Tratamento de falhas de software

Revisão da tentativa 1

Iniciado em	terça, 9 outubro 2012, 10:42
Completado em	terça, 16 outubro 2012, 10:55
Tempo empregado	7 dias
Notas	29.4/30
Nota	**98** de um máximo de 100(**98**%)

Question1
Notas: 1

De acordo com o artigo "Fighting Bugs: Remove, Retry, Replicate, and Rejuvenate" de Michael Grottke e Kishor S. Trivedi, publicado na IEEE Computer, em fevereiro de 2007, qual a definição de defeito de sistema (*system failure*)?

Escolher uma resposta.

	a. defeito de sistema ocorre quando o comportamento do sistema conduz a perdas de vidas ou danos irremediáveis ao ambiente ✗
	b. defeito de sistema ocorre quando o comportamento real do sistema leva a prejuízos

	financeiros ✗
◉	c. defeito de sistema ocorre quando o comportamento real do sistema se desvia do serviço correto ✓
○	d. defeito de sistema ocorre quando o sistema trava ✗
○	e. defeito de sistema ocorre quando o comportamento real do sistema se desvia da sua especificação ✗

Correto

Notas relativas a este envio: 1/1.

Question2

Notas: 1

Em que fase do ciclo vida do software o reparo de bugs é mais oneroso (ou seja, apresenta maior custo)?

Escolher uma resposta.

○	a. fase de descarte ✗
○	b. fase de aquisição ✗
○	c. fase de teste ✗
○	d. fase de desenvolvimento ✗
◉	e. fase operacional ✓

Correto

Notas relativas a este envio: 1/1.

Question3

Notas: 1

De acordo com Grottke e Tivedi, o que pode tornar uma tarefa muito difícil o diagnóstico e o isolamento das falhas que são responsáveis por um defeito observável ?

Escolher uma resposta.

- a. o fato de que alguns defeitos não podem ser reproduzidos ✓
- b. o fato dos testadores não dominarem estratégias de teste dinâmicas ✗
- c. o fato dos testadores não dominarem estratégias de teste estáticas ✗
- d. o desconhecimento da especificação do sistema por parte dos testadores ✗
- e. o fato dos desenvolvedores documentarem mal o sistema ✗

Correto

Notas relativas a este envio: 1/1.

Question4
Notas: 1

O que são Bohrbugs?

Escolher uma resposta.

- a. falhas que provocam defeitos que são difíceis de serem reproduzidos ✗
- b. falhas que se manifestam consistentemente sob condições bem definidas ✓
- c. falhas que os métodos tradicionais de teste não conseguem detectar ✗

d. falhas que de acordo com Niels Bohr afetam a estrutura dos átomos ✗

Correto

Notas relativas a este envio: 1/1.

Question5

Notas: 1

Determine se a sentença a seguir é verdadeira ou falsa:
"Toda a ativação de uma falha de software sempre causa um defeito imediatamente pois todas as falhas de software são fixas no código."

Resposta:

Verdadeiro ✗ Falso ✓

Correto

Notas relativas a este envio: 1/1.

Question6

Notas: 1

Determine se a afirmação a seguir é verdadeira ou falsa:
"Uma falha de software, que foi ativada pela execução da parte do código onde a falha está localizada, produz uma condição interna no sistema diferente da condição correta. Essa condição interna é conhecida como erro."

Resposta:

Verdadeiro ✓ Falso ✗

Correto

Notas relativas a este envio: 1/1.

Question7

Notas: 1

Um erro pode se desenvolver em outros erros e assim sucessivamente até que um defeito finalmente ocorra. Essa cadeia de eventos é chamada de:

propagação de erros .

Correto

Notas relativas a este envio: 1/1.

Question8

Notas: 1

Uma instrução mal codificada na implementação de um algoritmo pode levar a um valor incorreto para uma variável de um programa. O programa pode usar esse valor incorreto para computações posteriores levando a incorreções adicionais. Posteriormente, um desses valores incorretos pode influenciar o comportamento observável do sistema.

