



INSTITUTO DE GESTÃO EM TECNOLOGIA DA INFORMAÇÃO

IGTI

## Visão Geral de Arquitetura de Software

Marco Aurélio S. Mendes

[WWW.INSTITUTOGTI.COM.BR](http://WWW.INSTITUTOGTI.COM.BR)

## ✓ Tópicos da Apresentação:

- A importância da Arquitetura de Software no desenvolvimento de produtos.
- Atividades de Arquitetura de Software.
- O Papel do Arquiteto de Software.

### Problemas na venda online dos ingressos para U2

16/01/2006 11:30 - Redação

### **[PSL-GO] Portal da Caixa não funciona com Firefox**

Hugo Do Nascimento [hadnas@gmail.com](mailto:hadnas@gmail.com)

*Segunda Setembro 24 21:29:19 BRT 2007*

Comprar passagens no site da Gol:  
missão impossível? Março 24, 2006

---

- ✓ Uma equipe de TI em Belo Horizonte recebeu a incumbência de desenvolver um novo sistema Java EE para implementar um sistema de força de vendas para uma empresa que fatura mais de 1 bilhão de reais anualmente.
- ✓ A arquitetura foi definida com:
  - Um arquiteto alocado.
  - Frameworks Struts e Hibernate.
  - Diversos padrões de desenho.
  - Um ambiente de produtividade baseado em uma famosa IDE de mercado.
  - Modelos UML
- ✓ Ainda assim, algo não funcionou. O projeto foi cancelado com grande prejuízo para a empresa de TI prestadora do serviço.

O que fiz de errado?



✓ A equipe não considerou diversas premissas fundamentais para o sucesso arquitetural do projeto:

- Pressões extremas de tempo.
- Riscos técnicos da camada de persistência, que envolvia múltiplos bancos de dados.
- Pessoas e relacionamentos sinérgicos.
- Geração de valor para os interessados no projeto.
- Tecnologias e middlewares não implicam em arquitetura.
- Desconhecimento sobre o que significa uma “arquitetura de software”.

O que fiz de errado?



Arquitetura são modelos e papel.

<Minha tecnologia favorita> é arquitetura.

Trabalho unicamente do arquiteto de software.

Arquitetura é uma ciência exata.

Arquitetura é uma arte.

Arquiteto da “Torre de Marfim”



### Caso 1

- Uma equipe de TI no governo federal descobriu na semana da entrega que o envio de email não funcionava devido a topologia física desconhecida e restrições de segurança (Certificados digitais e HTTPS)

### Caso 2

- Uma outra equipe teve que refazer toda a camada visual de JavaScripts ao “descobrir” em produção que o sistema não atendia o Firefox.

### Caso 3

- Uma terceira equipe de uma grande empresa de Minas descobriu em produção que o tempo médio dos casos de uso era de 40 segundos.

### Caso 4

- Uma equipe descobriu que a aplicação travava o banco de dados em produção periodicamente (deadlocks) e parava toda uma universidade.



Todo sistema em produção possui uma arquitetura de software.

- Se você não investiu tempo e cuidado para construí-la, ela pode ser muito diferente do que você esperava!

**O edifício Palace 2 é implodido três dias depois de desabamento parcial**



## ✓ Algumas definições de arquitetura:

*“A arquitetura de um software é a estrutura ou estruturas do sistema, o que compreende componentes de software, propriedades desses componentes que são visíveis externamente e o relacionamento entre eles”, Paul Clements, SEI.*

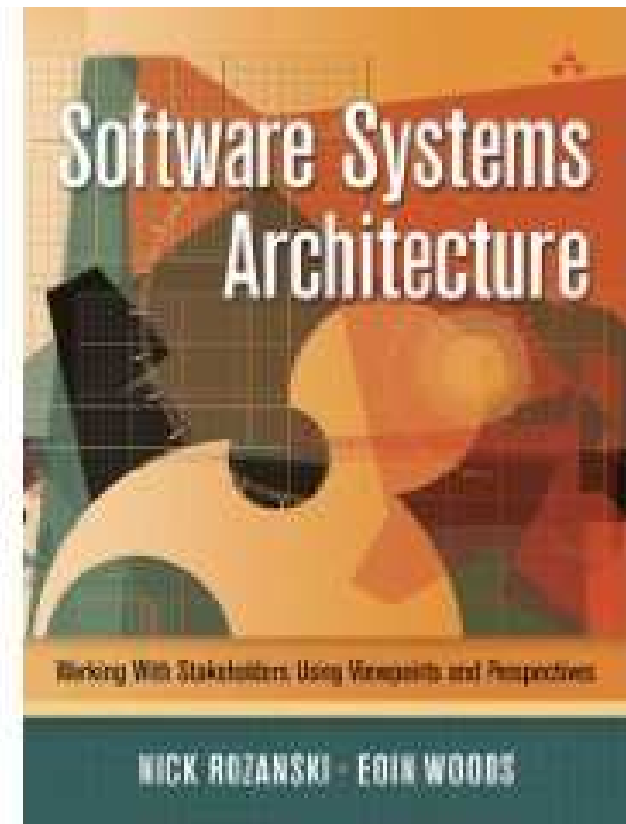
“A arquitetura de um sistema de software compreende um conjunto de componentes, conexões e restrições de sistema e de software; um conjunto de necessidades de stakeholders; uma lógica que demonstra que se os componentes, conexões e restrições definem um sistema que se implementando irá atender as necessidades dos stakeholders”, Barry Boehm.

