

# The Architecture Tradeoff Analysis Method

Júlio César - Reuso de Software – DCC/UFMG

# Autores

+ Rick Kazman, Mark Klein, Mario Barbacci, Tom Longstaff,  
Howard Lipson, Jeromy Carriere

Software Engineering Institute

Carnegie Mellon University

Pittsburgh, PA 15213

# Architecture Tradeoff Analysis

- + Todo Software tem uma Arquitetura
  - + Atributos de Qualidades
    - + Disponibilidade impacta segurança.
    - + Segurança afeta desempenho .
  - + Objetivos
    - + Identificar pontos de troca entre os atributos.
    - + Facilitar a comunicação entre os envolvidos.
    - + Esclarecer e refinar requisitos não funcionais.
    - + Fornecer um quadro para o processo de analise.

# Porque Usar ?

+ **Arquitetura boa** nem sempre...

+ **Arquitetura ruim** dificilmente...



+ **Decisões Arquiteturais**

+ Preocupações estratégicas de negócios.

+ Atendendo às limitações de custo e cronograma.

+ Utilizando o pessoal disponível.

“...cada atributo de qualidade tem ligações com outros atributos , por meio de elementos arquiteturais específicos.”

# O ATAM

## + Modelo Espiral

- + Onde em cada iteração tem maior compreensão e redução dos riscos.

Cenário e Requisitos  
Pontos de vista e realização cenário  
Construção de modelos  
Análise e compensações.

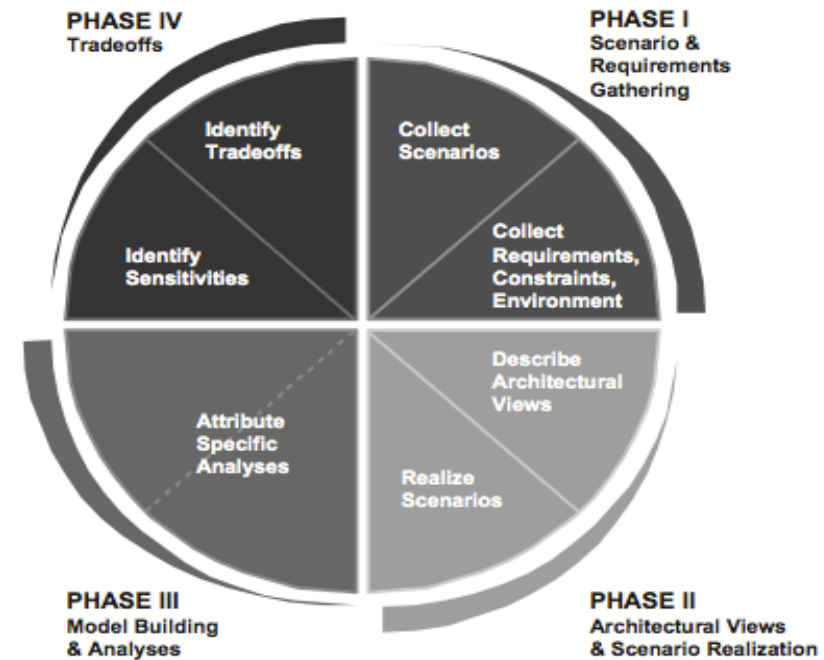


Figure 1. Steps of the Architecture Tradeoff Analysis Method

# O ATAM

## + Passo 1 - Colete Cenários

- + Provoque o Cenário.
- + Coleta de requisitos.
- + Levante as restrições ambientais.
- + Identifique os atores.

## + Passo 2 - Coletar os requisitos / restrições / Ambiente

- + Desempenho e Segurança .
- + Como eles evoluem ?
- + Revisitar passo anterior

# O ATAM

## + Passo 3 - Descreva Visões de Arquitetura

### + Arquiteturas candidatas

- + Os dados necessários para uma análise de cada um desses atributos de qualidades é normalmente capturada em distintas visões arquiteturais .
- + Vários pontos de vista
  - + Modular, Processo, Fluxo, Classe.