Considerando o exemplo, associe a descrição ao termo correspondente:

execução do trecho de programa que contém a instrução mal codificada:	erro
incorreções conduzindo a novas incorreções:	propagação de erro
valor incorreto em uma variável:	falha
incorreção observável no comportamento do sistema:	defeito
instrução mal codificada:	ativação da falha

Parcialmente correta

Notas relativas a este envio: 0.4/1.

Question9

Notas: 1

Os autores apresentam **duas** explicações para o software se comportar de forma diferente em situações aparentemente idênticas. São elas:

Escolha pelo menos uma resposta.

- [ ] a. usuários com comportamento caótico e difícil de identificar ✗
- [ ] b. propagação do erro invertida, ou seja, um erro provocando uma falha ✗
- [ ] c. ocorrência de falhas randômicas de hardware ✗
- [x] d. outros elementos do sistema podem influenciar o comportamento da falha numa aplicação específica ✓
- [x] e. longo retardo entre a ativação da falha e a ocorrência do defeito ✓

Correto

Notas relativas a este envio: 1/1.

Question10
Notas: 1

Vários elementos do sistema de software, como o sistema operacional e outras aplicações, podem influenciar o comportamento de uma falha. Um exemplo citado no artigo é a sincronização inadequada de múltiplas threads que, dependendo da ordem do escalonamento das threads, pode provocar comportamentos diferentes. Michael Grottke and Kishor Trivedi nomeia o conjunto desses elementos como:

Escolher uma resposta.

- ○ a. condições de temporização do erro ✗
- ○ b. condições de indeterminismo computacional ✗

c. ambiente interno ao sistema da aplicação ✓

d. escalonamento multithreaded ✗

e. sincronização tempo-real ✗

Correto

Notas relativas a este envio: 1/1.

Question11

Notas: 1

Complete os espaços vazios com os termos correspondentes, de acordo com Grottke e Tivedi:

Uma falha pode fazer com que o software exiba um comportamento caótico e até não determinístico com respeito a ocorrência ou não de defeitos. Profissionais da área se referem a falhas com essa propriedade como Mandelbugs. Algumas vezes, na literatura, essas falhas também são chamadas de Heisenbugs, referindo-se ao fato de que tais falhas trocam seu comportamento quando são observadas ou isoladas.

Correto

Notas relativas a este envio: 1/1.

Question12

Notas: 1

Segundo Michael Grottke e Kishor S. Trivedi, qual o principal problema que o desenvolvedor ou engenheiro de teste enfrenta com bugs que são chamados de Mandelbugs?

Escolher uma resposta.

a. grande probabilidade de serem detectados durante o teste e com isso forçarem o programador a corrigir o código ✗

b. se transformarem em Bohrbugs ✗

c. grande probabilidade de não serem detectados durante o teste ✓

d. envelhecerem e ficarem obsoletos ✗

Correto

Notas relativas a este envio: 1/1.

Question13

Notas: 1

Michael Grottke and Kishor Trivedi afirmam ser plausível concluir que a maioria das falhas que permanecem não detectadas em trechos de software bem testados são de apenas um determinado tipo. Qual o tipo?

Escolher uma resposta.

a. falhas de interação ✗

b. Mandelbugs ✓

c. falhas de temporização ✗

d. falhas de sincronização ✗

e. Bohrbugs ✗

Correto

Notas relativas a este envio: 1/1.

Question14

Notas: 1

Qual a percentagem de falhas detectadas atribuídas a Mandelbugs após a entrada em operação de um software?

Escolher uma resposta.

- a. 1 a 8% ✗
- b. 30 a 40% ✗
- c. 90 a 100% ✗
- d. 15 a 80% ✓

Correto

Notas relativas a este envio: 1/1.

Question15
Notas: 1

De acordo com Grottke e Trivedi, qual a maneira simples e eficiente de lidar com falhas que aparentemente exibem comportamento não determinístico (Mandelbugs) e que não seria adequada para falhas do tipo Bohrbugs?

Escolher uma resposta.

- a. substituição do software ✗
- b. atualização de código ✗
- c. substituição da equipe de desenvolvedores ✗
- d. retentativa, por exemplo restart ou reboot ✓
- e. reprojeto a partir da mesma especificação ✗

Correto

Notas relativas a este envio: 1/1.

