

PROGRAMAÇÃO MODULAR

QUALIDADE DE SOFTWARE: UMA INTRODUÇÃO

PROF. JOÃO CARAM

ENGENHARIA DE SOFTWARE

2

- ▣ Objetivo da Engenharia de Software:
 - ▣ Produção sistemática de software de boa qualidade.



▣ Produção sistemática:

▣ Processo de desenvolvimento de software:

- Especificação;
- Desenvolvimento;
- Teste/manutenção;
- Evolução.

■ Boa qualidade:

- Fatores externos: aqueles percebidos pelos usuários ou com os quais os usuários interagem.
- Fatores internos: aqueles relacionados com o projeto, com os quais os programadores interagem.

FATORES EXTERNOS

5

- 
- ▣ Correção
 - ▣ Robustez
 - ▣ Extensibilidade
 - ▣ Reusabilidade
 - ▣ Compatibilidade
 - ▣ Portabilidade
 - ▣ Eficiência
 - ▣ Facilidade de uso
 - ▣ Integridade
 - ▣ Verificabilidade

FATORES INTERNOS

6

- Fatores externos são os percebidos pelos usuários
 - *Muito importante!!*
- Fatores externos são atingidos por meio dos fatores internos:
 - Legibilidade (+ Inteligibilidade)
 - Manutenibilidade
 - Modularidade

FATORES INTERNOS

7

■ Legibilidade:

- Facilidade de identificar os elementos que compõem o código-fonte.
 - Nomeação de componentes, organização do código.

■ Inteligibilidade:

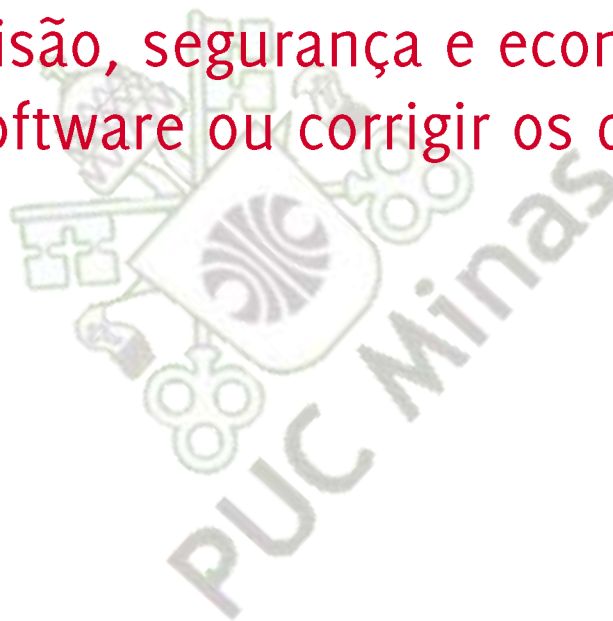
- Facilidade de compreender a coerência de um software em um nível superior do que a legibilidade proporciona.

FATORES INTERNOS

8

▣ Manutenibilidade:

- ▣ Facilidade, precisão, segurança e economia para modificar um software ou corrigir os defeitos detectados.



FATORES INTERNOS

9

▣ Modularidade:

- ▣ *“Mecanismo para aumentar a flexibilidade e compreensibilidade de um sistema, ao mesmo tempo em que permite a redução do seu tempo de desenvolvimento.”*

(David Parnas)

FATORES INTERNOS

10

■ Modularidade:

- Divisão do sistema em partes distintas: *módulos*.
- Ideia fundamental: quebrando o problema em partes menores, facilita-se sua solução e a reutilização das partes.

FATORES INTERNOS

11

▣ Modularidade:

- ▣ Encapsulamento;
- ▣ Abstração;
- ▣ Coesão e acoplamento.



FATORES INTERNOS

12

▣ Modularidade:

- ▣ Isolamento para testes e manutenção.
- ▣ Interface pública uniforme.
- ▣ Criação de bibliotecas reutilizáveis.
- ▣ Economia de memória.

QUALIDADE DE SOFTWARE

13

- Os sete pecados capitais no desenvolvimento de software.

POTENCIAIS
ERROS

DUPLICAÇÃO

QUEBRA DE
REGRAS DE
CÓDIGO

FALTA DE
TESTE
UNITÁRIO

MÁ
DISTRIBUIÇÃO DE
COMPLEXIDADE

PROJETO
ESPAGUETE

POUCOS/MUITOS
COMENTÁRIOS

COMEÇANDO: E O SEU CÓDIGO?

14

- Escreva um programa que permita ao usuário realizar, quantas vezes quiser, as quatro operações aritméticas básicas.
- Faça o melhor possível utilizando as técnicas, algoritmos e estruturas aprendidas em AEDs I e II.

UM RECADO DO FOWLER

15

“Any fool can write code that a computer can understand... But only good programmers write code that humans can understand.”

(Martin Fowler)



OBRIGADO.

DÚVIDAS?