

Refatoração:

Melhorando a Qualidade
de Código Pré-Existente



Cursos de Verão 2009 – IME/USP

www.agilcoop.org.br

Mariana Bravo & Hugo Corbucci

Refatoração

- Uma modificação no sistema que **não altera o seu comportamento funcional**, mas **melhora sua estrutura interna**.
- Limpa o código minimizando as chances de introduzir erros.
- Melhorar o *design* depois que o código foi escrito.

De onde vem?

- Surgiu na comunidade de Smalltalk nos anos 80/90.
- Desenvolveu-se formalmente na Universidade de Illinois em Urbana-Champaign.
- Grupo do Prof. Ralph Johnson.
 - Tese de PhD de William Opdyke (1992).
 - John Brant e Don Roberts:
 - *The Refactoring Browser Tool*
- Kent Beck (XP) e Martin Fowler na indústria.

O espírito da refatoração



Antes

O espírito da refatoração



Antes

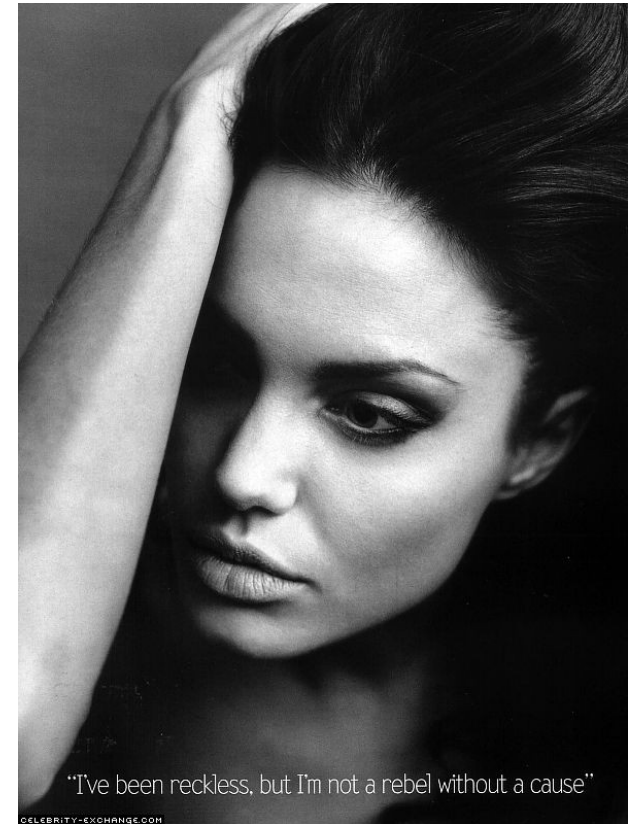
→
(refatoração)

O espírito da refatoração



Antes

→
(refatoração)



Depois

Exemplos



Extrair método

- Transforma um fragmento de código em um método com um nome explicativo.
- Motivação:
 - Diminuir e facilitar o entendimento de trecho de código.
 - Aumentar as chances de reutilização do novo método.

Renomear variável

- Muda o nome de uma variável para expressar melhor o seu propósito
- Motivação:
 - Não é fácil entender a função da variável em seu escopo

Dinâmica da refatoração

- Cada mudança é simples...
- Mas seu efeito acumulado pode melhorar muito o *design* e o código

- É construindo o sistema que podemos descobrir como melhorá-lo

Refatorar para...

- Melhorar o *design* do software
- Tornar o software mais fácil de entender
- Encontrar falhas mais facilmente
- Programar mais rapidamente

Quando refatorar

- Sempre que você precisar fazer algo e o código estiver atrapalhando:
 - Na terceira vez, refatore
 - Quando adiciona funcionalidade
 - Quando corrige um erro
 - Quando revisa o código
- Quando o código cheira mau
 - “*If it stinks, change it.*” (Se feder, troque-o.)
Grandma Beck, falando sobre como cuidar de bebês.

Alguns maus cheiros

- Nomes de variáveis obscuros
- Código duplicado
- Método muito longo
- Classe muito grande
- Intimidade inapropriada
- Comentários
- Muitos parâmetros

Primeiro passo: testes

- Antes de refatorar, tenha um conjunto sólido de testes para garantir que o comportamento não seja alterado.
- Refatorações podem adicionar erros.
 - porém, como são feitas em pequenos passos, é fácil recuperar-se de uma falha
- Os testes ajudam a detectar erros se eles forem criados.
- Testes têm que ser automáticos e ser capazes de se auto-verificarem.

Maus cheiros em testes

- Código dos Testes:
 - Testes Obscuros
 - Código Complexo: condições
 - Difícil de testar
 - Replicação de código de testes
 - Lógica de testes em produção
- Cheiros de comportamento:
 - Testes frágeis
 - Intervenções manuais
 - Testes lentos



Exemplos



Mover método

- Move um método que utiliza mais funcionalidades de outra classe do que aquela em que se encontra.
- Motivação:
 - Uma classe tem muitos comportamentos
 - O acoplamento entre classes é muito forte
- O velho método delega ou é removido.

Usando ferramentas



Substituir temporário com chamada

- Substitui o uso de um variável por uma chamada a um método que realiza as operações.
- Motivação:
 - Variáveis temporárias incentivam seu uso prolongado por terem um escopo limitado

Mais um exemplo



E o desempenho?

"We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil."

(Devemos esquecer as pequenas eficiências em 97% do tempo: otimização prematura é a raiz de todo o mal.)

Donald Knuth

- Usar *profiling* para encontrar os gargalos
- Otimizar apenas estes gargalos
- Código fatorado:
 - Compra tempo para otimizar
 - Aumenta a precisão na otimização

Dois chapéus

- Refatoração



- Adicionando funcionalidades



- Corrigindo erros

Mais exemplos: sugestões?



Problemas com refatorações

- Refatoração de sistemas grandes ou enormes
- Refatoração com bancos de dados
- Refatoração de APIs públicas

- Quando NÃO refatorar?
 - Quando é tão ruim que reescrever é melhor
 - Quando você está próximo de um prazo!

Ferramentas para Refatoração

- Refactoring Browser Tool.
- Dá suporte automatizado para uma série de refatorações.
- Pode melhorar em muito a produtividade.
- Existem há vários anos para Smalltalk.
- Já há várias para C++ e Java.
- Iniciativas acadêmicas (Ralph@UIUC).
- Agora, integrado no Eclipse e no Visual Works.

Catálogo de Refatorações

- [Fowler, 2000] “Refatoração”, contém 72 refatorações.
- Análogo aos padrões de desenho orientado a objetos [Gamma et al. 1995] (GoF).
- [Kerievsky, 2004] “Refactoring to patterns”, catálogo com 27 refatorações que aplicam padrões!

Formato de refatorações nos catálogos

- **Nome** da refatoração.
- **Resumo** da situação na qual ela é necessária e o que ela faz.
- **Motivação** para usá-la (e quando não usá-la).
- **Mecânica**, i.e., descrição passo a passo.
- **Exemplos** para ilustrar o uso.

Referências

- www.refactoring.com
- Martin Fowler. *Refactoring: improving the design of existing code.*
- Kerievsky. *Refactoring to patterns*