

CI 221

DISCIPLINA: Engenharia de Software

AULA NÚMERO: 17

DATA: ____/____/____

PROFESSOR: Andrey

APRESENTAÇÃO

Nesta aula serão apresentados e discutidos os conceitos de Qualidade de Software

DESENVOLVIMENTO

QUALIDADE DE SOFTWARE

Definição:

Qualidade de software pode ser definida como a satisfação dos requisitos funcionais, dos requisitos de desempenho explicitamente declarados, normas de desenvolvimento explicitamente documentadas e características que são esperadas em todo software desenvolvido profissionalmente.

Fatores de Qualidade de Software segundo McCall:

- Correção: Quando um programa satisfaz sua especificação e preenche os objetivos da missão do cliente
- Confiabilidade: Quando se pode esperar que um programa realize a função pretendida com a precisão exigida.
- Eficiência: Quantidade de recursos de computação e código necessário para um programa realizar sua função
- Integridade: Quanto do acesso ao software ou dados por pessoas não autorizadas pode ser controlado.
- Usabilidade: O esforço necessário para aprender, operar, preparar entradas e interpretar saídas de um programa.
- Manutenibilidade: O esforço necessário para localizar e consertar um erro em um programa.
- Flexibilidade: O esforço necessário para modificar um programa operacional.
- Testabilidade: Esforço necessário para testar um programa, a fim de garantir que ele realize a função desejada.
- Portabilidade: Esforço necessário para transferir o programa de um ambiente de hardware ou software para outro.
- Reutilização: Quanto de um programa pode ser reusado em outros programas.
- Interoperabilidade: Esforço necessário para acoplar um programa a outro

Fatores de Qualidade segundo ISO 9126

- Funcionalidade: Grau que o software satisfaz as necessidades declaradas, conforme indicadas pelos seguintes atributos: adequabilidade, precisão, interoperabilidade, atendibilidade e segurança.
- Confiabilidade: Período de tempo em que o software está disponível para uso, conforme indicado pelos seguintes atributos: maturidade, tolerância a falhas e recuperabilidade.
- Usabilidade: Grau em que o software é fácil de usar, conforme: inteligibilidade, facilidade de aprendizado e operabilidade.

- **Eficiência:** Grau em que o software faz uso otimizado dos recursos do sistema, conforme: comportamento em relação ao tempo, comportamento em relação aos recursos.
- **Manutenibilidade:** Facilidade com a qual podem ser feitos reparos no software, conforme: legibilidade, mutabilidade, estabilidade e testabilidade.
- **Portabilidade:** Facilidade com a qual o software pode ser transposto de um ambiente para outro, conforme: adaptabilidade, instalabilidade, conformidade e permutabilidade.

Garantia da qualidade de software

Consiste no conjunto de funções para auditar e relatar que avalia a efetividade e completeza das atividades de controle de qualidade.

A meta da garantia da qualidade é fornecer à gerência os dados necessários para que fique informada sobre a qualidade do produto.

Controle de qualidade

Controle de qualidade envolve uma série de inspeções, revisões e testes ao longo do processo de software para garantir que cada produto de trabalho satisfaça os requisitos estabelecidos. O controle de qualidade inclui um ciclo de realimentação do processo.

Custo da qualidade

Custos de prevenção: planejamento de qualidade, revisões, técnicas formais, equipamentos de teste e treinamento.

Custos de avaliação: avaliação da condição do produto na primeira execução. Inspeção inter e intra processos, calibração e manutenção do equipamento e teste.

Custos de falha: Falhas internas (falhas antes de entregar o produto) Falhas externas (falhas após entregar o produto)

Atividades SQA

Para se conseguir a garantia da qualidade de software SQA devem ser executadas algumas atividades:

- Aplicação de métodos técnicos
- realização de revisões técnicas formais
- Atividades de teste
- aplicação de padrões
- controle de mudança
- medição
- manutenção de registros

Revisões e revisões formais

São aplicadas em vários pontos do desenvolvimento para descobrir defeitos que possam ser eliminados. Técnicas: Walkthrough, inspeções e revisões round-robin

Objetivos:

- (1) descobrir erros na função, lógica ou na implementação para qualquer representação do software;
- (2) verificar se o software satisfaz seus requisitos;
- (3) garantir que o software tenha sido representado de acordo com os padrões;
- (4) conseguir que o software seja desenvolvido de maneira uniforme e

(5) tornar os projetos mais administráveis.

Tipos de revisões

a) Prova de corretude: Especificações em Z

b) garantia estatística de Qualidade

Coletar informações sobre os defeitos, rastrear os defeitos até suas causas, Princípio de Pareto (80% dos defeitos se remetem a 20% das causas), identificar as 20% das causas, corrigir as causas que geram os defeitos.

c) Seis Sigma: Motorola 1980

Definir os requisitos do cliente; medir o processo para coletar os defeitos; determinar as poucas causas vitais, aperfeiçoar o processo pela eliminação das causas vitais, controlar o processo para que ele não inclua novos defeitos.

d) Sala Limpa

É uma abordagem que enfatiza a necessidade de se fazer a correção do software a medida que ele vai sendo construído. Em vez de fazer o produto e depois remover os defeitos, é proposto que sejam eliminados os defeitos na especificação e no projeto para poder fabricar o produto de maneira limpa.

ATIVIDADE

1. Descreva a qualidade funcionalidade.
2. Qual a diferença entre a manutenibilidade na visão de McCall e da ISO?
3. Discuta as diferenças entre a funcionalidade e a eficiência.
4. Descreva em suas palavras quais os fatores principais de qualidade de software

BIBLIOGRAFIA BÁSICA

PRESSMAN, R. S.. Engenharia de Software. 6a. ed. McGraw-Hill. 2006

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.. UML guia do usuário. Editora Campus. 2000.

BEZERRA, E.. Princípios de Análise e Projeto de Sistemas com UML. Editora Campus. 2003.

CARVALHO, A. M. B. R.; CHIOSSI, T. C. S.. Introdução à engenharia de software. Editora da Unicamp. 2001.