

SOFT

DISCIPLINA: Engenharia de Software

AULA NÚMERO: 19

DATA: \_\_\_\_/\_\_\_\_/\_\_\_\_

PROFESSOR: Andrey

## APRESENTAÇÃO

O objetivo desta aula é apresentar e discutir conceitos relacionados a Manutenção de Software.

## DESENVOLVIMENTO

### Conceito de manutenção

Os sistemas de software frequentemente necessitam passar por mudanças. Os motivos para essas mudanças podem ser:

- erros não detectados nos testes,
- mudanças nos requisitos,
- evolução da plataforma de software/hardware,
- evolução do sistema devido a novas necessidades dos usuários entre outras.

A atividade de efetuar essas mudanças de forma controlada e organizada dá-se o nome de manutenção de software. Segundo Pressman (2006) estima-se que 60 % do esforço de uma empresa de desenvolvimento de software é gasto em manutenção de sistemas já existentes. Isto se deve em parte ao fato que grande parte do software tem em média entre 10 e 15 anos. Este software necessita se adaptar a novas plataformas e evoluir para atender a novas necessidades do negócio.

### Atividades de manutenção

Segundo Pressman (1992), existem 4 atividades de manutenção. São elas:

- manutenção corretiva,
- manutenção adaptativa,
- manutenção evolutiva e
- manutenção preventiva.

Mesmo efetuando-se a atividade de teste de forma correta, é comum que alguns erros só sejam descobertos com o uso do sistema pelo usuário final. A manutenção corretiva é o processo que inclui o diagnóstico e a correção de erros do programa após o programa já ter sido entregue.

As plataformas de software e hardware estão em rápida evolução. A cada ano são lançados novas gerações de computadores, periféricos, sistemas operacionais e aplicativos com os quais o sistema interage. A manutenção adaptativa modifica o software para que ele tenha uma interface adequada com este ambiente mutante.

As características do negócio e as necessidades dos usuários se modificam ao longo da vida útil do sistema. Novas capacidades e novas funcionalidades são requeridas. Se o sistema não evolui para atender a essas mudanças ele se torna obsoleto. A manutenção evolutiva é a atividade de modificar o sistema para atender a essas requisições. Esta atividade é responsável pela maior parte do esforço de manutenção.

A manutenção preventiva ocorre quando o software é modificado para melhorar características de confiabilidade ou manutenibilidade futuras. Esta atividade é caracterizada pelo uso das técnicas de engenharia reversa e reengenharia.

### Características da manutenção (fluxo de atividades, custo e problemas)

Quando o software não foi desenvolvido de forma organizada, de acordo com uma metodologia de desenvolvimento e este não foi bem documentado a manutenção não pode ser feita de forma estruturada. Uma manutenção “Estruturada” é composta dos seguintes passos:

1. Avaliar a documentação existente,
2. planejar a abordagem de manutenção,
3. modificar o projeto,
4. modificar o código,
5. revisar e efetuar testes de regressão.

O esforço de manutenção pode ser dividido em atividades produtivas como análise, avaliação, modificação do projeto, codificação e atividades “quebra-cabeça” como por exemplo, tentar compreender a estrutura de dados, características de interface ou limites de desempenho. O custo de manutenção, segundo Belady e Lehman (72) pode aumentar exponencialmente se a abordagem usada no desenvolvimento foi ruim e se a pessoa que desenvolveu não está disponível para realizar a manutenção.

Problemas com a manutenção:

- Dificuldade de rastrear o processo através do qual o software foi criado.
- Dificuldade de rastrear a evolução do software através de suas muitas versões.
- Dificuldade em compreender o programa que outra pessoa fez.
- Falta de disponibilidade da pessoa que desenvolveu o software.
- Documentação do desenvolvimento do software inexistente ou muito ruim
- Software não projetado para receber mudanças. Pouca manutenibilidade.
- Frustração associada à atividade de manutenção.

## **Manutenibilidade**

Conceito de manutenibilidade. Manutenibilidade de software pode ser definida qualitativamente como a facilidade com que um software pode ser entendido, corrigido, adaptado e/ou aumentado.

Um processo de desenvolvimento mal feito tem um impacto negativo sobre a manutenibilidade de um software. Uma configuração de software ruim tem um impacto negativo semelhante.

Fatores que afetam a manutenibilidade:

- cuidado inadequado com o projeto, codificação e teste
- configuração de software ruim
- disponibilidade de pessoal qualificado de software
- facilidade de manusear o sistema
- uso de linguagens de programação padronizadas
- uso de sistemas operacionais padronizados
- estruturas padronizadas de documentação
- disponibilidade de um computador próprio para a manutenção
- disponibilidade da pessoa ou grupo que desenvolveu o software
- o planejamento para manutenibilidade (fator mais importante que afeta a manutenibilidade)

medidas de manutenibilidade:

Pode-se determinar a manutenibilidade indiretamente considerando-se atributos das atividades de manutenção que podem ser medidos revisões

A manutenibilidade pode ser medida indiretamente considerando medidas da estrutura do projeto e medidas da complexidade do software

## **Tarefas da manutenção (organização, relatórios, fluxo de eventos, registros, avaliação)**

1. Estabelecer uma organização para a manutenção (“de fato” ou formal)
2. Descrever procedimentos de avaliação e de comunicação
3. Definir sequencias padronizadas de eventos (para os pedidos de manutenção)
4. Estabelecer procedimentos para registrar a história das atividades de manutenção
5. Definir critérios de revisão e avaliação

## **Efeitos colaterais da manutenção**

Codificação

Dados

Documentação

## Manutenção de código alienígena

Os programas “alienígenas” são assim chamados porque:

- Programas com fluxo de controle equivalente a um prato de espaguete
- Módulos muito grandes
- Poucas linhas de comentários significativos
- Não existe nenhum outro elemento da configuração de software, além do código.
- Nenhum membro do pessoal atual de manutenção trabalhou no desenvolvimento do programa.
- Nenhuma metodologia de desenvolvimento foi aplicada:
  - Projeto de dados e projeto arquitetural ruins
  - Documentação e registro histórico das alterações incompletos

## BIBLIOGRAFIA BÁSICA

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

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

OLIVEIRA, M. Técnicas de Testes Caixa-Preta. Apresentação no II Encontro Brasileiro de Teste de Software, outubro 2007.