

# Padrões de Software

## E sua aplicação para Programas de Medição

Renata Alchorne

([renata.alchorne@gmail.com](mailto:renata.alchorne@gmail.com))

Centro de Informática - UFPE / Mestrado - 2008.2

Disciplina de Qualidade, Processos e Gestão de Software

Prof. Alexandre Vasconcelos

Recife, 13 de dezembro de 2008

# Agenda

- Motivação
- Padrões
- Classificação dos Padrões
- Linguagem de Padrão para Programas de Medição
- Conclusões
- Trabalhos Futuros
- Referências

# Motivação

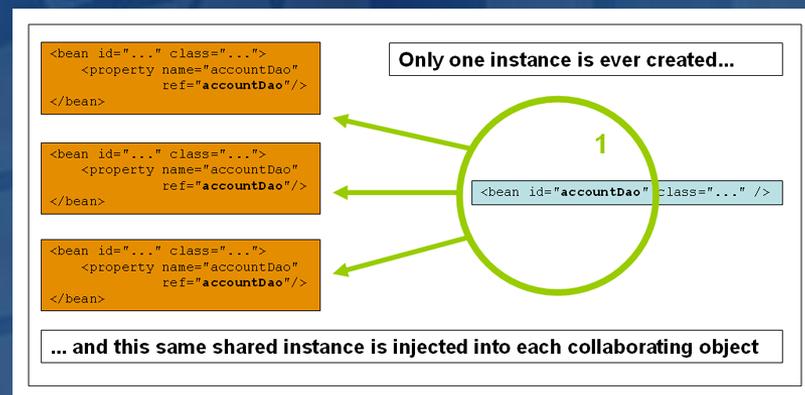
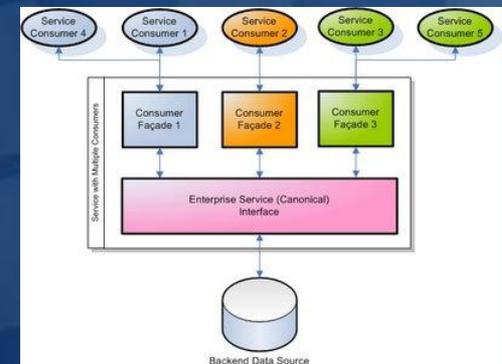
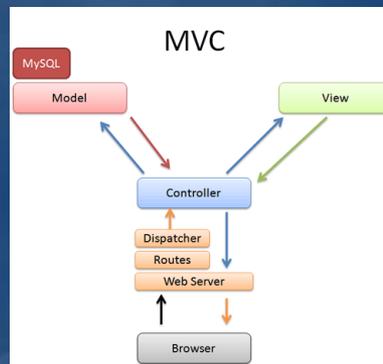
- O reuso de software é um tema que têm recebido bastante destaque nos últimos anos dentro da Engenharia de Software;
- O objetivo é reaproveitar o máximo possível do software produzido e informações associadas em projetos futuros , diminuindo o custo e aumentando a produtividade no desenvolvimento e própria evolução do produto;
- As organizações estão buscando, através de reuso, a redução de custo na produção de software pela diminuição do esforço;



# Motivação

- Os padrões de software têm sido pesquisados na última década como uma forma promissora de reuso
- Não se restringindo apenas à reutilização de código, mas também de:

- Projeto;
- Análise;
- Arquitetura;
- Processo;
- Interface;
- ...



