

ENGENHARIA DE SOFTWARE

Refatoração de software



O QUE É REFATORAÇÃO DE SOFTWARE?

Uma alteração feita na estrutura interna do software para torná-lo **mais fácil de ser entendido e menos custoso de ser modificado** sem alterar seu comportamento observável.

O QUE É REFATORAÇÃO DE SOFTWARE?

Mais do que apenas “**limpeza de código**”.

Técnica para limpar código de forma **eficiente e controlada**.

POR QUE REFATORAR?

Melhora o projeto do software.

Torna o software mais fácil de entender.

Ajuda a encontrar defeitos/falhas.

Ajuda a programar mais rapidamente.

BENEFÍCIO DA REFATORAÇÃO

Permite pensar em um projeto inicial que não representa a melhor solução, mas sim uma **solução razoável**.

À medida que a solução é construída e o problema é melhor compreendido, uma **solução melhor**, diferente da original, pode ser percebida.

QUANDO REFATORAR?

Algo a ser feito o tempo todo, em pequenas quantidades.

“Três vezes, e você refatora!”

Quando acrescentar funções.

Quando precisar consertar um defeito.

Quando estiver revisando o código.

LIMITAÇÕES DA REFATORAÇÃO

Difícil refatorar bancos de dados.

Difícil refatorar interfaces de objetos.

Pode diminuir o desempenho do software.

CATÁLOGO DE REFATORAÇÃO

- ✓ Compondo métodos
- ✓ Movendo recursos entre objetos
- ✓ Organizando dados
- ✓ Simplificando expressões condicionais
- ✓ Tornando as chamadas de métodos mais simples
- ✓ Lidando com generalização
- ✓ Refatorações grandes

CATÁLOGO DE REFATORAÇÃO

➤ Composto métodos:

1. Extrair método
2. Internalizar método
3. Internalizar variável temporária
4. Substituir variável temporária por consulta
5. Introduzir variável explicativa
6. Dividir variável temporária
7. Remover atribuições a parâmetros
8. Substituir método por objeto método
9. Substituir o algoritmo

CATÁLOGO DE REFATORAÇÃO

➤ Movendo recursos entre objetos:

1. Mover método
2. Mover campo
3. Extrair classe
4. Internalizar classe
5. Ocultar delegação
6. Remover intermediário
7. Introduzir método externo
8. Introduzir extensão local

CATÁLOGO DE REFATORAÇÃO

➤ **Organizando dados:**

1. **Autoencapsular campo**
2. **Substituir atributo por objeto**
3. **Mudar de valor para referência**
4. **Mudar de referência para valor**
5. **Substituir vetor por objeto**
6. **Duplicar dados observados**
7. **Transformar associação unidirecional em bidirecional**
8. **Transformar associação bidirecional em unidirecional**
9. **Substituir números mágicos por constantes simbólicas**
10. **Encapsular campo**

CATÁLOGO DE REFATORAÇÃO

➤ **Organizando dados:**

11. Encapsular coleção
12. Substituir registro por classe de dados
13. Substituir enumeração por classe
14. Substituir enumeração por subclasses
15. Substituir enumeração pelo padrão state/strategy
16. Substituir subclasse por campos

CATÁLOGO DE REFATORAÇÃO

➤ **Simplificando expressões condicionais:**

1. **Decompor condicional**
2. **Consolidar expressão condicional**
3. **Consolidar fragmentos condicionais duplicados**
4. **Remover flag de controle**
5. **Substituir condição aninhada por cláusulas guarda**
6. **Substituir comando condicional por polimorfismo**
7. **Introduzir objeto nulo**
8. **Introduzir asserção**

CATÁLOGO DE REFATORAÇÃO

➤ Tornando as chamadas de métodos mais simples:

1. Renomear método
2. Acrescentar parâmetro
3. Remover parâmetro
4. Separar a pesquisa do modificador
5. Parametrizar método
6. Substituir parâmetro por métodos explícitos
7. Preservar o objeto inteiro
8. Substituir parâmetro por método
9. Introduzir objeto parâmetro
10. Remover método de gravação

CATÁLOGO DE REFATORAÇÃO

➤ Tornando as chamadas de métodos mais simples:

11. Ocultar método
12. Substituir construtor por um método fábrica
13. Encapsular downcast
14. Substituir código de erro por exceção
15. Substituir exceção por teste

CATÁLOGO DE REFATORAÇÃO

➤ Lidando com generalização:

1. Subir campo na hierarquia
2. Subir método na hierarquia
3. Subir o corpo do construtor na hierarquia
4. Descer método na hierarquia
5. Descer campo na hierarquia
6. Extrair subclasse
7. Extrair superclasse
8. Extrair interface
9. Condensar hierarquia
10. Criar um método padrão
11. Substituir herança por delegação
12. Substituir delegação por herança

CATÁLOGO DE REFATORAÇÃO

➤ Refatorações grandes:

1. Desembaraçar herança
2. Converter projeto procedural em objetos
3. Separar o domínio da apresentação
4. Extrair hierarquia

REFERÊNCIAS

FOWLER, M. Refatoração – Aperfeiçoando o projeto de código existente. Porto Alegre: Bookman, 2008.