

Reengenharia, Refatoração e Bad Smell

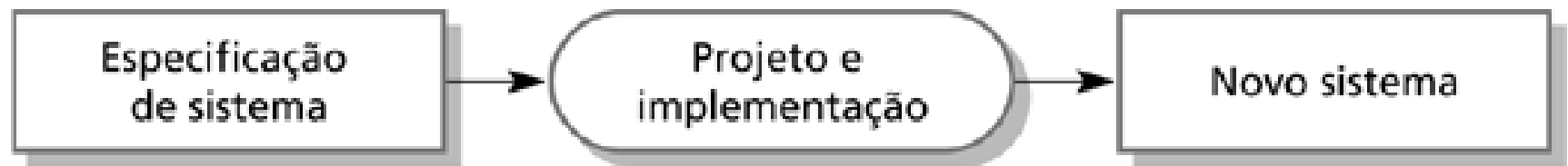
Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

[Reengenharia de Sistemas]

- Reestruturação ou reescrita de parte ou de todo um sistema legado sem mudança na sua funcionalidade
 - O objetivo é melhorar a qualidade
- O sistema pode ser reestruturado e/ou redocumentado
 - Aplicável a subsistemas ou sistemas de grande porte que necessitam de manutenção frequente

[Engenharia x Reengenharia]



Engenharia direta



Reengenharia de software

[Benefícios da Reengenharia]

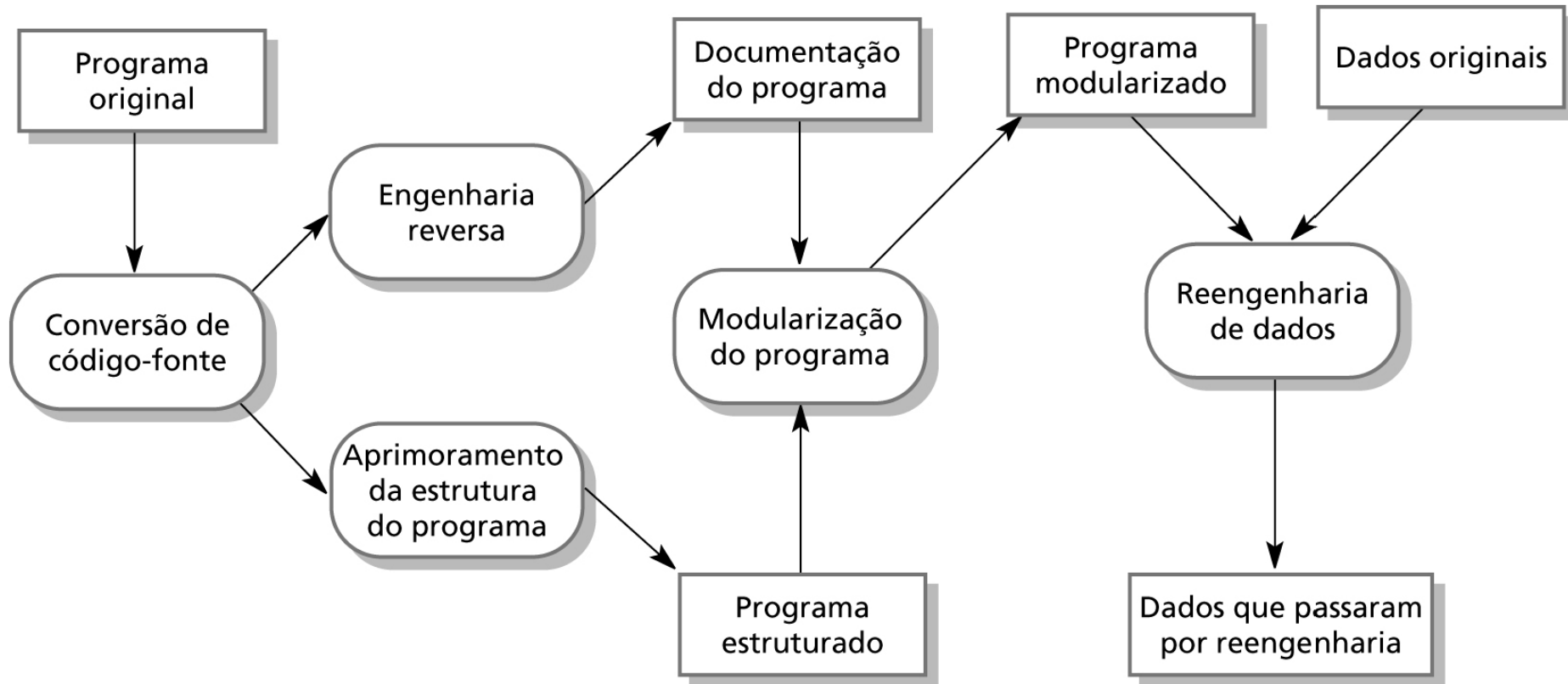
■ Risco Reduzido

- O risco de desenvolver o sistema novamente é alto
- Podem ocorrer erros de especificação ou problemas no desenvolvimento