# Motivação

- Diversos padrões têm sido propostos, seja para solucionar problemas de:
  - Programação [Coplien 92];
  - Projeto [Gamma 95, Pree 95];
  - Análise [Coad 92, Coad 95, Fowler 97].
- Conferências internacionais para discussão do assunto têm ocorrido desde 1994:
  - SugarLoafPLoP - Conferência Latino Americana de Linguagens de Padrões

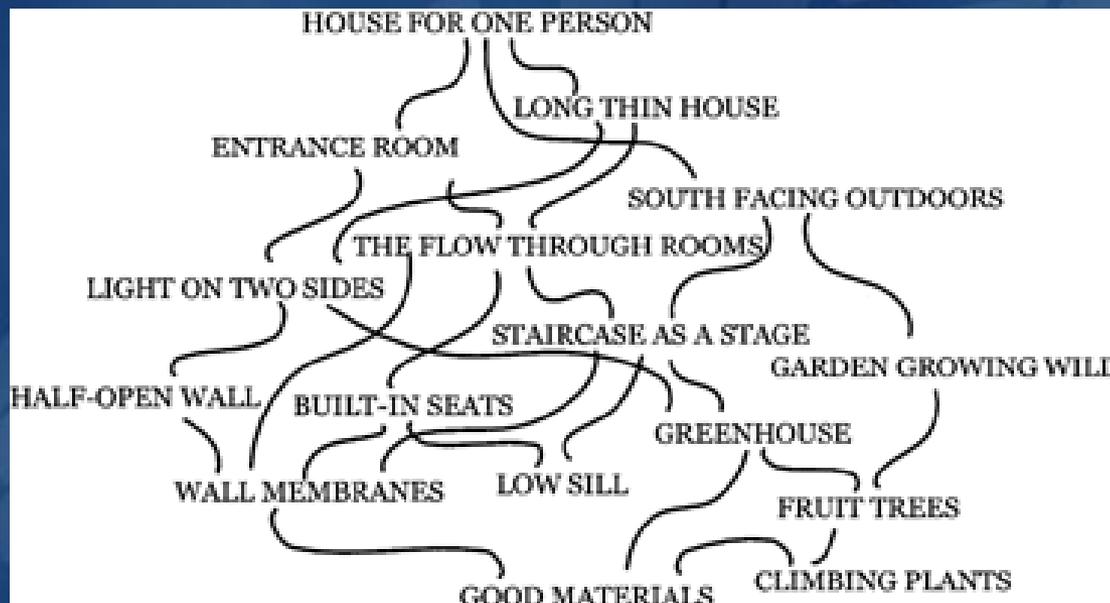


[http://sugarloafplop.dsc.upe.br/index\\_en.html](http://sugarloafplop.dsc.upe.br/index_en.html)

# Padrões

# Padrões

- Christopher Alexander cunhou o conceito de padrões de projeto na arquitetura predial;
- A primeira definição foi dada em 1979: “Cada padrão é uma regra de três partes, que expressa uma relação entre um certo contexto, um problema e uma solução” [Alexander 79];
- Um padrão de projeto consiste, portanto, na documentação de uma solução para um problema.



Exemplo de linguagem de padrões de Christopher Alexander [1]

# Padrões de Software

- Surgiram na conferência sobre programação orientada a objetos (OOPSLA) 1987;
- A intenção é a mesma: descrever soluções para problemas que ocorrem frequentemente no desenvolvimento de software e que podem ser reusadas por outros envolvidos em projetos futuros;
- Pode ser considerado um par “problema/solução” [Buschmann 96];
- O padrão capta a estrutura essencial e o raciocínio de uma família de soluções comprovadamente bem sucedidas para um problema repetido que ocorre sob um determinado contexto e um conjunto de repercussões [Appleton 97]

# Componentes de um Padrão

- Definição de alguns elementos ao se ler um padrão:
  - Nome
  - Problema
  - Contexto
  - Forças
  - Solução
    - Estrutura
    - Participantes
    - Dinâmica
    - Implementação
    - Variantes
  - Exemplos
  - Contexto resultante
    - Consequências
  - Raciocínio
  - Padrões relacionados
  - Uso conhecido