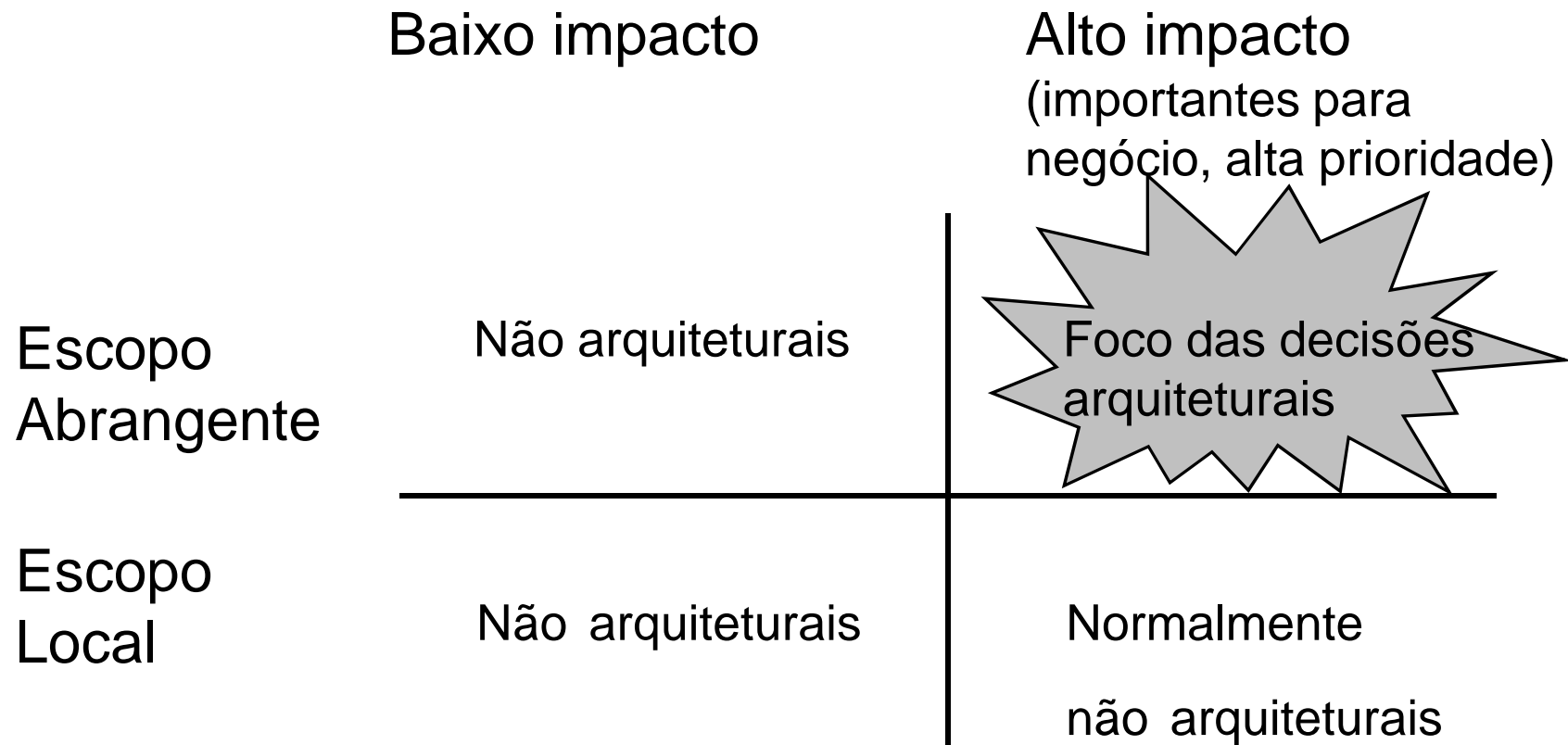
## ✓ Algumas definições de arquitetura:

“A Arquitetura de Software é a organização fundamental de um sistema, incluindo seus componentes, o relacionamento entre esses componentes e com o ambiente e os princípios que definem o desenho e a evolução dos componentes.”,  
IEEE 1471/2000 Recommended Practice for Architectural Description of Software-Intensive Systems