Question16

Notas: 1

De acordo com Grottke e Trivedi, a maneira simples e eficiente de lidar com falhas que exibem comportamento não determinístico pode ser melhorada combinando-a com uma técnica conhecida por:

Escolher uma resposta.

○	a. isolamento de falhas ✗
○	b. armazenamento estável ✗
◉	c. checkpointing ✓
○	d. reboot ✗
○	e. detecção de falhas ✗

Correto

Notas relativas a este envio: 1/1.

Question17

Notas: 1

De acordo com Michael Grottke e Kishor S. Trivedi, falhas de software são erros humanos que se escondem no código. Assim, por exemplo, diferentes instalações do mesmo sistema operacional executando as mesmas aplicações deveriam conter as mesmas falhas.

Resposta:

◉ Verdadeiro ✓ ○ Falso ✗

Correto

Notas relativas a este envio: 1/1.

Question18

Notas: 1

Acadêmicos têm discutido que replicação não se aplica a software com as mesmas vantagens obtidas quando aplicada a hardware. Quando se executam comandos em uma segunda instalação para o mesmo software, um usuário vai encontrar as mesmas falhas que levam ao mesmo defeito encontrado na primeira instalação. Grottke e Trivedi concordam com essa argumentação.

Resposta:

Verdadeiro ✗ Falso ✓

Correto

Notas relativas a este envio: 1/1.

Question19

Notas: 1

Acadêmicos têm discutido que replicação não se aplica a software com as mesmas vantagens obtidas quando aplicada a hardware. Quando se executam comandos em uma segunda instalação para o mesmo software, um usuário vai encontrar as mesmas falhas que levam ao mesmo defeito encontrado na primeira instalação. Grottke e Trivedi afirmam que essa argumentação só é válida se todas as falhas forem Bohrbugs.

Resposta:

Verdadeiro ✓ Falso ✗

Correto

Notas relativas a este envio: 1/1.

Question20

Notas: 1

Os autores afirmam que como a maioria da falhas que não se manifestam consistentemente sob condições bem definidas são de fato Mandelbugs, então a replicação de software pode ser considerada útil.

Correto

Notas relativas a este envio: 1/1.

Question21
Notas: 1

Em que situações reiniciar um programa antes de ocorrer um defeito pode reduzir a ocorrência futura de defeitos?

Escolher uma resposta.

- a. quando a taxa de ocorrência de defeitos se mantém constante em relação ao tempo de execução ✗
- b. nunca, isso é uma piada dos autores ✗
- c. quando a taxa de ocorrência de defeitos diminui com o tempo de execução ✗
- d. quando a taxa de ocorrência de defeitos aumenta com o tempo de execução ✓

Correto

Notas relativas a este envio: 1/1.

Question22
Notas: 1

Sistemas de software rodando continuamente durante um longo período de tempo tendem a mostrar um desempenho degradado e uma taxa de defeitos crescente. Esse fenômeno é chamado de:

Escolher uma resposta.

◉	a. envelhecimento de software ✓
○	b. rejuvenescimento de software ✗
○	c. fim da vida útil do software ✗
○	d. inadequação de plataforma de hardware ✗
○	e. obsolescência planejada ✗

Correto

Notas relativas a este envio: 1/1.

Question23
Notas: 1

Sistemas de software rodando continuamente durante um longo período de tempo tendem a mostrar um desempenho degradado e uma taxa de defeitos crescente. Técnicas preventivas contra esse fenômeno são chamadas de:

Escolher uma resposta.

◉	a. rejuvenescimento de software ✓
○	b. replicação de software ✗
○	c. substituição de software ✗
○	d. obsolescência planejada ✗
○	e. envelhecimento de software ✗

Correto

Notas relativas a este envio: 1/1.

Question24
Notas: 1

Porque a taxa de ocorrência de defeitos de um programa aumenta com o tempo sem que o código sofra modificações? Grottke e Trivedi citam duas soluções possíveis a esse quebra-cabeça. Qual a **primeira** solução citada no artigo?

Escolher uma resposta.