# Anti-Padrões

- Representam uma “lição aprendida”;
- Podem ser de dois tipos:
  - Aqueles que descrevem uma solução ruim para um problema que resultou em uma solução ruim;
  - Aqueles que descrevem como escapar de uma situação ruim e como proceder para, a partir dela, atingir uma boa solução.
- Eles são necessários, porque é tão importante ver e entender soluções ruins quanto as boas;
- O anti-padrão pode ser visto como uma mensagem “Não faça isso”



# Padrões e outras formas de reuso

- Padrões X Frameworks
  - [Johnson 97 ] diz que padrões são mais abstratos do que frameworks, porque um framework é um software executável, enquanto um padrão representa conhecimento e experiência sobre software;
  - Padrões são menores que frameworks; Frameworks podem englobar vários padrões dentro;
  - Os padrões são menos especializados do que os frameworks, já que os frameworks são desenvolvidos para um domínio específico e os padrões podem ser utilizados em mais de uma aplicação.

# Classificação de Padrões

# Classificação de Padrões

- Padrões de Software abrangem diferentes níveis de abstração. Porém, essa classificação não é rigorosa, podendo haver padrões que se encaixam em mais de uma classificação.
- As categorias mais faladas são:
  - *Padrões de Processo*: problemas envolvidos na engenharia de software;
  - *Padrões Arquiteturais*: expressam o esquema fundamental de sistemas de hardware ou software. Ex.: MVC - Model, View, Controller;
  - *Padrões de Padrão*: descreve como um padrão deve ser escrito, expresso para o usuário;
  - *Padrões de Análise*: descrevem soluções para problemas de análise de sistemas;
  - *Padrões de Projeto*: definem soluções para problemas de projeto de software. Ex.: Visitor (muito utilizado em compiladores);
  - *Padrões de Interface*: definem soluções para problemas comuns no projeto da interface de sistemas;

# Classificação de Padrões

- As categorias mais faladas são:
  - *Padrões de Programação*: descrevem soluções de problemas particulares de uma determinada linguagem ou estilos de programação. Ex.: Padrão de codificação Java;
  - *Padrões de Persistência*: descrevem soluções para problemas de armazenamento de informações em arquivos ou banco de dados. Ex.: XML.

# Coletâneas de Padrões

- Um padrão resolve um problema, mas pode gerar outros que, por sua vez, serão resolvidos através da aplicação de outros padrões;
- Em geral, não existem padrões isolados;
- Deve-se pensar em agrupar esses padrões segundo algum critério de forma que facilite sua recuperação e reuso:
  - Coleções de padrões;
  - Catálogos de padrões;
  - Sistemas de padrões;
  - Linguagem de padrões.

# Coletâneas de Padrões

- Coleções de Padrões
  - Trata-se de uma coletânea qualquer;
  - Não possuem vínculo entre si;
  - Podem estar juntos por terem sido apresentados em uma conferência, por exemplo;
  - Não possuem relacionamento semântico significativo.
- Catálogos de Padrões
  - Trata-se de uma coletânea de padrões relacionados, mesmo que fracamente ou informalmente;
  - Podem estar divididos em categorias;
  - Adiciona uma certa quantidade de estrutura e organização a uma coleção de padrões, mas não vai muito além de mostrar essa relação mais superficial.