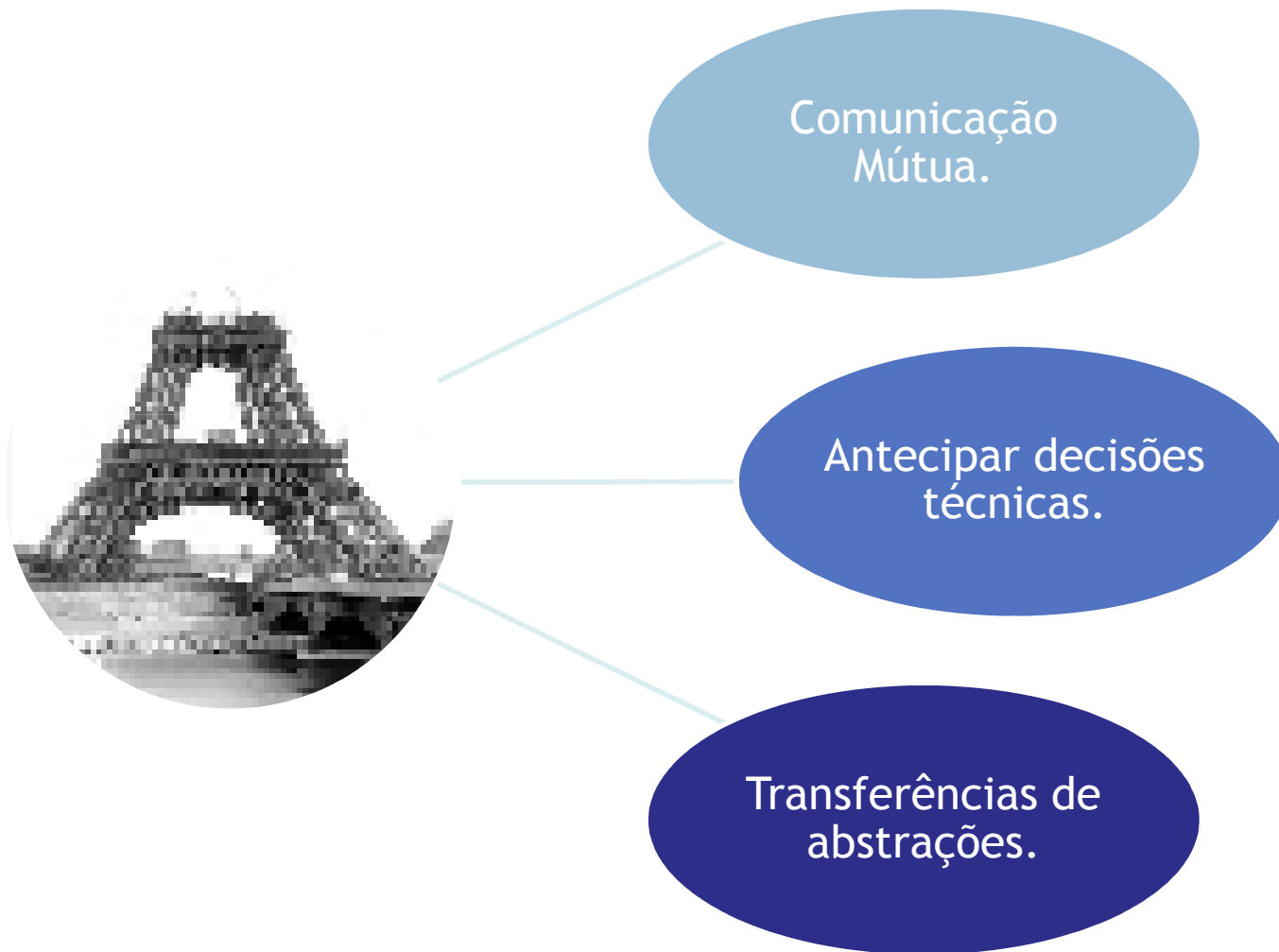
“A Arquitetura de Software inclui o conjunto de decisões significantes sobre a organização de um software tais como a seleção dos elementos estruturais e suas interfaces; o comportamento entre esses elementos; a composição destes elementos estruturais e de comportamento em subsistemas maiores e o estilo arquitetural que guia esta organização.”, Booch, Kruchten, Reitman, Bittner, and Shaw

“Arquitetura de software é o conjunto de decisões que, se feitas incorretamente, podem causar o cancelamento do projeto.” – Eoin Woods

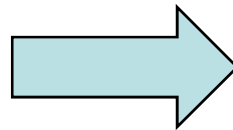




# Importância da Arquitetura de Software



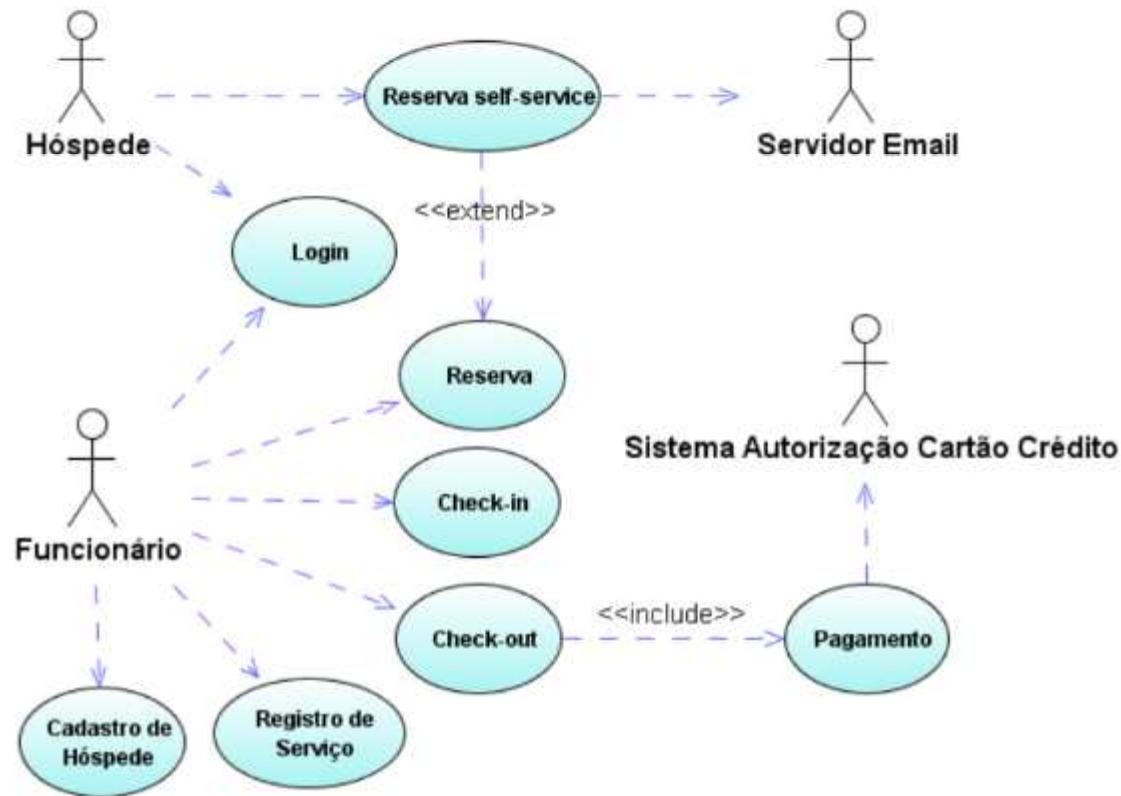
- ✓ Nosso exemplo: Sistema para evolução da rede de hotéis ACME, que ganhou um financiamento do BNDES.



