

Fatores de qualidade de software

Prof. Hugo de Paula



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Departamento de Ciência da Computação

Sumário

- 1 Qualidade de Software
 - Engenharia de software

- 2 Fatores de qualidade
 - Fatores externos
 - Fatores internos
 - Custos de Manutenção
 - Qualidade de código



Qualidade de Software

Engenharia de Software busca

*produção de software de boa qualidade
e de forma sistemática.*

Fases:

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



Fatores de qualidade de software

- Dois tipos de fatores influenciam a qualidade de um software:
 - **Fatores externos** são aqueles com os quais os usuários interagem.
 - **Fatores internos** são aqueles com os quais os programadores interagem.



Fatores externos de qualidade

- **Correção:**

- Executar exatamente sua função, conforme definida pelos requisitos e especificação: garante o funcionamento no que foi previsto.

- **Robustez:**

- Funcionar mesmo em condições anormais: garante funcionamento em situações imprevistas
- *Graceful degradation* (degradação suave): sistema continua a operar mesmo no caso de falha de alguns dos seus componentes.



Fatores externos de qualidade

- **Extensibilidade:**

- Medida da facilidade com que o software pode ser adaptado para atender a mudanças na sua especificação.
- Facilitada por projetos simples e módulos autônomos.

- **Reusabilidade:**

- Ser usável em novas aplicações.

- **Compatibilidade:**

- Medida da facilidade com que um software pode ser combinado com outros.
- O uso de padronização facilita a compatibilidade.



Fatores externos de qualidade

- **Eficiência:**

- Relacionado ao desempenho. Consiste no bom uso dos recursos de hardware.

Correção × eficiência

Para que serve um sistema muito eficiente, mas que produz saídas incorretas?

- Correção é recurso prioritário.



Fatores externos de qualidade

- **Portabilidade:**

- Medida da facilidade de transporte de um software para vários hardwares ou ambientes de programação.

- **Facilidade de uso:**

- Preparação de dados de entrada, interpretação de resultados, recuperação de erros de uso, etc.



Fatores externos de qualidade

- **Segurança (integridade):**
 - Proteção dos componentes contra acesso ou modificação não-autorizada.
- **Verificabilidade:**
 - Medida da facilidade de se preparar procedimentos de aceitação, dados para testes, etc, durante a fase de validação ou de operação.



Fatores internos de qualidade

- Fatores externos são os percebidos pelos usuários → *importante*.
- Mas só podem ser atingidos por meio dos fatores internos, como:
 - Modularidade;
 - Legibilidade;
 - Manutenibilidade.



Legibilidade

- **Legibilidade:**

- Facilidade de entender o código-fonte
- Exemplos: nomeação de componentes, organização do código.
- *Inteligibilidade*
 - Facilidade de compreender a coerência de um software em um nível superior do que a legibilidade proporciona.



Manutenibilidade

- **Manutenibilidade:**

- Facilidade para modificar um software ou corrigir problemas.
- Refere-se à facilidade, precisão, segurança e economia na execução de ações de manutenção nesse sistema ou produto.
- Finalidades:
 - Corrigir defeitos.
 - Adequar-se a novos requisitos.
 - Adequar-se a um ambiente novo.



Modularidade

- Divisão do sistema em partes distintas.
- Ideia da “caixa preta”: mapeamento de entradas em saídas.
- **Módulo:** grupo de comandos com uma funcionalidade bem definida e o mais independente possível em relação ao resto do algoritmo.
 - deve haver um módulo principal de controle e módulos específicos para as funções do sistema.



Modularidade

- O ideal é que os módulos não sejam grandes demais.
 - Facilidade de compreensão.
 - Isolamento de função.
- Vantagens:
 - Isolamento para testes e manutenção.
 - Interface pública uniforme.
 - Criação de bibliotecas reutilizáveis.
 - Economia de memória.



Custos de manutenção

- Manutenção é modificação para atender mudanças do mundo externo.
- Manutenção também é remoção dos erros que não deveriam estar lá.
- Cerca de 70% do custo de um software é manutenção.
- A maioria dos erros apresentados por um software são gerados na fase de projeto.



Custos chave do desenvolvimento de software

	Passado	Presente
Ciclos de <i>release</i>	longos	curtos
Requisitos	detalhados	básicos
Interdependências	poucas	muitas
Taxa de mudança	Trimestral/ Anual	Diária/ Semanal
Planejamento	Completo	Ad-hoc
Pontos de coordenação	poucos	muitos
% do tempo em código de outra pessoa	< 25%	> 75%
Manutenção como % do custo total	< 50%	> 50%

Adaptado de Krugle, *The 4 hidden Costs of Code Maintenance*

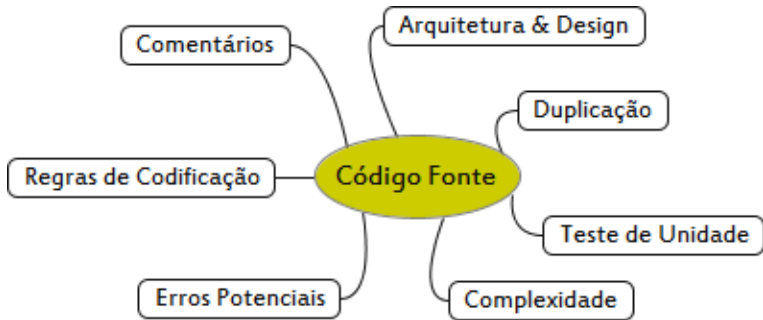


Exemplo de qualidade de código

```
int calcSubsidio() {  
    int subsidio;  
    if (!idade) {  
        // não acima do limite  
        if (!sub) { //sem subsidio  
            if (!naoIntegral) {  
                // não tempo parcial  
                subsidio = 500;  
            } else {  
                subsidio = 250;  
            }  
        } else { // já subsidiado  
            subsidio = 250;  
        }  
    } else { // acima do limite  
        subsidio = -1;  
    }  
    return subsidio;  
}
```

```
int calcSubsidio() {  
    int subsidio;  
    if (eMaior) {  
        subsidio = REJEITA_MAIOR;  
    } else if (jaSubsidiado) {  
        subsidio = JA_SUBSIDIADO;  
    } else if (eTempoParcial) {  
        subsidio = TEMPO_INTEGRAL*FATOR;  
    } else {  
        subsidio = TEMPO_INTEGRAL;  
    }  
    return subsidio;  
}
```

7 pecados capitais do desenvolvimento de software



(SonarQube, 2014)

Finalizando

Martin Fowler



“Any fool can write code that a computer can understand...

But only good programmers write code that humans can understand.”