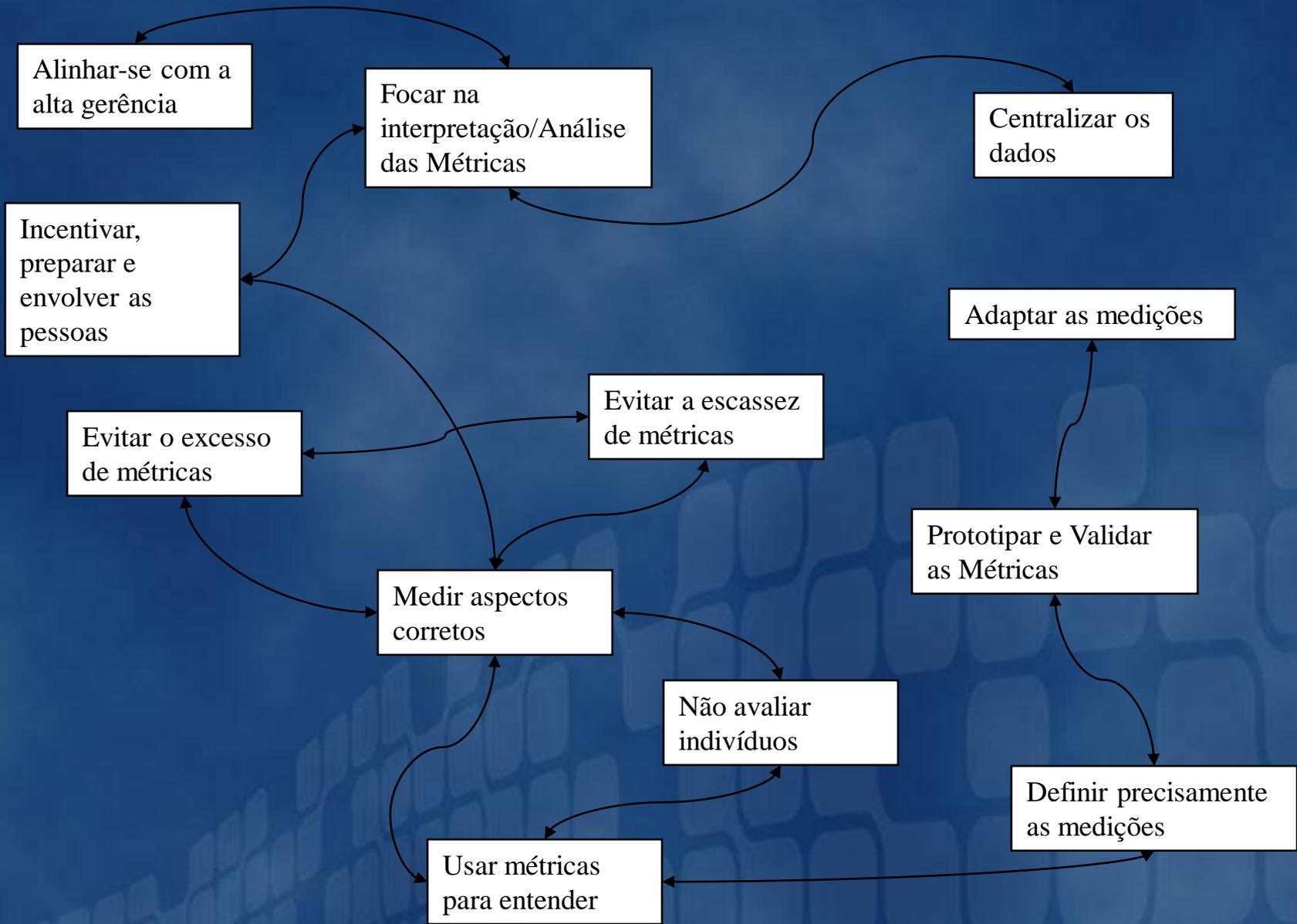
# Coletâneas de Padrões

- **Sistemas de Padrões**
  - Conjunto coeso de padrões co-relacionados;
  - Os padrões trabalham juntos para apoiar a construção e evolução de arquiteturas completas;
  - Descreve as diversas inter-relações entre os padrões e seus grupamentos e como eles podem ser combinados e compostos para resolver problemas mais complexos;
  - No sistema, os padrões devem ser descritos num estilo consistente e uniforme e precisam cobrir uma base de problemas e soluções suficientemente abrangente;
  - Um sistema de padrões é um veículo poderoso para expressar e construir arquiteturas de software [Buschmann 96]
  - Além da estrutura e organização oferecidas por um catálogo de padrões, num sistema de padrões é adicionada uma maior profundidade à estrutura, maior riqueza à interação dos padrões e maior uniformidade ao catálogo de padrões [Appleton 97].

