

Exercícios

Aula 03

1) Qual o objetivo dos métodos, ferramentas e procedimentos utilizados em um software?

Produzir software de alta qualidade.

2) O que abrange a Qualidade de software ?

Métodos e ferramentas de análise, revisões técnicas, estratégias de testes, controle da documentação e adequação nos padrões de desenvolvimento.

3) O que são requisitos do software?

Os requisitos do software compõem a base a partir da qual a qualidade do mesmo é medida. Logo, a falta de conformidade aos requisitos demonstra também falta de qualidade.

4) Quais são os fatores da Qualidade de software e como são medidos?

Os fatores de qualidade de software geralmente são focados em três aspectos importantes: suas características operacionais, sua facilidade de manutenção e sua adaptabilidade a outro ambiente.

Podem ser medidos de diretamente – como erros, KLOC (*kilo lines of code*) ou unidades de tempo – ou indiretamente – como usabilidade ou experiência de usuário.

5) Quais são as descrições que representam os fatores de Qualidade de software?

- Corretude: à medida que um programa satisfaz sua especificação e cumpre os objetivos;
- Confiabilidade: à medida que se pode esperar que um programa execute sua função pretendida com precisão exigida;
- Eficiência: quantidade de recursos de computação e de código exigida para que um programa execute sua função;
- Integridade: à medida que o acesso ao software ou os dados por pessoas não-autorizadas é controlado;
- Usabilidade: esforço para aprender, operar, preparar a entrada e interpretar a saída de um programa;
- Manutenibilidade: esforço exigido para localizar e reparar erros num programa;
- Flexibilidade: esforço exigido para modificar um programa operacional;
- Testabilidade: esforço exigido para testar um programa a fim de garantir sua execução;
- Portabilidade: esforço exigido para transferir o programa de um ambiente para outro;
- Reusabilidade: um programa pode ser reusado por outros programas;
- Interoperabilidade: esforço exigido para se acoplar um sistema a outro;
- Concisão: compactação do programa em termos de linhas de código;
- Segurança: controle que protege programa e dados;
- Simplicidade: o quanto um programa pode ser entendido sem dificuldade;
- Modularidade: independência funcional dos componentes;
- Expansibilidade: o quanto o projeto de arquitetura, procedimental e de dados podem ser ampliados;
- Inteireza: o quanto a implementação total da função requerida foi conseguida;
- Autodocumentação: o quanto o código-fonte apresenta documentação significativa.