- ✓ Um processo didático e simplificado:
  1. Alinhar o projeto com a visão, objetivos, requisitos e restrições dos interessados.
  2. Desenvolver os requisitos arquiteturalmente significativos
  3. Identificar e atacar riscos e restrições técnicas.
  4. Modelar a arquitetura.
  5. Construir fisicamente a arquitetura.
  6. Avaliar a arquitetura.

- ❑ 01. Alinhar o projeto com os princípios, visão, objetivos, requisitos e restrições dos interessados.
  - Visão: Suportar o crescimento do hotel ACME e a automação dos principais processos de negócio que envolvem clientes e hóspedes.
  - Princípios: Simplicidade, flexibilidade, alinhamento com negócio.
  - Requisitos Funcionais:
    - Reserva self-service
    - Reserva
    - Pagamento
    - Login
    - Check-in
    - Check-out
    - Cadastro de hóspede
    - Registro de serviços

## 02. Desenvolver os requisitos arquiteturalmente significativos



### 02. Desenvolver os requisitos arquiteturalmente significativos

- Requisitos de qualidade/não-funcionais.
  - O hóspede pode fazer sua reserva remotamente (WWW) e de forma segura.
  - Pico de 1000 acessos simultâneos em alta temporada.
  - Tempo de resposta para reserva pelo hóspede: 6 seg.
  - Tempo de resposta para reserva por funcionário: 2 seg.
  - A interface para o funcionário deve ser rica (gráfica, *drag-n-drop* , Ajax) e acessível pela intranet.
- Requisitos de infra-estrutura
  - Disponibilidade do sistema de 99,9%.
  - Estimativa de crescimento de 20% ao ano.

## 02. Desenvolver os requisitos arquiteturalmente significativos

- ❑ Classificando requisitos de qualidade.

Requisito de Qualidade	Classificação
Reserva remota segura.	Segurança
Usuários identificados.	Segurança
Pico de 1000 acessos simultâneos	Performance
Confirmação de reserva por email.	Interoperabilidade
Tempo de resposta reserva hóspede 8 seg.	Performance
Tempo de resposta reserva funcionário 5 seg.	Performance
A interface para o funcionário deve ser rica.	Usabilidade
Disponibilidade do sistema de 99,9%.	Disponibilidade
Estimativa de crescimento de 20% ao ano.	Escalabilidade

## 02. Desenvolver os requisitos arquiteturalmente significativos

- ❑ Classificando requisitos funcionais.

Ordem	Caso de Uso	Requisitos Suplementares
1	Reserva self-service	Segurança, Performance, Interoperabilidade, Disponibilidade
2	Reserva	Usabilidade
3	Pagamento	Interoperabilidade, Segurança

### 03. Identificar e atacar riscos e restrições técnicas.

- O sistema de autorização de cartões de crédito (SACC) é provido por terceiros e as transações serão realizadas através de webservices.
- O servidor de aplicações deve ser o Sun GlassFish.
- O SGBD deve ser o Oracle 10G.
- A aplicação web deve ser compatível com IE 6 e Firefox 2 ou versões superiores.
- Principais riscos identificados: Apesar de ser compatível com Java EE, o servidor de aplicações não é conhecido pelos desenvolvedores.