# Coletâneas de Padrões

- Linguagens de Padrões
  - Uma linguagem de padrões é uma coleção estruturada de padrões que se apoiam uns nos outros para transformar requisitos e restrições numa arquitetura [Coplien 98];
  - Cobrem todos os aspectos importantes em um dado domínio;
  - Pelo menos um padrão deve estar disponível para aspecto da construção e implementação de um sistema de software;
  - Não pode haver “vazios” ou “brancos”;
  - É uma forma de subdividir um problema geral e sua solução complexa em um número de problemas relacionados e suas respectivas soluções;
  - Cada padrão da linguagem resolve um problema específico no contexto comum compartilhado pela linguagem;
  - Cada padrão pode ser usado separadamente ou com um certo número de padrões da linguagem

# Linguagem de Padrão para Programas de Medição



# Padrão 1: Alinhar-se com a alta gerência

[Aquino 2007]



# Padrão 1: Alinhar-se com a alta gerência [Aquino 2007]

- Contexto
  - Em uma organização a alta gerência contém os principais stakeholders dos resultados das medições
- Problema
  - Alta gerência se opõem às métricas
  - Alta gerência não requer explicitamente os dados das métricas
  - Alta gerência não participa diretamente da implantação do programa
- Solução
  - Alinhar o programa de métricas com os objetivos de negócio
  - Envolver a alta gerência no programa de medição

# Padrão 2: Evitar excesso de métricas

[Aquino 2007]



# Padrão 2: Evitar excesso de métricas

[Aquino 2007]

- Contexto
  - Muitos atributos em organizações de software podem ser medidos (processos, produtos, projetos e recursos)
  - Diferentes stakeholders requerem diferentes conjuntos de informações
- Problema
  - Iniciar com muitas métricas
    - Grande esforço de Coleta
    - Impossibilidade de analisar os dados coletados
- Solução
  - Começar pequeno e com um conjunto balanceado de métricas
  - Expandir à medida que os participantes se acostumem com as atividades relacionados e que os gerentes percebam o valor das informações

# Padrão 3: Evitar escassez de métricas

[Aquino 2007]



# Padrão 3: Evitar escassez de métricas

[Aquino 2007]

- Contexto
  - Na tentativa de evitar o problema relacionado ao Padrão 2, inicia-se o programa com pouquíssimas métricas
- Problema
  - Não provê informação suficiente para tomada de decisão
  - Pode levar à conclusão que o esforço do programa não vale a pena
- Solução
  - Medir aspectos opostos (Ex. Qualidade e Produtividade)
  - Priorizar segundo a importância das medições
  - Distribuir as medições de forma a não sobrecarregar um papel ou pessoa

# Padrão 4: Medir aspectos corretos

[Aquino 2007]