■ Custo Reduzido

- O custo de reengenharia pode ser muito menor que o custo de desenvolvimento
- Ferramentas podem automatizar parcialmente

[Processo de Reengenharia]



[Atividades de Reengenharia]

- Conversão de código-fonte
 - Converter o código para uma nova linguagem
- Engenharia reversa
 - Analisar o programa para compreendê-lo e documentá-lo
- Aprimoramento da estrutura (refatoração)
 - Analisar e modificar a estrutura para facilidade de entendimento

Atividades de Reengenharia

- Modularização de programa
 - Reorganizar a estrutura do programa para torná-lo mais modular
 - Partes redundantes são identificadas e removidas
- Reengenharia de dados
 - Limpar e reestruturar os dados (ou estruturas de dados) do sistema
 - Conversão do banco de dados

[Refatoração]

- Processo de alterar a estrutura de um programa sem mudar o seu comportamento observável
 - Objetivo é reduzir a complexidade e torná-lo mais compreensível
- Refatoração pode ser visto como uma manutenção preventiva
 - Nenhuma nova funcionalidade é incluída nem há adaptações ou correções

[Refatoração e o XP]

- Refatoração é muito praticada na Programação Extrema (XP)
 - Reflexo das mudanças contínuas que podem degradar a estrutura do sistema
- Alterar a estrutura do programa podem causar novos erros
 - O uso frequente de testes ajudam na detecção de erros de refatoração





Bad Smells

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

[*Bad Smell* e Refatoração]

- *Bad smell* é uma situação no qual a estrutura do programa pode ser melhorada com refatoração
- Exemplos de *bad smells*
 - Código Duplicado
 - Classes ou métodos longos



[*Large Class / God Class*]

- Uma classe que faz coisa demais no sistema
 - Um sintoma pode ser o excesso de atributos
- Refatorações sugeridas
 - *Extract Class*: dividir a classe em duas
 - *Extract Subclass*: criar uma subclasse para a classe



Long Method / God Method



- Métodos que centralizam a funcionalidade da classe
 - Estes métodos são difíceis de entender e de manter
- Refatorações sugeridas
 - *Extract Method*: dividir o método em dois
 - *Replace Method with Method Object*: transformar um método em uma classe

[*Feature Envy*]

- Parte do código de uma classe “inveja” outra classe
 - Por exemplo, um método de uma classe usa atributos somente da outra classe
- Refatorações sugeridas
 - *Move Method* e *Move Field*: mover métodos e atributos entre classes
 - *Inline Class*: juntar duas classe em uma

[*Divergent Change*]

- Ocorre quando uma classe pode mudar frequentemente de diferentes formas e por razões distintas
 - Idealmente, cada classe deve ser alterada apenas por um tipo de mudança
- Refatorações sugeridas
 - *Extract Class*: dividir a classe em duas

[Shotgun Surgery]

- Oposto do *Divergent Change*
 - Toda vez que você alterar uma classe, você tem que fazer várias pequenas mudanças em outras classes diferentes
- Refatorações sugeridas
 - *Move Method* e *Move Field*: mover métodos e atributos entre classes
 - *Inline Class*: juntar duas classes em uma



[*Refused Bequest*

- Uma classe herda atributos e métodos de outra classe, mas não os usa
 - Algo está errado com a decomposição hierárquica (classe e subclasses)
- Refatorações sugeridas
 - *Push Down Method / Field*: mover o método ou atributo da superclasse para a subclasse
 - *Replace Inheritance with Delegation*: substituir o relacionamento de herança por associação

[Comments]

- Comentários não é uma coisa ruim
 - Entretanto, excesso de comentários pode indicar que os nomes de métodos/atributos não estão suficientemente expressivos
- Refatorações sugeridas
 - *Extract Method*: quebrar um método em dois
 - *Rename Method/Field*: re-nomear

[Bibliografia da Aula]

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
 - Seção 9.3 Manutenção de Software
- Martin Fowler. **Refactoring: Improving the Design of Existing Code**. Addison-Wesley Professional, 1st edition, 1999.
 - Capítulo 3 Bad Smell in Code