### 03. Identificar e atacar riscos e restrições técnicas.

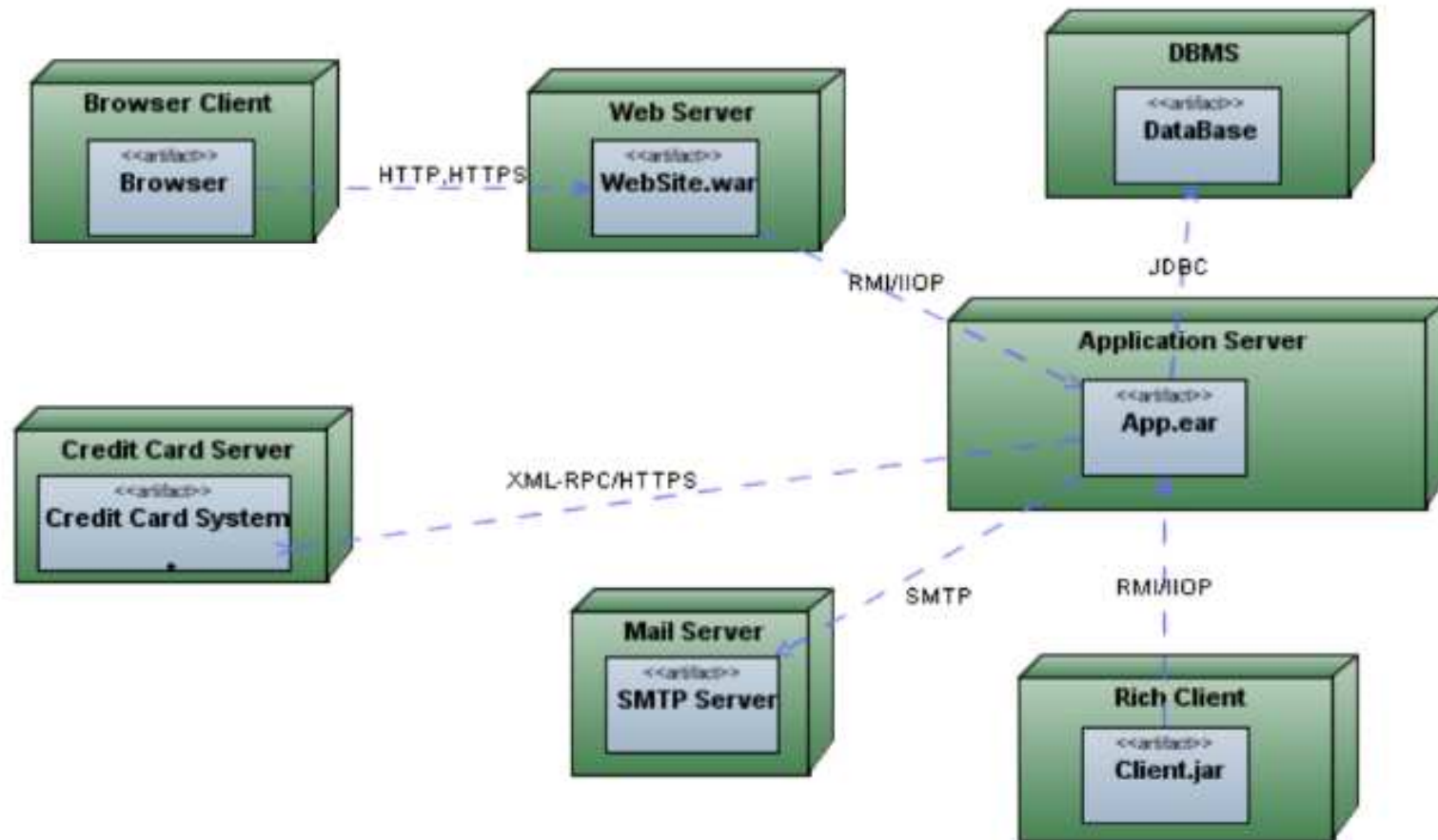
- Ações de mitigação de riscos e aos requisitos de qualidade mais severos (POC - Provas de Conceito)

Risco ou Requisito severo	Mitigação
Performance	Teste de stress sobre UC Reserva self-service
Disponibilidade	Teste de maturidade
Disponibilidade email	POC Email – Componentes assíncronos
Interoperabilidade	POC Integração com Sistema de Cartão de Crédito
Escalabilidade	POC Cluster
Avaliação servidor de aplicação	POC Avaliação, Capacitação da Equipe

## 04. Modelar a Arquitetura

- Explorar e avaliar opções arquiteturais de alto-nível.
- Prover um entendimento da estrutura para os patrocinadores e demais interessados.
- Levar em consideração:
  - Desenho da rede pré-existente.
  - Bancos de dados pré-existentes.
  - Ambiente web.
  - Configuração dos servidores.
  - Uso de padrões.
- Exemplo: Visão 4+1 (Kruchten/RUP).

