



FACULDADE ZACARIAS DE GÓES

13/12/2008

CAMILO LOPES DE MEDEIROS NETO

SIMONE P. CRUZ (Orientador Geral)

ADONAI ESTRELA MEDRADO (Orientador Específico)

**Valença - Bahia
2008**

TEMA

Implicações de refatoração em desenvolvimento e manutenção de *Software*

PROBLEMA

Quais as implicações de refatoração em desenvolvimento e manutenção de Software?

JUSTIFICATIVA

Auxiliar no desenvolvimento de sistemas orientados a objetos implementados na linguagem *Java* que possuem baixa **reusabilidade de código**, **manutenção difícil** e a **adição ou adequação a novos requisitos** demandar muito esforço do engenheiro de *Software*, programador, analista de sistemas pontos que podem gerar grande custo ao projeto.

OBJETIVO GERAL

Verificar as implicações da técnica de refatoração em desenvolvimento e manutenção de *Software*

OBJETIVOS ESPECÍFICOS

- a) realizar um estudo bibliográfico visando descrever e analisar de forma qualitativa as aplicações da técnica de refatoração no desenvolvimento e manutenção de *Software*;
- b) apresentar uma análise quantitativa através de estudos de casos da literatura existente, relacionando refatoração com o desenvolvimento e manutenção de *Software*.

HIPÓTESES

- a) a técnica de refatoração pode ser aplicada no desenvolvimento e manutenção do *Software*;
- b) refatoração possibilita o acréscimo de nova funcionalidade de maneira fácil evitando a desestruturação do código;
- c) com o uso da refatoração gasta-se menor tempo possível na manutenção a nível de código, melhora a legibilidade e estrutura o código.

MÉTODO DE TRABALHO

- ✓ Abordagem qualitativa com um levantamento bibliográfico consultando livros, trabalhos acadêmicos e artigos na internet.
- ✓ Processo quantitativo por dados, onde a tabulação das informações serão no **modo texto, tabelas, estudo de caso e códigos na linguagem *Java*** como exemplos para as explicações.

