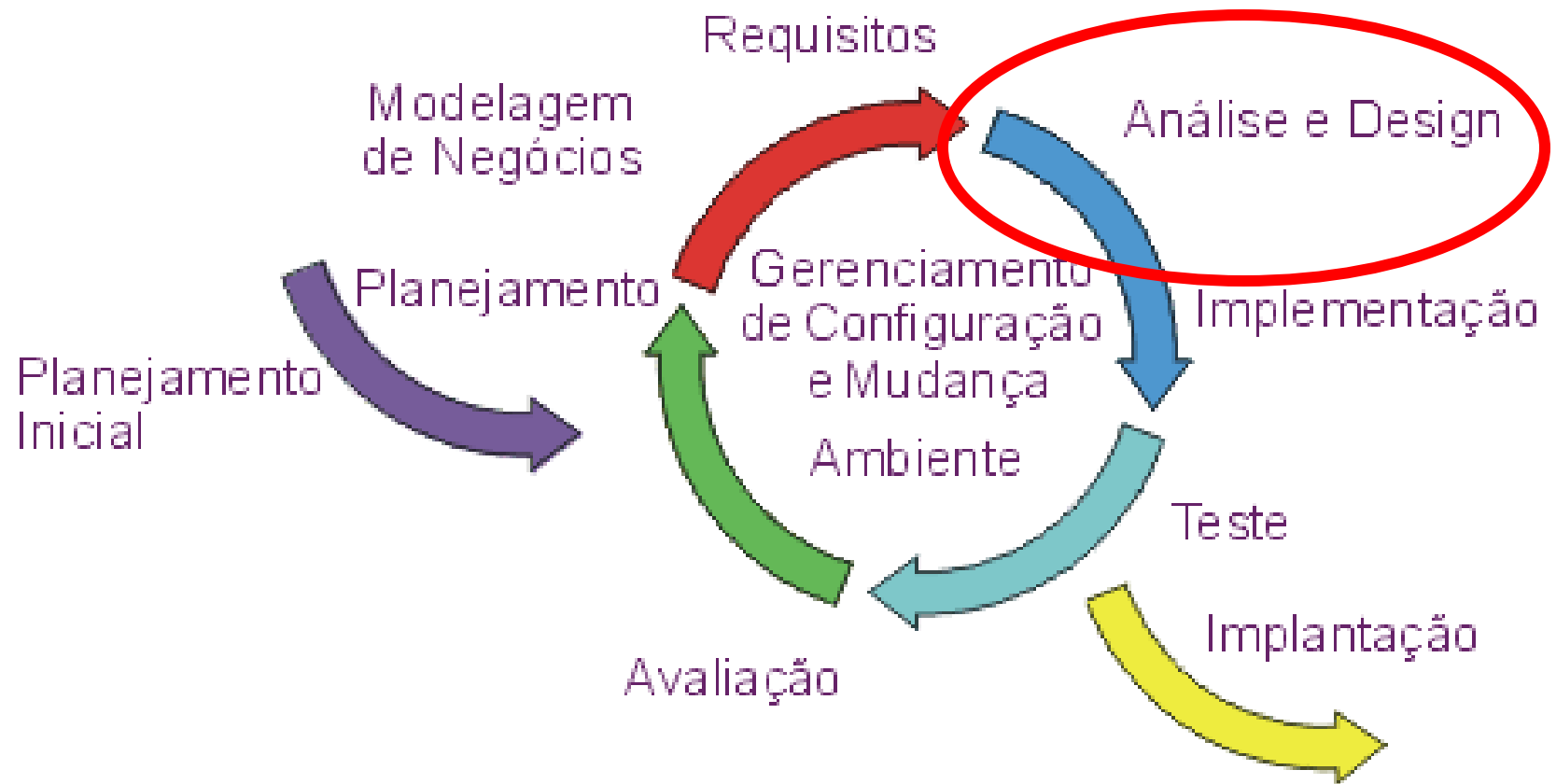
## *Marketplace de imagens na nuvem*



O que faremos?

