

INF01209 - Fundamentos de Tolerância a Falhas

Você acessou como João Luiz Grave Gross (Sair)

Moodle do INF ► FTF 2012/2 ► Questionários ► Dependabilidade: conceitos básicos ► Revisão da tentativa 1

Dependabilidade: conceitos básicos

Revisão da tentativa 1

Terminar revisão

Iniciado em	terça, 28 agosto 2012, 15:41
Completado em	quinta, 30 agosto 2012, 10:45
Tempo empregado	1 dia 19 horas
Notas	27.68/30
Nota	92.28 de um máximo de 100(92%)

- 1
Notas:
1
- Considerando os conceitos básicos apresentados no artigo do Avizienis e demais autores (A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 11– 33, 2004.), associe o conceito ao termo usado no artigo.

Composto por um conjunto de componentes interligados, onde cada componente é outro sistema.

estrutura do sistema ▼

O que é esperado que o sistema faça e é descrito na especificação funcional.

comportamento do sistema ▼

Conjunto dos seguintes estados: computação, comunicação, informação armazenada, interconexão e condição física.

estado total ▼

O que o sistema faz para executar o esperado e é descrito por uma sequência de estados.

função do sistema ▼

Parcialmente correta

Notas relativas a este envio: 0.5/1.

2 Um serviço fornecido por um sistema, no seu papel de provedor de serviço, é:

Notas:

1

- Escolher uma resposta.
- ☒ a. o comportamento do sistema conforme percebido pelo usuário ✓
 - ☐ b. uma sequência de estados internos do sistema ✗
 - ☐ c. a especificação funcional do sistema ✗
 - ☐ d. a interface do sistema percebida pelo usuário ou outro sistema ✗
 - ☐ e. uma execução correta do sistema ✗

Correto

Notas relativas a este envio: 1/1.

3 Aqui traduzimos *fault* por falha, *error* por erro e *failure* por defeito. Alguns grupos no Brasil usam outras traduções. De acordo com o artigo do Avizienis e considerando a tradução que usamos, selecione os termos adequados para os espaços vazios:

Notas:

1

Um serviço é fornecido quando o serviço implementa do sistema. de serviço é um evento que ocorre quando o serviço fornecido desvia .

Parcialmente correta

Notas relativas a este envio: 0.75/1.

4 Traduzimos *fault* por falha, *error* por erro e *failure* por defeito. Assinale verdadeiro ou falso baseando-se nos conceitos básicos do artigo do Avizienis e na tradução adotada.

Notas:

1

Um defeito de serviço é uma transição do serviço correto para o serviço incorreto.

Dizemos que um serviço apresenta defeito quando ele não segue sua especificação funcional ou quando a especificação não descreve adequadamente a função do sistema.

Dizemos que um serviço apresenta defeito apenas quando a especificação não descreve adequadamente a função do sistema.

Dizemos que um serviço apresenta defeito

apenas quando ele não segue sua especificação funcional.

Correto

Notas relativas a este envio: 1/1.

5 Um defeito de serviço significa que ao menos um estado externo do sistema

Notas: desvia do estado correto do serviço. Esse desvio é chamado de

1 . A causa real ou suposta de um erro é chamada de .

Correto

Notas relativas a este envio: 1/1.

6 De acordo com o artigo de Avizienis, o período em que o sistema apresenta

Notas: um serviço incorreto é chamado de . A

1 transição de um serviço incorreto para correto é chamada de . O desvio de um serviço correto pode assumir diferentes formas, essas formas são chamadas .

Correto

Notas relativas a este envio: 1/1.

7 Assinale verdadeiro ou falso

Notas:

1 Alguns erros nunca chegam a provocar um defeito.

Uma falha geralmente primeiro causa um erro no estado de um componente, que é parte do estado interno do sistema, e o estado externo não é imediatamente afetado.

Um erro é parte do estado total de um sistema que pode conduzir a um defeito de serviço.

Toda falha causa um defeito de serviço.

A causa real ou hipotética de um erro é um defeito.

Parcialmente correta

Notas relativas a este envio: 0.6/1.

