

Engenharia de Software - COM390 - Turma 001

Página Inicial

Avisos

Cronograma

Atividades

Fóruns

Collaborate

Calendário Lives

Notas

Menu das Semanas

Semana 1

Semana 2

Semana 3

Semana 4

Semana 5

Semana 6

Semana 7

Semana 8

Orientações para realização da prova

Orientações para realização do exame

Documentos e informações gerais

Gabaritos

Referências da disciplina

Facilitadores da disciplina

Repositório de REAs

Revisar envio do teste: Semana 5 - Atividade Avaliativa

Usuário

LIZIS BIANCA DA SILVA SANTOS

Curso

Engenharia de Software - COM390 - Turma 001

Teste

Semana 5 - Atividade Avaliativa

Iniciado

10/09/24 14:15

Enviado

10/09/24 14:32

Data de vencimento

11/09/24 23:59

Status

Completada

Resultado da tentativa

10 em 10 pontos

Tempo decorrido

16 minutos

Instruções

Olá, estudante!

1. Para responder a esta atividade, selecione a(s) alternativa(s) que você considerar correta(s);

2. Após selecionar a resposta correta em todas as questões, vá até o fim da página e pressione “Enviar teste”.

3. A cada tentativa, as perguntas e alternativas são embaralhadas

Pronto! Sua atividade já está registrada no AVA.

Resultados exibidos

Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente

Pergunta 1

1,44 em 1,44 pontos

As técnicas de teste auxiliam na seleção de casos de teste que podem ser utilizados para testes de *software*. O objetivo é selecionar, com base nos dados de entrada, casos de teste, tendo em vista a impossibilidade de testar todos os cenários possíveis, que permitam a maior cobertura possível.

Assinale a alternativa que indica as técnicas de teste conhecidas como testes baseados na especificação de requisitos.

Resposta Selecionada:

c. Testes de caixa preta.

Respostas:

a. Testes de caixa cinza.

b. Testes de unidade.

c. Testes de caixa preta.

d. Testes de caixa branca.

e. Testes de integração.

Comentário da resposta:

JUSTIFICATIVA

Os testes de caixa preta consistem nos testes funcionais, que visam verificar as funcionalidades do *software*. São conhecidos como testes baseados na especificação. Os testes de unidade e testes de integração visam testar códigos de unidades e suas integrações, não sendo conhecidos como testes baseados na especificação, Os testes de caixa branca são conhecidos como testes estruturais e visam a testar a estrutura interna do *software*. Os testes de caixa cinza são uma combinação dos testes de caixa preta e caixa branca.

Pergunta 2

1,44 em 1,44 pontos

Os casos de teste contêm procedimentos para a realização do teste, apresentando os dados de entrada que serão utilizados e o resultado esperado. Uma forma de selecionar casos de testes é por meio do uso de técnicas como os testes de caixa preta e os testes de caixa branca.

Nesse contexto, avalie as asserções a seguir e a relação proposta entre elas.

I. Os testes de unidade podem fazer uso das técnicas de testes estruturais, que são conhecidos como testes de caixa branca.

PORQUE

II. Os testes estruturais são técnicas de seleção de dados de entrada baseadas na estrutura interna do código a ser testado.

A respeito dessas asserções, assinale a alternativa correta.

Resposta Selecionada:

e. As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.

Respostas:

a. As asserções I e II são falsas.

b. As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.

c. A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.

d. A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.

e. As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.

Comentário da resposta:

JUSTIFICATIVA

A asserção I é verdadeira, pois os testes de unidade visam testar uma unidade, que pode ser um componente, um módulo de *software*, com o objetivo de testar a estrutura do código utilizando técnicas de testes estruturais.

A asserção II é verdadeira e justifica a proposição I, pois os testes estruturais, também conhecidos como testes de caixa branca, visam testar a estrutura interna do código por meio dos caminhos de seleção, das estruturas de controle e do fluxo de dados, aplicáveis aos testes de unidade.

Pergunta 3

1,42 em 1,42 pontos

O sistema de notas deve indicar se um aluno está apto ou não para realizar um exame de recuperação com base na média final obtida. A regra é que somente notas de média final igual ou acima de 4,0 e inferiores a 7,0 podem realizar o exame de recuperação. Os que apresentam nota da média final inferior a 4,0 estão, automaticamente, reprovados; e os que obtêm nota da média final acima ou igual a 7,0 estão, automaticamente, aprovados.

Assinale a alternativa que corresponde ao valor de notas a ser usado para os casos de testes utilizando análise de valor limite para essa funcionalidade.

Resposta Selecionada:

d. 3, 4, 5, 6, 7, 8.

Respostas:

a. 0, 3, 4, 5, 6, 7.

b. 3, 4, 5, 7, 9, 10.

c. 2, 5, 6, 7, 9, 10.

d. 3, 4, 5, 6, 7, 8.

e. 1, 2, 3, 4, 5, 6.

Comentário da resposta:

JUSTIFICATIVA

Na análise de valor limite, são utilizados valores para testar os limites das classes de equivalência. As classes de equivalência podem ser divididas em: classe de equivalência I: nota de média final > 4 (reprovado); classe de equivalência II: nota média final >=4 e < 7 (recuperação); classe de equivalência III: nota da média final >= 7 (aprovado). Dessa forma, as notas de média final válidas para a recuperação são 3, 4 e 5, do limite entre as classes de equivalência I e II; e 6, 7 e 8, que estão no limite entre as classes de equivalência II e III. As notas 0, 1, 2, 9 e 10 não estão nos limites das classes de equivalência.