INTRODUÇÃO

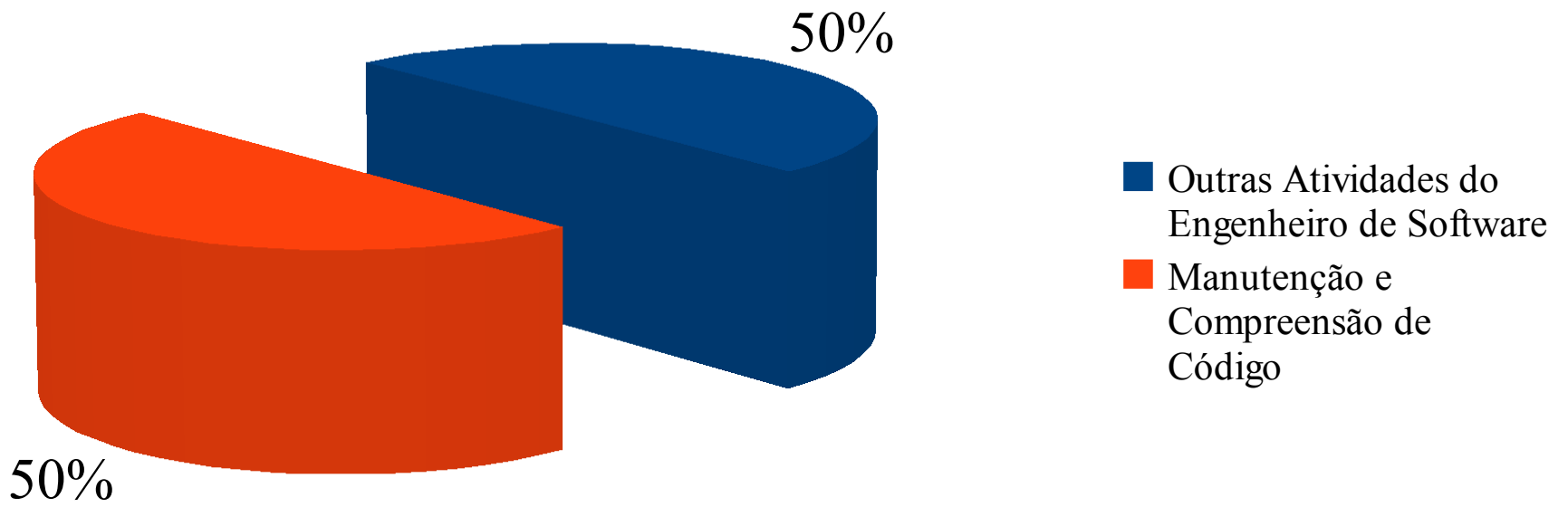


Gráfico 1 Estima-se que cerca de 50% do tempo de um engenheiro de *Software* é gasto com tarefas de manutenção e compreensão de código

Fonte: Maia (2004)

INTRODUÇÃO

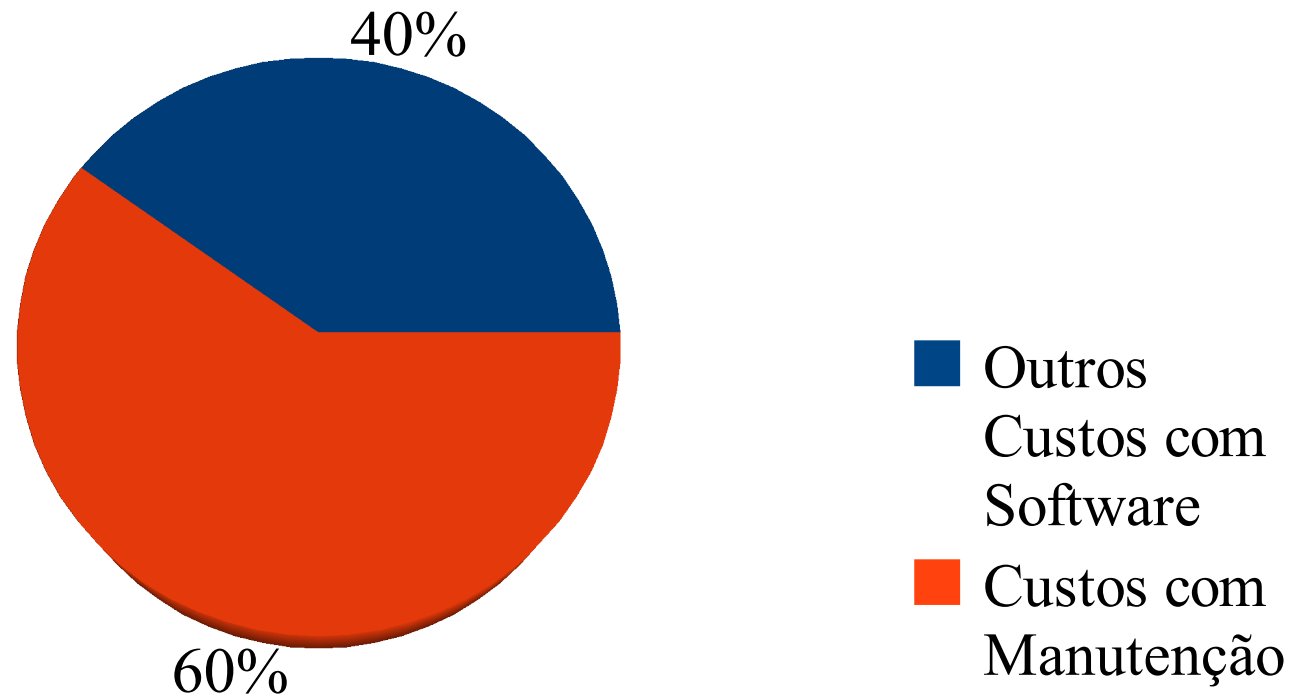


Gráfico 2 ao longo das últimas três décadas mais de 60% dos custos de desenvolvimento de *Software* das organizações foram gastos com manutenção.

Fonte: Maia (2004)

REFATORAÇÃO

A refatoração ajuda a tornar o código mais legível e resolver problemas de códigos mal escritos (MAIA,2004).