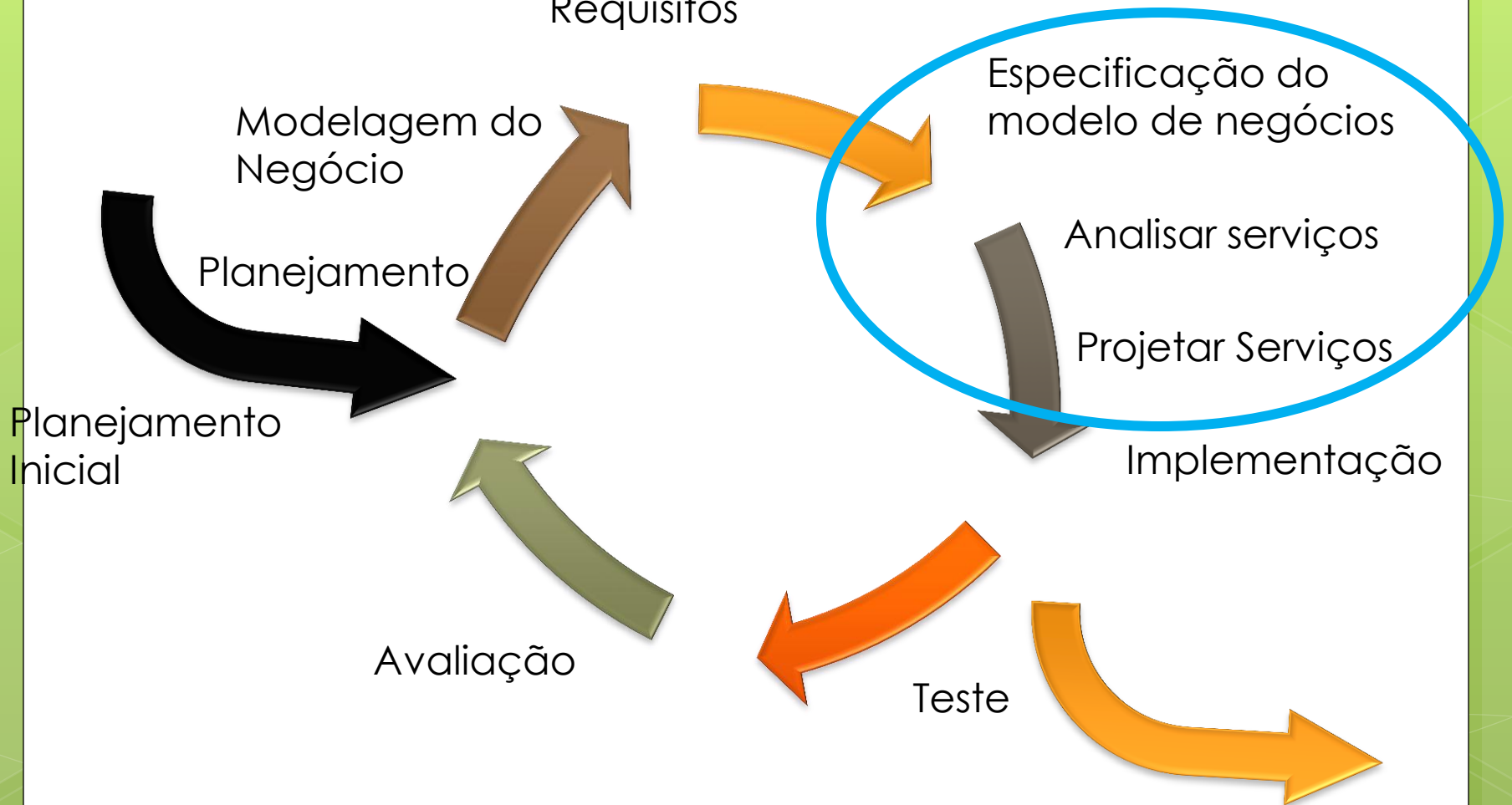
# Uma Abordagem de Desenvolvimento de Software Dirigida a Modelos e Orientada a Serviços

# Visão geral (Exemplo: RUP)



# Visão geral

Requisitos



# Referências

- The Standish Group. Disponível em <http://www.standishgroup.com/>, maio 2009.
- CHAOS Summary 2009. Disponível em [http://www.standishgroup.com/newsroom/chaos\\_2009.php](http://www.standishgroup.com/newsroom/chaos_2009.php) , maio 2009.
- George Stepanek. Software Project Secrets: Why Software Projects Fail. Apress, 2005
- From Stakeholders to Models: It Is All a Matter of Viewpoints. Disponível em [http://msdn.microsoft.com/pt-br/library/bb447667\(en-us\).aspx](http://msdn.microsoft.com/pt-br/library/bb447667(en-us).aspx), maio 2009.
- Proof-of-Concept Design. Disponível em <http://msdn.microsoft.com/en-us/library/cc168618.aspx>, maio 2009.

- Kent, S. Model Driven Engineering. [IFM 2002](#): 286-298, 2002
- France, R. and Rumpe, B. Model-driven Development of Complex Software: A Research Roadmap. International Conference on Software Engineering, 2007.
- Mattsson, A. Lundell, B. Lings, B. and Fitzgerald, B. Linking Model-Driven Development and Software Architecture: A Case Study. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 35, NO. 1, 2009.
- L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, second ed. Addison-Wesley, 2003.