Pergunta 4

1,42 em 1,42 pontos

Assinale a alternativa que mostra corretamente a ordem em que as etapas (ou níveis) de teste devem ser executados:

Resposta Selecionada:

unidade → integração → validação → sistema.

Respostas:

unidade → integração → validação → sistema.

unidade → integração → sistema → validação.

validação → sistema → integração → unidade.

validação → sistema → unidade → integração.

sistema → validação → integração → unidade.

Comentário da resposta:

JUSTIFICATIVA

A única ordem correta é aquela apresenta na alternativa correta; todas as outras não fazem sentido considerado a teoria associado a estratégias de teste de software.

Pergunta 5

1,42 em 1,42 pontos

Marque a alternativa abaixo que apresenta apenas exemplos de técnicas de teste funcional:

Resposta Selecionada:

Baseado em grafos; Particionamento de equivalência; Análise de valores limite; Teste de matriz ortogonal; Baseado em modelos.

Respostas:

Baseado em grafos; Particionamento de equivalência; Análise de valores limite; Teste de matriz ortogonal; Baseado em modelos.

Teste do caminho básico; Baseado em grafos; Particionamento de equivalência; Análise de valores limite; Baseado em modelos.

Teste do caminho básico; Baseado em grafos; Particionamento de equivalência; Análise de valores limite; Teste de matriz ortogonal.

Teste do caminho básico; Baseado em grafos; Particionamento de equivalência; Teste de matriz ortogonal; Baseado em modelos.

Teste do caminho básico; Particionamento de equivalência; Análise de valores limite; Teste de matriz ortogonal; Baseado em modelos.

Comentário da resposta:

JUSTIFICATIVA

Teste do caminho básico é uma técnica de teste estrutural.

Pergunta 6

1,43 em 1,43 pontos

Uma equipe de testes está em fase de execução dos testes de sistema, sendo que uma série de defeitos tem sido reportada para o time de desenvolvimento. Uma prática comum dessa fase de testes é retestar todos os testes funcionais já executados a cada nova versão do *software* com as correções incluídas, porém isso tem se mostrado pouco efetivo na detecção de novos defeitos. Além disso, outros testes precisam ser realizados, para assegurar que o sistema possa responder, em um tempo adequado, às solicitações de usuários e para que não haja vulnerabilidades para invasões.

Assinale a alternativa que aponte os tipos de teste adequados para o cenário apresentado.

Resposta Selecionada:

c. Testes de regressão, testes de desempenho e testes de segurança.

Respostas:

a. Testes de regressão, testes de conectividade e testes em condições naturais.

b. Testes de segurança, testes de compatibilidade e testes de *stress*.

c. Testes de regressão, testes de desempenho e testes de segurança.

d. Testes de certificação, testes de fumaça e testes de segurança.

e. Testes de desempenho, testes de usabilidade e testes de segurança.

Comentário da resposta:

JUSTIFICATIVA

Os testes de regressão permitem escolher os testes que devem ser retestados por alto risco de efeitos colaterais de mudanças do *software*. Os testes de desempenho permitem verificar se o tempo de resposta está adequado no processamento das solicitações recebidas. Os testes de segurança visam encontrar defeitos no que tange, principalmente, a potenciais vulnerabilidades do *software*. Os testes de *stress* se referem a uma alta carga sendo demandada ao *software*. Os testes de compatibilidade verificam se o sistema atua corretamente em diferentes combinações de *hardware* e *software*. Os testes de usabilidade verificam o nível de praticidade de uso do *software*. Os testes de conectividade verificam a tolerância do sistema a problemas de conexão com a rede e se o sistema consegue se conectar a quaisquer redes. Os testes de fumaça verificam se a construção do sistema foi bem-sucedida na integração.

Pergunta 7

1,43 em 1,43 pontos

Uma equipe de desenvolvedores está discutindo sobre o planejamento dos testes de unidade, tendo em vista uma grande quantidade de módulos que foram desenvolvidos e o prazo de integração que se aproxima. A equipe está preocupada, pois trata-se de um projeto muito importante e com altas expectativas do cliente.

Assinale a alternativa que apresenta a melhor opção para o time de desenvolvimento em relação ao planejamento dos testes de unidade.

Resposta Selecionada:

c. Selecionar as operações dos componentes com maior complexidade ciclomática.

Respostas:

a. Planejar o teste de todos os componentes e negociar um prazo maior.

b. Pular a etapa de testes de unidade e partir para a integração dos componentes.

c. Selecionar as operações dos componentes com maior complexidade ciclomática.

d. Selecionar os componentes menores, para otimizar a quantidade de testes de unidade.

e. Utilizar o método de teste do caminho básico para o teste de todos os componentes.

Comentário da resposta:

JUSTIFICATIVA

A complexidade ciclomática oferece uma métrica que indica a complexidade lógica de um programa. Testar as operações de componentes com a maior complexidade ciclomática significa testar as operações com maior probabilidade de apresentar defeitos. Tentar planejar os testes de todos os componentes pode apresentar um tempo inviável, sendo que negociar pode ser prejudicial para a imagem do time perante as expectativas dos clientes. Pular a etapa pode postergar a detecção de defeitos em etapas posteriores do projeto, o que pode acarretar um custo de retrabalho maior. Selecionar apenas os componentes menores pode não ser efetivo para garantir um nível de qualidade adequado por parte dos testes unitários. O teste do caminho básico de todos os componentes não garante uma otimização para efetividade em um tempo menor de projeto.

Domingo, 16 de Março de 2025 18h27min27s BRT

← OK