

ESTUDO DIRIGIDO - 20/10/2023

Evolução de Software

INSTRUÇÕES

Com base nos conhecimentos apresentados na aula, e pesquisas relacionadas, realize os exercícios a seguir:

Após, encaminhe suas as respostas em um arquivo no **Classroom**.

DATA DE ENTREGA: até 27/10/2023

EXERCÍCIO 01

A) - Explique por que um sistema de software utilizado em um ambiente do mundo real deve mudar ou se torna cada vez menos útil.

Pois um software desatualizado possui várias desvantagens, ele se torna obsoleto, ineficiente, inseguro e incapaz de atender às necessidades em constante evolução de seus usuários e do ambiente operacional. Atualizações são importantes, pois a partir delas, é possível:

- Melhorar a usabilidade
- Correção de falhas internas
- Desbloquear novas funcionalidades

B) - Explique por que os sistemas legados devem ser considerados sistemas sociotécnicos em vez de simplesmente sistemas de software que foram desenvolvidos com tecnologia antiga.

Porque são produtos de uma interação complexa entre tecnologia, pessoas, processos, cultura e contexto organizacional. Mudar ou substituir esses sistemas envolve não apenas desafios técnicos, mas também desafios sociais e organizacionais significativos.

C) - Quais são as opções estratégicas para evolução dos sistemas legados? Quando você substituiria normalmente todo ou parte de um sistema em vez de continuar a manutenção do software?

Manutenção (corretiva, adaptativa e evolutiva), a modernização de componentes obsoletos do sistema, a reengenharia, revisão, reescrita e reestruturação do sistema legado. A substituição de todo ou parte de um sistema em vez de continuar a manutenção do software geralmente é apropriada quando o sistema legado enfrenta

desafios significativos relacionados à tecnologia, custo, segurança, conformidade, desempenho ou capacidade de atender às necessidades atuais.

D) - Descreva resumidamente os três tipos principais de manutenção de software. Por que às vezes é difícil distinguir entre eles?

- Manutenção corretiva: Correção de bugs e problemas de segurança, mas não envolve a introdução de novos recursos ou funcionalidades.
- Manutenção adaptativa: Fazer ajustes no sistema para mantê-lo funcionando.
- Manutenção evolutiva: Adição de novos recursos e funcionalidades ao sistema legado.

As vezes é difícil distinguir entre eles pois as atividades de manutenção muitas vezes se sobrepõem, pois a evolução de um software envolve frequentemente correções, adaptações e melhorias. Por exemplo:

- Uma melhoria de desempenho (manutenção adaptativa) pode ser vista como uma adição de funcionalidade que melhora a experiência do usuário (manutenção evolutiva).
- A inclusão de novos recursos (manutenção evolutiva) pode exigir correção de bugs que surgem durante o desenvolvimento desses recursos (manutenção corretiva).

E) - Explique as diferenças entre reengenharia e refatoração de software.

A reengenharia de software é uma abordagem mais radical que visa transformar significativamente o software existente, geralmente com o propósito de substituir ou modernizar um sistema legado. Ela visa uma transformação profunda do software para melhorar sua arquitetura, qualidade, desempenho e funcionalidades.

Enquanto a refatoração é uma abordagem mais cirúrgica que visa melhorar a qualidade do código sem alterar sua funcionalidade. Ela visa tornar o código mais limpo, organizado, manutenível e eficiente.

EXERCÍCIO 02

A) - Pesquise a respeito do termo “*bad smell*” na engenharia de software, explique sobre o que trata e liste no mínimo 05 categorias explicando sucintamente cada uma delas.

Se refere a padrões ou características de código que indicam problemas ou más práticas que podem afetar a qualidade, manutenibilidade e eficiência do software. São indícios de que algo no código pode estar errado, desorganizado ou ineficiente.

- Duplicação de Código: Refere-se à presença de código idêntico ou semelhante em várias partes do sistema.
- Complexidade Excessiva: Isso ocorre quando um trecho de código é excessivamente complexo, com muitos desvios condicionais, aninhamentos profundos ou lógica difícil de entender.

- Métodos ou Funções Longas: Quando uma função ou método é muito longo, torna-se difícil de compreender e manter. As funções devem ser curtas e focadas em tarefas específicas
- Acoplamento Excessivo: O acoplamento excessivo ocorre quando módulos, classes ou componentes de software estão fortemente interligados.
- Falha em Seguir Padrões de Codificação: Quando os desenvolvedores não seguem as diretrizes ou padrões de codificação estabelecidos pela equipe ou pela organização, isso pode resultar em códigos inconsistentes e de difícil manutenção.

B) - Pesquise sobre as “Leis de Lehman”, descreva e comente a respeito.

São leis que descrevem padrões de evolução de sistemas de software ao longo do tempo. Elas fornecem um quadro conceitual para entender como os sistemas de software evoluem e por que a manutenção contínua e a gestão de mudanças são importantes.

Essas leis são importantes para conseguirmos entender os motivos pelos quais os sistemas de software não podem se manter estagnados e devem evoluir, mas que também devemos equilibrar a evolução do software com a manutenção do software, que devemos manter os dois juntos e não focar em apenas um deles.