DEFINIÇÃO REFATORAÇÃO

Refatoração é o processo de reestruturar o sistema sem alterar suas funcionalidades (Fowler,2004).

MOTIVOS PARA USAR A REFORAÇÃO

Segundo Fowler (2004), refatora-se para:

- a) melhorar o projeto do *Software*;
- b) ter um o *Software* mais fácil de entender a nível de código;
- c) programar mais rapidamente

QUANDO REFATORAR

Segundo Fowler (2004):

- a) Código Duplicado;
- b) Método Longo;
- c) Comentários supérfluos;
- d) Comandos *switch*

FERRAMENTAS

- a) **XRefactoring**: As linguagens compatíveis são C e Java
essa ferramenta possibilita a geração de documentação no formato HTML.
- b) **RefactorIT**: É uma ferramenta para refatoração que pode integrada em várias IDEs.
- c) **JFactor**: É uma ferramenta que permite a aplicação automática da refatoração a programas em *Java*.

RESULTADOS

- a) Facilita no processo de manutenção e desenvolvimento do *Software*
- b) Pode ser aplicada a qualquer linguagem orientada a objetos
- c) Torna o *Software* flexível para adição de novas funcionalidades sem desestruturar do código
- d) mantém o comportamento do *Software*

ESTUDO DE CASO 1

“*Refactoring* no direitolivre aplicado padrões de projetos” (RAMOS,2006)

- **Desafio:** planejar e melhorar um sistema já existente denominado DireitoLivre
- **Arquitetura:**JavaEE
- **Problema:**código duplicado, teste condicional com lógica complexa, lista grandes nos parâmetros dos métodos.
- **Resultados:** legibilidade código, benefícios área engenharia de *Software*, manteve o sistema inalterado.

ESTUDO DE CASO 2

“Refatorando o SimGRIP: Um estudo de caso acerca da aplicação de técnicas de refatoração de *Software*” (LIMA,2007).

- **Desafio:** planejar e melhorar um sistema já existente denominado SimGRIP
- **Arquitetura:**Java
- **Problema:**código com difícil compreensão, difícil inserção de nova funcionalidade, Código Duplicado.
- **Resultados:**melhorou adequação de novas funcionalidades,legibilidade do código e tornou o *Software* mais fácil para atualizações.

CONSIDERAÇÕES FINAIS

Do que foi apresentado, conclui-se que:

a) a Técnica de refatoração pode ser aplicada no desenvolvimento e manutenção do *Software*, uma vez que, como demonstrado ao longo do texto que a técnica pode auxiliar os engenheiros e programadores no processo de **desenvolvimento e manutenção preventiva de *Software***;

b) refatoração possibilita o acréscimo de nova funcionalidade de maneira fácil evitando a desestruturação do código, já que essa técnica propicia a **legibilidade a nível de código** que evita desestruturação;

c) com o uso da refatoração gasta-se menor tempo possível na manutenção a nível de código, melhora a legibilidade e estrutura o código, uma vez que remove **código duplicados, comentários desnecessário, códigos condicionais difíceis de ler**.

*"Não existe sorte. A sorte é o encontro do
preparo com a oportunidade.
Portanto, se quiser ter sorte na vida, esteja
preparado para as oportunidades."*

Albert Einstein

REFERÊNCIAS

FOWLER, M. **Refatoração Aperfeiçoando o Projeto de Código Existente** 1. ed. Porto Alegre - RS: Bookman,2004.

LIMA, C., Guedes, et al. **O SIMGRIP: Um estudo de caso acerca da aplicação de técnicas de refatoração de Software** . João Pessoa, PB: Centro Federal de Educação Tecnológica da Paraíba, 2007.

MAIA, P. H. **REFAX:Um arcabouço para desenvolvimento de ferramentas de refatoração baseado XML**. Monografia (Programa de Pós Graduação em Ciência da Computação) Universidade FederalCeará, Fortaleza 2004.

RAMOS, L. **Refactoring aplicando padrões de projetos JavaEE** . Monografia. Universidade Luterana do Brasil, Gravatai, 2006.