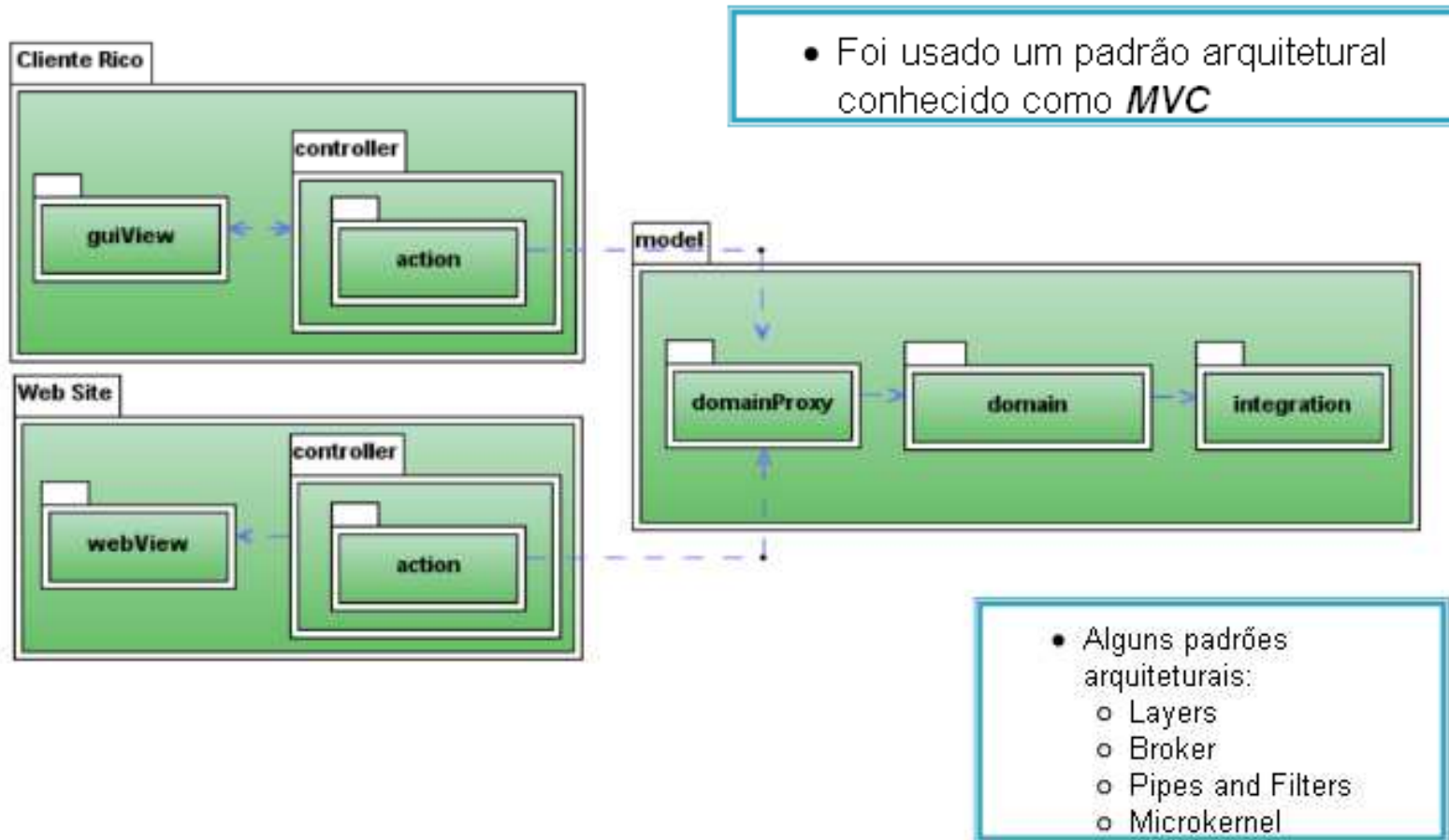
## 04. Modelar a Arquitetura



### 04. Modelar a Arquitetura

- Criar uma estrutura inicial para o modelo de design.
- Mostrar pacotes de desenho arquiteturalmente significativos.
- Aplicar padrões de desenho gentilmente.

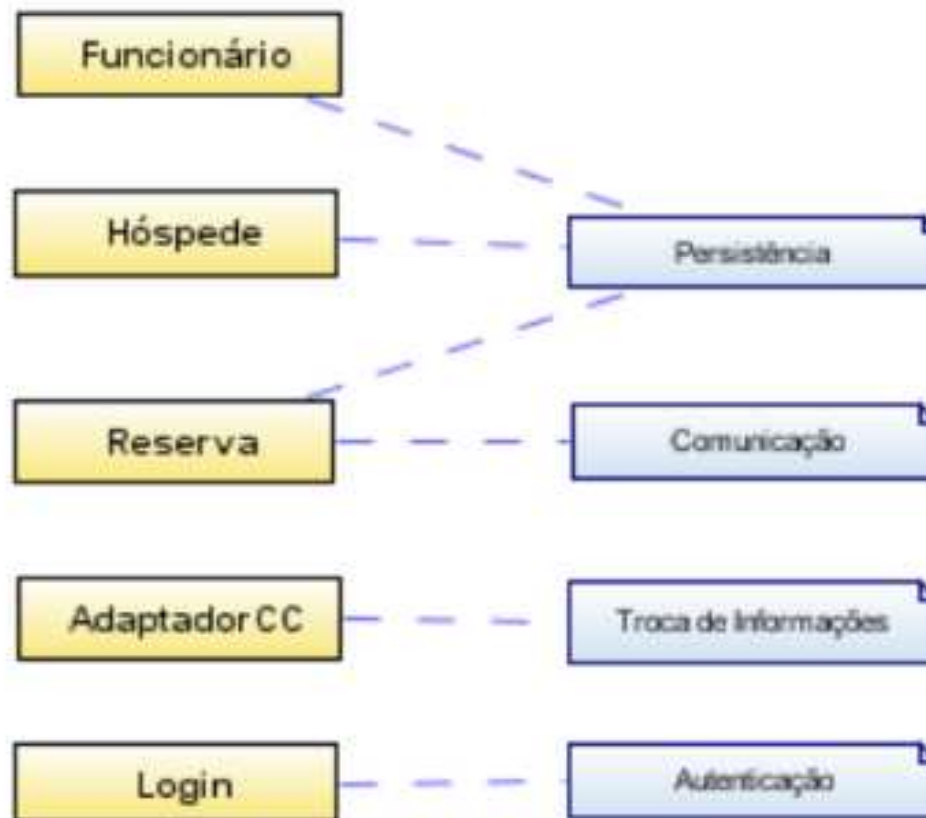
## 04. Modelar a Arquitetura



### 04. Modelar a Arquitetura

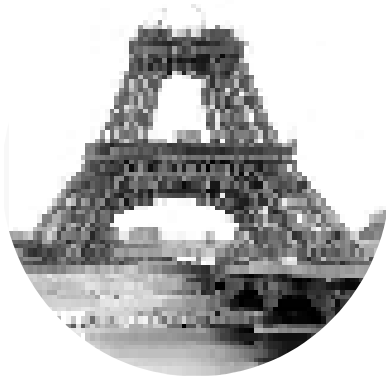
- *Mecanismos* de análise e desenho.
- Decisões táticas que irão realizar a arquitetura.
- Alguns mecanismos de análise:
  - Persistência.
  - Segurança.
  - Gerência de transações.
  - Troca de informações/Interoperabilidade.

### 04. Modelar a Arquitetura



## 05. Construir fisicamente a arquitetura.

- Time (arquitetos, projetistas e desenvolvedores) escolhem cenários de risco e constroem juntos a arquitetura executável do sistema.
- Em processos como o UP, entre 5 a 20% dos requisitos mais prioritários e complexos são endereçados até o 3/10 do tempo do projeto.

