Introdução à Engenharia de Software

Atividades de Aprendizagem e Avaliação

Aluno: Lucas Gabriel Schumann Garcia RA: 2206293

Use esta cor em seu texto

1. Considerando o conteúdo no link “Engenharia de Software – Preambulo”, complete
   a) Engenharia de Software é uma área da Computação dedicada a investigar os desafios e propor soluções que permitam desenvolver sistemas de software

   b) Engenharia de Software trata da aplicação de abordagens sistemáticas, disciplinadas e quantificáveis para desenvolver, operar, manter e evoluir software.

   c) A Engenharia de Software surgiu da necessidade de sistemas comerciais, como folha de pagamento, controle de clientes, controle de estoques, etc.

   d) Segundo Brooks existem dois tipos de dificuldades em Desenvolvimento de Software

      1) Dificuldades acidentais – relacionadas à área do problema

      2) Dificuldades essenciais – relacionadas à tecnologia

   e) São dificuldades essenciais: a complexidade; a conformidade; a facilidade de mudanças; e a Invisibilidade

   f) As 12 áreas de Engenharia de Software são:

      Engenharia de Requisitos

      Projeto de Software

      Construção de Software

      Testes de Software

      Manutenção de Software

Gerência de Configuração

Gerência de Projetos

Processos de Software

Modelos de Software

Qualidade de Software

Prática Profissional

Aspectos Econômicos

g) Os requisitos funcionais de um sistema definem o que o sistema deve fazer.

h) Os requisitos não funcionais de um sistema definem como o sistema deve operar.

i) O projeto de um Sistema de Software se inicia pela definição de suas principais unidades de código no nível de interfaces (módulos)

j) **Interfaces providas** se relacionam com serviços que uma unidade de código torna público para uso pelo resto do sistema

k) **Interfaces requeridas** se relacionam com interfaces das quais uma unidade de código depende para funcionar.

l) A **Arquitetura de Software** trata da organização de um sistema em um nível de abstração mais alto do que aquele que envolve classes ou construções semelhantes.

m) Testes de software mostram a presença de bugs mas não a sua ausência.

n) Testes de usabilidade objetivam verificar a usabilidade do sistema.

o) Os **testes** podem ser usados para **verificação** com o objetivo de verificar se o sistema atende sua especificação, ou para **validação** com o objetivo de garantir que atende as necessidades dos seus clientes.

p) **Defeitos** são problemas técnicos com resultado indesejado, já **falhas** ocorrem quando um código com defeito for executado.

q) Nem todo defeito resulta em uma falha pois pode acontecer que o código defeituoso nunca seja executado.

r) O defeito no código do foguete ‘Ariane 5’ estava relacionado com a conversão de um número em ponto flutuante de 64 bits para um número inteiro com 16 bits, a falha ocorreu quando o código com defeito foi executado.

s) As manutenções de software podem ser classificadas em: corretiva, preventiva, adaptativa, refactoring e evolutiva.

t) Manutenção adaptativa tem por objetivo adaptar um sistema a uma mudança em seu ambiente.

u) *Refactoring* é um tipo de manutenção que tem por objetivo a melhoria do seu código e/ou projeto.

v) Gerência de Configuração se relaciona com o conjunto de políticas para gerenciar as versões de um sistema.

w) A **Lei de Brooks** é: “A inclusão de novos desenvolvedores em um projeto que está atrasado contribui para torna-lo ainda mais atrasado”

x) A gerência de Projetos se ocupa de atividades tais como: prazos; aprofundamento; etc.

y) Um PROCESSO DE DESENVOLVIMENTO DE SOFTWARE define quais etapas devem ser seguidas para construir e entregar um sistema de software.

z) Processos ***Waterfall*** (em cascata) foram inspirados nos processos usados em engenharias tradicionais e são largamente sequenciais.

aa) As etapas de um processo de software em cascata são:

1) Levantamento de requisitos
2) Análise
3) Projeto
4) Codificação
5) Testes
6) Implantação

bb) O **Manifesto Ágil** foi produzido em fevereiro no ano de 2001 por um grupo de 17 engenheiros de software.

cc) A principal característica de um PROCESSO DE DESENVOLVIMENTO DE SOFTWARE ÁGIL é que um sistema deve ser construído de forma incremental e iterativa.

dd) XP, Scrum, Kanban e Lean Development são exemplos de processos ágeis.

ee) A **Integração Contínua** recomenda que desenvolvedores integrem o código que produziram de maneira imediata, se possível todo dia.

ff) Modelos criados para entender um sistema já implementado são instrumentos de engenharia reversa.

gg) A **Qualidade de Software** pode ser avaliada em duas dimensões:

1) Qualidade externa
2) Qualidade interna

hh) A qualidade externa considera fatores que podem ser aferidos sem analisar o código. Pode ser avaliado por usuários comuns.

ii) Conceitue

1) Robustez O software deve continuar funcionando mesmo com falhas externas, ou, ao menos informar o motivo de alguma falha (crash).
2) Eficiência Fazer bom uso dos recursos computacionais, de maneira inteligente, sem necessariamente necessitar de um hardware caro e custoso para funcionar.

jj) A qualidade interna considera propriedades e características relacionadas a implementação do sistema.

kk) São exemplos de atributos da qualidade interna

1) Modularidade
2) Legibilidade
3) Manutenibilidade
4) Testabilidade

ll) Cite um exemplo de métrica de processo: Número de linhas de um código.

mm) Revisões de código tem por objetivo detectar bugs e disseminar boas práticas de Engenharia de Software.

nn) ***Over-engineering*** é usar ferramentas e recursos desnecessários e sofisticados de mais para o propósito do sistema.