**Cada desenho arquitetural é analisado separado.**

# O ATAM

- + Passo 4 – Análises dos atributo específicos
  - + Pequeno conjunto de arquiteturas já proposto.
  - + Analisar cada atributo isoladamente .
  - + Separação de interesses e Recursos (especialistas).
- + Os resultados são valores:
  - + Os pedidos são respondidos em 60 ms.
  - + O tempo médio de falha falha é até 2,3 dias.
  - + O sistema tem que resistir a ataques de scripts.
  - + O hardware vai custar US \$ 80.000 por plataforma.
  - + O software vai exigir 4 pessoas por ano, para manutenção.

# O ATAM

## + Passo 5 - Identificar Sensibilidades

- + Qualquer ponto da arquitetura que é afetado por qualquer alteração é considerado um ponto sensível.



# O ATAM

- + Passo 6 - Identificar Tradeoffs
  - + Criticar os modelos construídos.
  - + Identificar tradeoffs.
  - + Localização dos pontos de troca.
  - + Desempenho, segurança e disponibilidade.



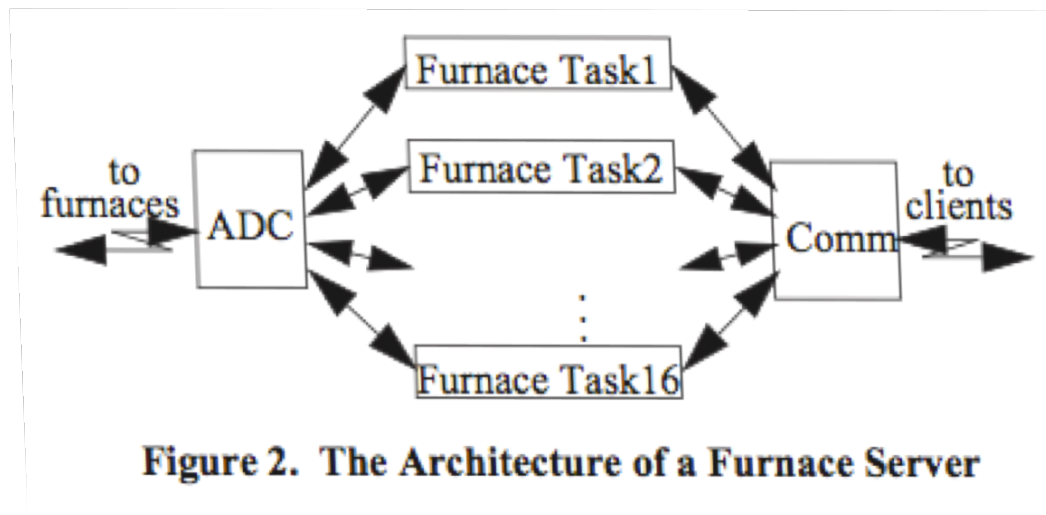
# Um exemplo em análise

## + Descrição do Sistema

### + RTS (remote temperature sensor)

+ Medir a temperatura de um conjunto de fornalhas

+ Reportar a temperatura para um operador qualquer cliente-servidor



# Coletando cenários

## + Cenário 1 – Desempenho

- + O cliente envia um pedido de controlo e recebe a primeira atualização periódica.
- + O cliente recebe atualizações periódicas à taxa especificada.

## + Cenário 2 – Disponibilidade

- + Servidor sofre uma falha de software e é reiniciado ou servidor sofrer uma falha de fornecimento de energia.

# Coletando requisitos

## + PR<sub>1</sub>

- + O cliente deve receber uma leitura de temperatura dentro  $F$  segundos depois de enviar uma solicitação de controle .

## + PR<sub>2</sub>

- + Dado que o cliente  $X$  pediu uma atualização periódica cada  $T(i)$ , segundo , deverá receber uma temperatura na média a cada  $T(i)$  segundos.