- B. Hailpern and P. Tarr, "Model-Driven Development: The Good, the Bad, and the Ugly," IBM Systems J., vol. 45, pp. 451-461, 2006.
- P. Baker, L. Shiou, and F. Weil, Model-Driven Engineering in a Large Industrial Context—Motorola Case Study," Proc. Eighth Int'l Conf. Model Driven Eng. Languages and Systems, pp. 476-491, 2005.
- Staron, M. Adopting Model Driven Software Development in Industry—A Case Study at Two Companies. Ninth Int'l Conf. Model Driven Eng. Languages and Systems, pp. 57-72, 2006

- Pfadenhauer, K. Dustdar, S. Kittl, B. Challenges and Solutions for Model Driven Web Service Composition. 4th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005.
- Castro, v. Mesa J. M. V. Herrmann, E. Marcos, E. A Model Driven Approach for the Alignment of Business and Information Systems Models. Mexican International Conference on Computer Science, 2008.

- Budgen D, Brereton P. Performing Systematic Literature Reviews in Software Engineering. *Writing*. 2006:1051-1052.
- Castro VD, Mesa JM, Herrmann E, Marcos E. A Model Driven Approach for the Alignment of Business and Information Systems Models. *2008 Mexican International Conference on Computer Science*. 2008:33-43. Available at:  
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4653237>.
- Cochinwala M, Shim H, Wullert J. A model-driven approach to rapid service introduction. *2005 9th IFIP/IEEE International Symposium on Integrated Network Management, 2005. IM 2005*. 2005:659-672. Available at:  
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1440838>.

- The Pragmatics of Model-Driven Development. *Ieee Software*. 2003:19-25.
- Exertier D, France T. MDSOFA: A Model-Driven Software Factory. *Technology*.:1-10.
- Fritzsche M, Gilani W, Spence I, et al. Towards Performance Related Decision Support for Model Driven Engineering of Enterprise SOA applications. 2008:57-65.
- Goknil A. Survey of Traceability Approaches in Model-Driven Engineering. 2007:313-324.
- Leaders M, Every S. The State Of Model-Driven Development. *Reproduction*. 2007.

- Lenoir J, Programme SF. MDD White Paper. *Group*. 2006;(January):1-11.
- Llorente T, Corp M. Model-Driven SOA with “Oslo”. *Complexity*. 2009;(September).
- What Models Mean. *Ieee Software*. 2003:26-32.
- Nikulsins V, Nikiforova O. Adapting Software Development Process towards the Model Driven Architecture. *2008 The Third International Conference on Software Engineering Advances*. 2008:394-399. Available at:  
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4668137>.

- Pastor O, Carlos Molina J. *Model-Driven Architecture in Practice*.
- Presentation I. Broadening the use of MDA ©. *Practice*. 2006;(May).
- Roald Kruit (Mendix Co-founder & CTO) DR. *Modeling the Agile Enterprise: From Software Engineering to Business Engineering. Development*.
- Salicki S, Research T. How MODELWARE contributes to Eclipse. *System*. 2006:1-9.

- ◉ Schmidt DC. Model- Driven Engineering. *Computer*. 2006;(February):25-31.
- ◉ Solheim I, Technology C. New Roles in Model-Driven Development. *Challenges*.
- ◉ Streitferdt D, Wendt G, Nenninger P, Nyßen A, Lichter H. Model Driven Development Challenges in the Automation Domain. *2008 32nd Annual IEEE International Computer Software and Applications Conference*. 2008:1372-1375. Available at:  
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4591784>.
- ◉ . Weigert T, Weil F. Practical Experiences in Using Model-Driven Engineering to Develop Trustworthy Computing Systems. *Proceedings of the IEEE*. 2006:0-7.

# Referências

## Padrões:



<http://www.w3.org/>



<http://www.oasis-open.org/>



<http://www.ws-i.org/>



Object Management Group  
<http://www.omg.org/corba/>



Object Management Group  
<http://www.omg.org/uml>



<http://www.ebxml.org/>



<http://www.omg.org/>



<http://www.bpmn.org/>

<http://www.omg.org/mda/>



<http://www.opengroup.org/projects/soa/>



<http://www.unicode.org>

## Referências:

CIO Executive Day - 2006  
Criando Valor para o Negócio com SOA.  
BEA Systems

SOA Principles of Service Design  
Thomas Erl  
Editora: Prentice Hall

Service-Oriented Architecture  
Concepts, Technology and Design  
Thomas Erl  
Editora: Prentice Hall

SOA in Practice  
Nicolai M. Josutis  
Editora: O'Reilly

Service-Oriented Architecture  
A Planning and Guide Implementation Guide  
for Business and Technology  
Eric A. Marks & Michael Bell  
Editora: Wiley

## Promoção do SOA:



<http://www.soa-consortium.org>

Revista Info Corporate:  
[www.infocorporate.com.br](http://www.infocorporate.com.br)

IBM:  
[www.ibm.com/soa](http://www.ibm.com/soa)