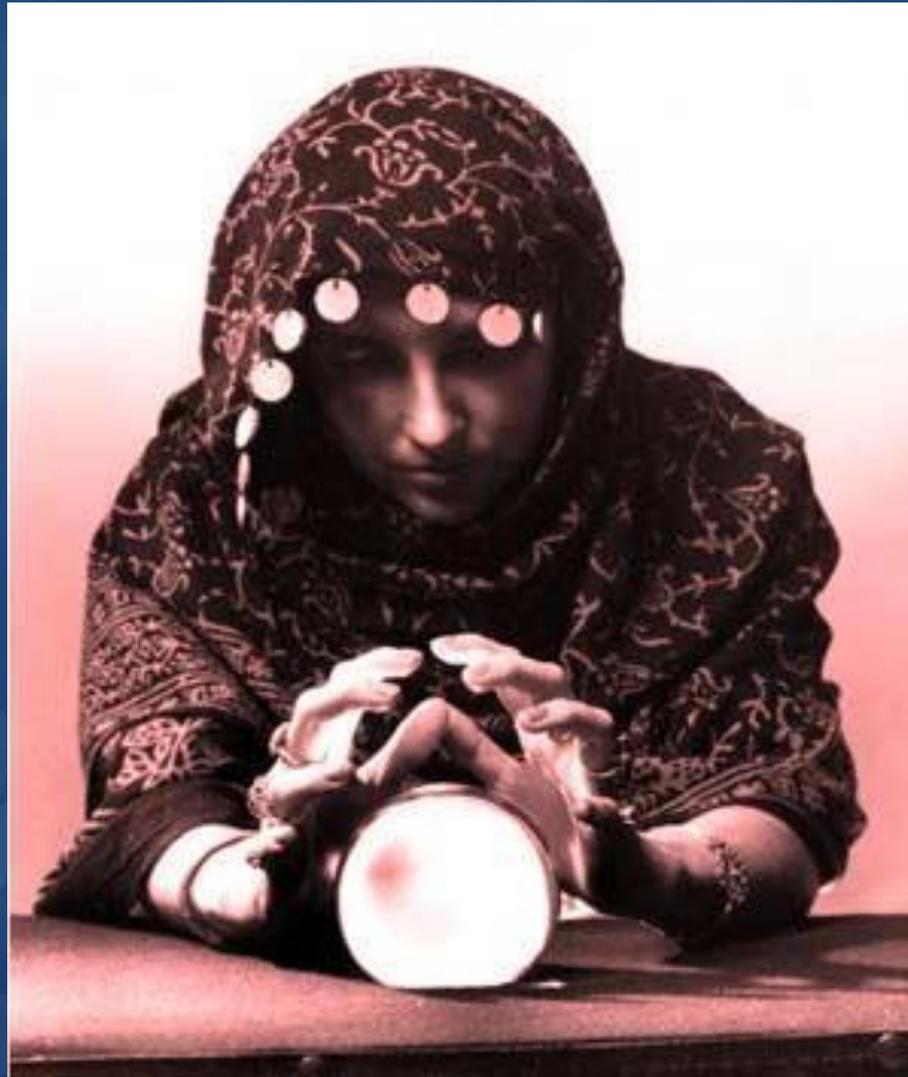


# Padrão 4: Medir aspectos corretos

[Aquino 2007]

- Contexto
  - Muitos atributos em organizações de software podem ser medidos (processos, produtos, projetos e recursos)
  - Há uma tendência natural em se medir o que é mais simples ou o que é requerido pela gerência
- Problema
  - Os itens coletados não estão diretamente relacionados à estratégia de negócio
  - Os gerentes não estão obtendo informações que eles precisam pra gerenciar melhor suas equipes
  - Não se consegue avaliar se mudanças no processo trouxeram melhorias
- Solução
  - Selecione métricas que ajudarão a avaliar as melhorias de processo
  - Utilize o GQM ou métodos similares

# Padrão 5: Definir precisamente as medições [Aquino 2007]



# Padrão 5: Definir precisamente as medições [Aquino 2007]

- Contexto
  - Existe uma tendência natural em priorizar a geração do gráfico, antes mesmo de se pensar ou se definir precisamente o que significa a medição
- Problema
  - Definição vaga ou ambígua da métrica
    - Diferentes interpretações
    - Dificuldades e retrabalho na coleta
- Solução
  - Definir de forma completa e precisa o conjunto de coisas que devem ser medidas

# Padrão 6: Não avaliar indivíduos

[Aquino 2007]



## BURNOUT

ATTITUDES ARE CONTAGIOUS. MINE MIGHT KILL YOU.