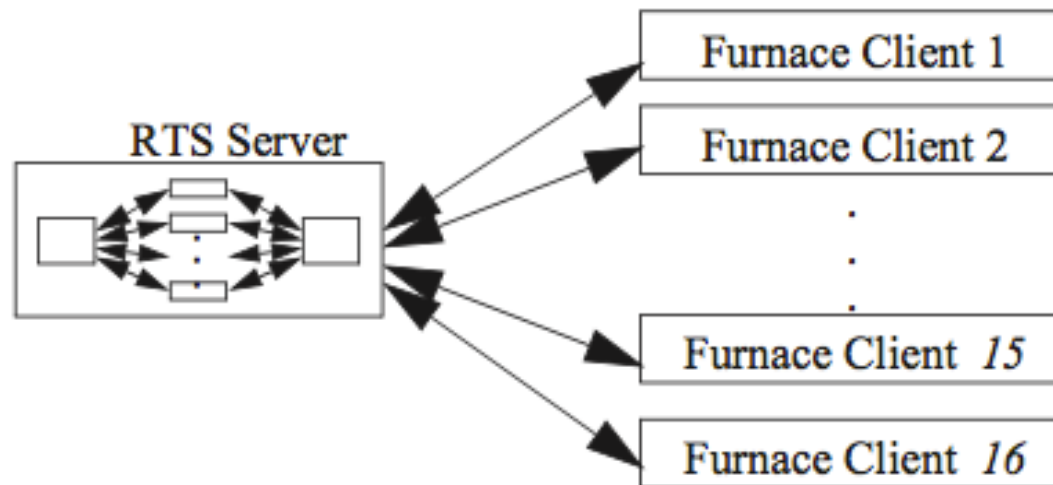
## + PR<sub>3</sub>

- + O intervalo entre as atualizações periódicas consecutivas não deve ser mais do que  $2T(i)$  segundo .

## + AR<sub>1</sub>

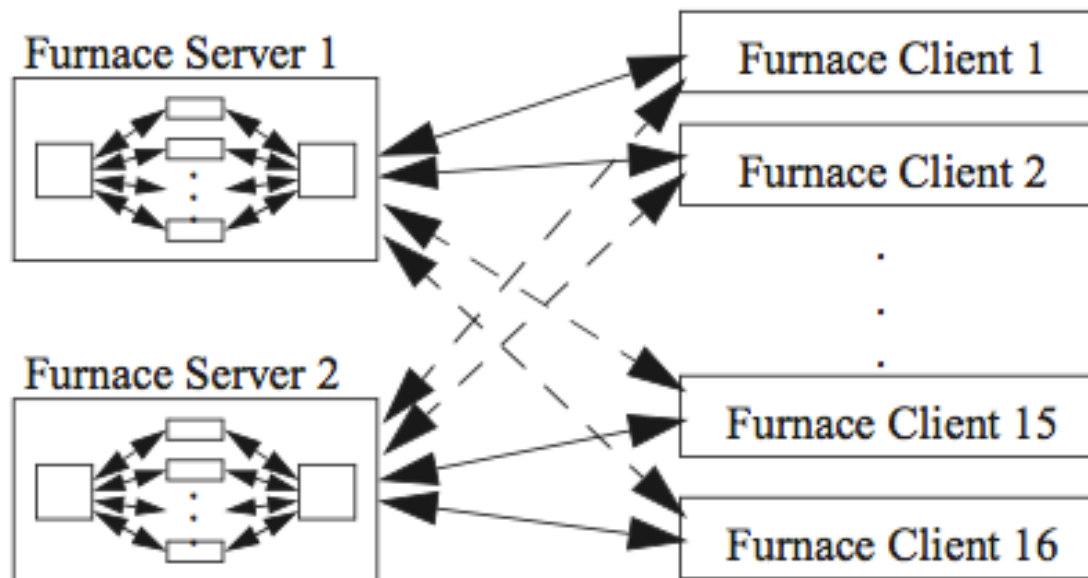
- + O sistema não deve estar indisponível para mais de 60 minutos por ano.