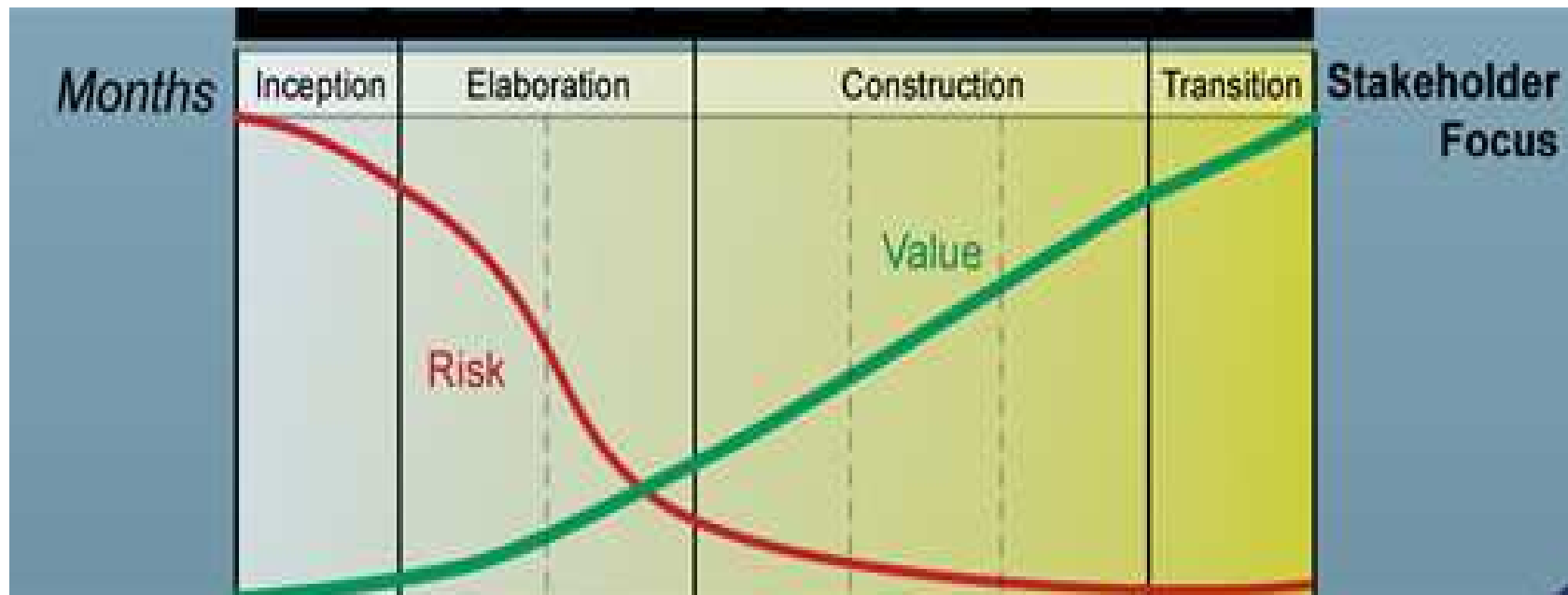


## 05. Construir fisicamente a arquitetura.

- Objetivo: Reduzir os riscos do projeto.

1/10 do tempo do projeto.  
Arquitetura candidata

3/10 do tempo do projeto.  
Arquitetura executável



### ✓ 06. Avaliar a Arquitetura

- O objetivo é rever o resultado e analisar alternativas. Avaliação do grau de atendimento dos requisitos de qualidade.
- Testabilidade da arquitetura.
- Utilização de checklists para validação.
- Métodos clássicos: ATAM, CBAM, SAAM e ARID (SEI / Carnegie Mellon).
- Ferramentas de análise estrutural e arquitetural podem auxiliar na avaliação arquitetura e conformidade do código.
  - Rational Software Architect, SA4J, Metrics (plugin do Eclipse), SourceMonitor, Jdepend.

Caso real acontecido em 2008, para o desenvolvimento de uma nova plataforma de telefonia móvel para uma grande operadora de TELECOM do Brasil.

- Pressões extremas de prazo.
- Grande complexidade técnica (mediação, adaptadores, conectores, protocolos, messageria, servidores de aplicação desconhecidos e complexos).

Resultado: Grande caso de sucesso!

*O que você acha que aconteceu para que a equipe tivesse sucesso no projeto?*

Gerência reconheceu a importância da solução técnica para o projeto e um **time técnico** foi estabelecido.

- Arquiteto alocado em tempo integral.
- Time técnico de desenvolvedores de apoio.
- Estabelecimento de **confiança** pelo arquiteto:
  - **Caráter:** Franqueza, preocupação sincera, transparência, fazer dos erros acertos, lealdade.
  - **Competência:** Resultados, Melhoria contínua, Enfrentar a realidade, responsabilidade.
  - **Caráter e competência:** Ouvir primeiro, honrar acordos, estender a confiança.

**Sinergia entre time técnico, analistas de requisitos e clientes.**

- **Captura e desenvolvimento dos requisitos arquiteturalmente significativos:**
  - Visão, requisitos de negócio.
  - Requisitos não-funcionais. Uso dos métodos FURPS+ e ISO 9126
  - Requisitos de infra-estrutura.
- **Foco do ataque (requisitos prioritários e de risco técnico).**

## Modelagem arquitetural

- Modelagem com o método da “**Visão 4+1**”, de Philippe Kruchten/RUP.
  - Visão de casos de uso.
  - Visão lógica.
  - Visão de implementação.
  - Visão de implantação.
- Exploração de **mecanismos arquiteturais** tais como persistência, interoperabilidade, segurança, mediação.
- **Documentação** em um artefato RUP chamado SAD (Software Architecture Document) e **comunicação contínua** das decisões para o time.