[www.despair.com](http://www.despair.com)

# Padrão 6: Não avaliar indivíduos [Aquino 2007]

- Contexto
  - Em uma organização muitos aspectos podem ser medidos, inclusive as pessoas
  - Avaliar a performance dos indivíduos baseado nas métricas reportadas, tende a ser um desejo natural da gerência
- Problema
  - As pessoas (produtoras da informação) tendem a esconder a informação caso essa venha lhe prejudicar
  - O processo pode ser burlado para se obter um “melhor valor da métrica”
  - O benefício individual passa a ser mais importante do que o benefício do time ou organização
- Solução
  - Deve ficar claro que o propósito do programa de métricas é
    - Entender como o software está sendo construído
    - Permitir que decisões possam ser tomadas com mais segurança
    - Avaliar o impacto de mudanças
  - Definir níveis de privacidade para algumas informações

# Padrão 7: Adaptar as medições (e não a organização) [Aquino 2007]



# Padrão 7: Adaptar as medições (e não a organização) [Aquino 2007]

- Contexto
  - Os stakeholders possuem necessidades de informação, para atendê-los, medições são definidas e implantadas na organização
- Problema
  - O modo da organização funcionar (processo) não viabiliza a implantação do indicador
  - Mudanças no processo são necessárias para implantar a medição definida
- Solução
  - Alterar o indicador e não o processo (a não ser que se perceba uma inconsistência semântica no processo)

# Conclusões

# Conclusões

- Padrões documentam uma parte repetida de um projeto, permitindo seu entendimento e aplicação em um contexto particular;
- Os padrões fornecem ao projeto um vocabulário comum aos *stakeholders*, facilitando a comunicação entre os mesmos;
- Os padrões constituem uma base de experiência para construção de software reutilizável;
- Padrões expõem conhecimento sobre a construção de software que foi ganho por especialistas em muitos anos.

# Trabalhos futuros

- Minha dissertação!
- Definir a classificação dos padrões
- Estruturar o padrão com base nas estruturas mais utilizadas e eficientes
- Levantar práticas dos programas de medição
- Documentar os padrões

# Referências

- [Coplén 92] Coplén, J.O. *Advanced C++ Programming Styles and Idioms*. Reading-MA, Addison-Wesley, 1992.
- [Gamma 95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns - Elements of Reusable Object-Oriented Software*. Reading-MA, Addison-Wesley, 1995.
- [Pree 95] Pree, Wolfgang. *Design Patterns for Object-Oriented Software Development*. Reading-MA, Addison-Wesley, 1995.
- [Coad 92] Coad, Peter. *Object-Oriented Patterns*. *Communications of the ACM*, V. 35, nº9, p. 152-159, setembro 1992.
- [Coad 95] Coad, P.; North, D.; Mayfield, M. *Object Models: Strategies, Patterns and Applications*, Yourdon Press, 1995.
- [Fowler 97] Fowler, M. *Analysis Patterns*. Menlo-Park-CA, Addison-Wesley, 1997.
- [Alexander 79] Christopher Alexander, *The Timeless Way of Building*, Oxford University Press, New York, 1979.
- [Buschmann 96] Buschmann, F. et al. *A System of Patterns*, Wiley, 1996.
- [Appleton 97] Appleton, Brad. *Patterns and Software: Essential Concepts and Terminology*, disponível na WWW na URL:  
<http://www.enteract.com/~bradappdocpatterns-intro.html>.

# Referências

- [Johnson 97 ] Johnson, Ralph E. CS497 Lectures - Lecture 12, 13, 14 e 17, disponível na WWW na URL: <http://st-www.cs.uiuc.edu/users/johnson/cs497/notes98/online-course.html>
- [Coplien 98] James O. Coplien. *Software Design Patterns: Common Questions and Answers*. In Linda Rising, editor, *The Patterns Handbook: Techniques, Strategies, and Applications*, p. 311-320. Cambridge University Press, New York, January 1998.
- [Aquino 2007] Aquino, G. *Padrões para implantação de programas de medição em organizações de software*. SugarLoafPLOP-2007.
- <http://sugarloafplop2005.icmc.usp.br/NotasDidaticasPadroes.pdf>

**Perguntas ?**