◉	a. Bugs podem provocar erros que se acumulam com o tempo (como erros e arredondamento em operações aritméticas). ✓
○	b. O aquecimento provocado por longa operação contínua faz com que bits da memória que contém o código troquem aleatoriamente gerando alterações no programa sendo executado. ✗
○	c. A atenção dos operadores do sistema é influenciada pelo tempo total que o sistema opera continuamente fazendo com que a taxa de falhas de interação aumente sensivelmente. ✗
○	d. O tempo total que o sistema opera continuamente pode influenciar a taxa de ativação de bugs relacionados a envelhecimento. ✗

Correto

Notas relativas a este envio: 1/1.

Question25
Notas: 1

Porque a taxa de ocorrência de defeitos de um programa aumenta com o tempo sem que o código sofra modificações? Grottke e Trivedi citam duas soluções possíveis a esse quebra-cabeça, qual a **segunda** solução citada no artigo?

Escolher uma resposta.

○	a. A atenção dos operadores do sistema é influenciada pelo tempo total que o sistema opera continuamente fazendo com que a taxa de falhas de interação aumente sensivelmente. ✗
◉	b. O tempo total que o sistema opera continuamente pode influenciar a taxa de ativação de bugs relacionados a envelhecimento. ✓
○	c. O acúmulo de atualizações ao longo da vida útil de um software pode aumentar a taxa de bugs em uma dada instalação. ✗
○	d. O aquecimento provocado por longa operação contínua faz com que bits da memória que contém o código troquem aleatoriamente gerando alterações no programa sendo executado. ✗
○	e. Bugs podem provocar erros que se acumulam com o tempo (como erros e arredondamento em operações aritméticas). ✗

Correto

Notas relativas a este envio: 1/1.

Question26

Notas: 1

Bugs relacionados a envelhecimento são Mandelbugs.

Correto

Notas relativas a este envio: 1/1.

Question27

Notas: 1

Rejuvenecimento pode limpar condições de erro internas em um sistema, mas tem como consequência:

Escolher uma resposta.

○	a. necessitar de um conhecimento completo das falhas internas e suas consequências ✗
○	b. maior esforço no desenvolvimento do projeto de software ✗
◉	c. maior custo ✓
○	d. necessidade de monitoramento online ✗
○	e. necessidade de reproduzir e isolar bugs ✗

Correto

Notas relativas a este envio: 1/1.

Question28

Notas: 1

Devido ao custo associado, rejuvenescimento requer otimização do momento de atuação (*optimal timing*). Gottke e Trivedi sugerem duas abordagens para essa otimização. Associe o conceito ao termo:

usa modelos analíticos para capturar a degradação do sistema e necessidade de rejuvenescimento	abordagem baseada em modelos
monitora atributos do sistema que possam mostrar sinais de envelhecimento	abordagem baseada em medidas

Correto

Notas relativas a este envio: 1/1.

Question29

Notas: 1

Devido ao custo associado, rejuvenescimento requer otimização do momento de atuação (*optimal timing*). Gottke e Trivedi sugerem duas abordagens para essa otimização e ilustram com exemplos. Associe o exemplo ao termo:

Considerando uma dada política que determina que um servidor em espera após x horas sem rejuvenescimento deve ser rejuvenecido, os operadores podem usar o modelo para calcular o intervalo ótimo x.	abordagem baseada em modelos
O aumento da quantidade de memória utilizada pode sugerir a existência de falhas na memória que eventualmente podem conduzir a colapso do sistema.	abordagem baseada em medidas

Correto

Notas relativas a este envio: 1/1.

Question30

Notas: 1

Considere as ideias reforçadas na conclusão do artigo de Michael Grottke e Kishor S. Trivedi. Para as sentenças abaixo, responda verdadeiro ou falso de acordo com as afirmações dos autores.

Os autores afirmam que:

Reproduzir e isolar um bug relacionado a envelhecimento é tão simples e fácil como reproduzir e isolar qualquer outro Mandelbug.	Falso
Monitorar por sinais de envelhecimento de software pode ajudar a detectar outras falhas de software que escaparam das avaliações nas fases de desenvolvimento e teste.	Verdadeiro
Rejuvenecimento de software na presença de falhas de envelhecimento só é adequado se os desenvolvedores entendem as falhas e conhecem sua localizão no código.	Falso

Correto

Notas relativas a este envio: 1/1.