# Descrivendo a arquitetura



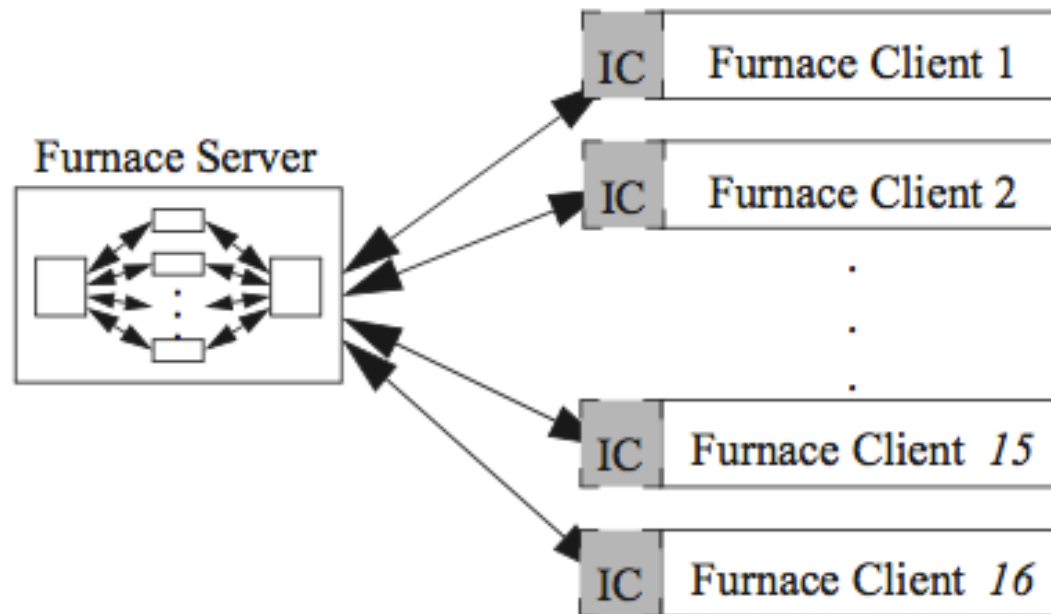
**Figure 3. Option 1's Architecture**

# Descrivendo a arquitetura



**Figure 4. Option 2's Architecture**

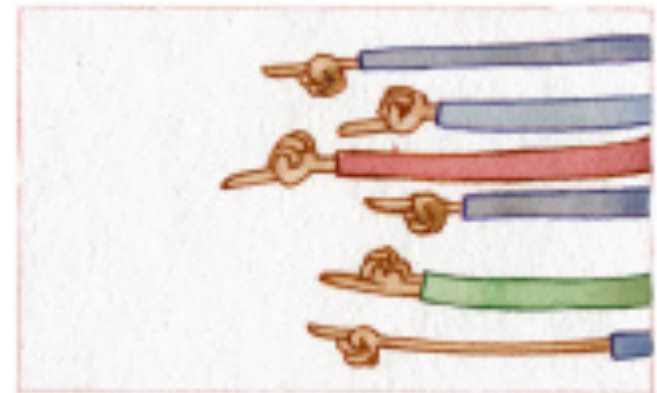
# Descrivendo a arquitetura



**Figure 5. Option 3's Architecture (with Cache)**

# Critica e analise

- + Opção 1
  - + Fraco desempenho e disponibilidade.
  - + Baixo custo de hardware
  
- + Opção 2
  - + Excelente disponibilidade, mas...
  - + Excelente desempenho, quando...
  - + Quando um servidor...
  
- + Opção 3
  - + Disponibilidade > Opção 1 && Desempenho > Opção 1
  - + Custo > Opção 1 && Custo < Opção 2



# As análises de sensibilidade

- + Opção 2 é a melhor, porem sensível ao número de servidores.
- + Opção 3 tem algumas características desejáveis em termos de custo e jitter.
- + Opção 2 + Opção 3
- + Tempo de vida do cache.



# TradeOff

- + Disponibilidade e desempenho estão positivamente correlacionada com o número de servidores.
- + A segurança é negativamente correlacionado com o número de servidores.
  - + Hot-spot (tradeoff)
    - + Custo
    - + Desempenho
    - + Disponibilidade
    - + Segurança

# Implicações do ATAM

- + Complexidade inerente ao projeto de software com o mundo real implica que uma análise raramente será uma atividade simples.
- + Gestão dos conflitos e interações que são revelados pelo ATAM coloca demandas pesadas sobre as habilidades de análise dos especialistas.



# Conclusão

- + Tornar racional as escolhas entre as arquiteturas concorrentes.
- + Proporcionar melhor compreensão e confiança
- + Pontos de tradeoff arquitetônicas
- + Planos de ação para modificar a arquitetura
- + Documentação com as justificativas das escolhas