## Construção incremental da arquitetura

- Realizado em paralelo com a modelagem para ataque aos mecanismos de maior risco.
- Geração de **código executável** que acomodava cenários críticos de casos de uso reais - Arquitetura executável.
- Forte **envolvimento com os clientes** para validação de códigos Alfas.
- Geração de **valor perceptível** para a gerência do projeto e clientes.
  - Aspecto mais importante para manter controlada a ansiedade do cliente.
- Trabalho coletivo - **Propriedade coletiva**.

## Validação e Verificação da arquitetura

- Interação contínua com time de testes para **testabilidade da arquitetura**.
- Testes de unidade, testes de desempenho, stress, maturidade, segurança, entre outros, realizados a quatro mãos (Time de Arquitetura + Time de Testes).

## Acompanhamento do Time

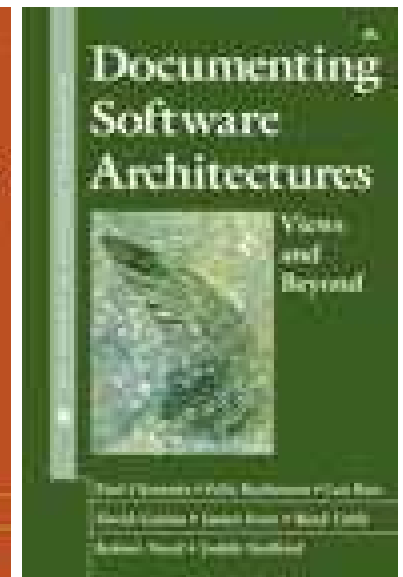
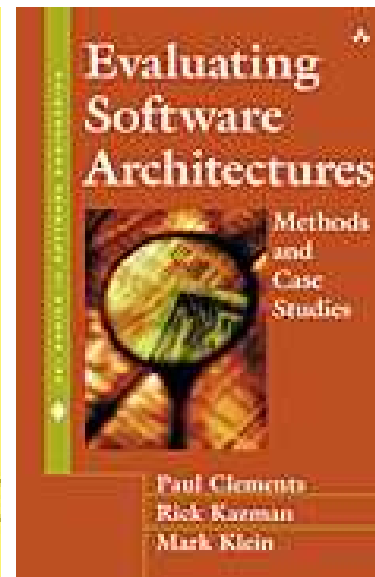
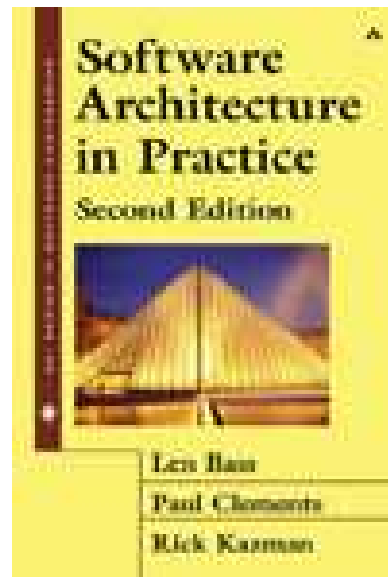
- Após a redução do risco técnico, arquiteto **acompanha o time** nas fases de construção e transição para garantir a evolução da arquitetura.

Competência Caráter POC  
Testabilidade Mitigação de Riscos  
Métodos Requisitos Arquiteturais  
4+1 Executáveis Propriedade Coletiva  
Visão Arquitetural Time Técnico  
Arquiteto de Software  
Comunicação Processos

- ✓ Existem outras arquiteturas além da arquitetura de software.



- ❑ DeArquitetura
  - ❑ (<http://dearchitettura.wordpress.com>)
- ❑ PANGEA (<http://pangeanet.org>)
  - Rede social de arquitetura de software.
- ✓ Livros diversos:



- ✓ Curso lato-sensu “Estratégia em Arquiteturas de Sistemas” do IGTI.
- ✓ Arquitetura de Software - SEI:
  - <http://www.sei.cmu.edu/architecture/>
  - Currículo de Arquitetura de Software:
    - [http://www.sei.cmu.edu/architecture/arch\\_curriculum.html](http://www.sei.cmu.edu/architecture/arch_curriculum.html)
  - Papers diversos:
    - [http://www.sei.cmu.edu/architecture/pub\\_by\\_topic.html](http://www.sei.cmu.edu/architecture/pub_by_topic.html)
- ✓ Agile Architectures, Scott Ambler
- ✓ *The “4+1” View Model of Software Architecture*, Philippe Kruchten

O arquiteto observa o mundo.  
Mas confia em sua visão interior.  
Ele permite que as coisas venham e vão.  
Seu coração é aberto como o céu.

O arquiteto não fala, age.  
Quando o trabalho está pronto,  
o time diz: "Incrível:  
nós fizemos tudo sozinhos!"

Quando o arquiteto lidera, o time  
dificilmente percebe que ele existe.  
O segundo melhor é o arquiteto que é  
amado. Depois, o que é temido.  
O pior deles é aquele desprezado.

© The Tao of the Software Architect  
Lao-Tsu, revisited by  
Philippe Kruchten