8 Falhas podem ser internas ou externas a um sistema. Uma vulnerabilidade é

Notas: uma falha que possibilita que uma falha cause um
1 no sistema. Uma falha é quando
causa um erro; caso contrário é .

Correto

Notas relativas a este envio: 1/1.

9 No artigo, Avizienis apresenta duas definições para dependabilidade.

Notas: Assinale as duas definições apresentadas no artigo:

1

- Escolha pelo menos uma resposta.
- ☐ a. habilidade do sistema de fornecer um serviço correto ✖
 - ☒ b. habilidade de evitar defeitos de serviço que sejam mais frequentes ou mais severos que o aceitável ✔
 - ☐ c. habilidade de evitar erros no estado interno do sistema ✖
 - ☒ d. habilidade do sistema de fornecer um serviço no qual se pode justificadamente confiar ✔
 - ☐ e. habilidade de evitar falhas permanentes ✖
-

Correto

Notas relativas a este envio: 1/1.

10 Assinale o conceito mais próximo a disponibilidade:

Notas:

1

- Escolher uma resposta.
- ☐ a. Ausência de consequências catastróficas para o usuário ou ambiente. ✖
 - ☒ b. Prontidão para serviço correto. ✔
 - ☐ c. Facilidade de executar modificações e reparos. ✖
 - ☐ d. Continuidade do serviço correto. ✖
 - ☐ e. Ausência de alterações impróprias no sistema. ✖
-

Correto

Notas relativas a este envio: 1/1.

11 Assinale o conceito mais próximo ao atributo conhecido por confiabilidade:

Notas:

1

- Escolher uma resposta.
- ☒ a. Continuidade do serviço correto. ✔
 - ☐ b. Prontidão para serviço correto. ✖
 - ☐ c. Ausência de alterações impróprias no sistema. ✖
 - ☐ d. Ausência de consequências catastróficas para o usuário ou ambiente. ✖
 - ☐ e. Facilidade de executar modificações e reparos. ✖

Correto

Notas relativas a este envio: 1/1.

12 Assinale o conceito mais próximo ao atributo conhecido por segurança funcional (safety):

Notas:
1

- Escolher uma resposta.
- ☐ a. Prontidão para serviço correto. ✖
 - ☐ b. Continuidade do serviço correto. ✖
 - ☐ c. Ausência de alterações impróprias no sistema. ✖
 - ☐ d. Facilidade de executar modificações e reparos. ✖
 - ☒ e. Ausência de consequências catastróficas para o usuário ou ambiente. ✔

Correto

Notas relativas a este envio: 1/1.

13 Assinale o conceito mais próximo ao atributo conhecido por integridade:

Notas:
1

- Escolher uma resposta.
- ☒ a. Ausência de alterações impróprias no sistema. ✔
 - ☐ b. Continuidade do serviço correto. ✖
 - ☐ c. Ausência de consequências catastróficas para o usuário ou ambiente. ✖
 - ☐ d. Facilidade de executar modificações e reparos. ✖
 - ☐ e. Prontidão para serviço correto. ✖

Correto

Notas relativas a este envio: 1/1.

14 Dos atributos listados abaixo, indicar quais são atributos de dependabilidade, quais são atributos de segurança (*security*) e quais são atributos de ambos (dependabilidade e segurança):

Notas:
1

confidencialidade	<input type="text" value="segurança (security)"/>
facilidade de manutenção	<input type="text" value="dependabilidade"/>
integridade	<input type="text" value="ambos"/>
disponibilidade	<input type="text" value="ambos"/>
segurança funcional (safety)	<input type="text" value="dependabilidade"/>
confiabilidade	<input type="text" value="dependabilidade"/>

Correto

Notas relativas a este envio: 1/1.

15

Notas:
1

Vários meios foram desenvolvidos para atender os atributos de dependabilidade e segurança computacional (security). Avizienis e os demais autores do artigo classificam esses meios em quatro categorias principais. Assinale entre as opções abaixo, uma opção que não corresponda a uma das quatro categorias:

Escolher
uma
resposta.

- ☐ a. tolerância a falhas ✖
- ☐ b. prevenção de falhas ✖
- ☐ c. remoção de falhas ✖
- ☒ d. detecção de falhas ✔
- ☐ e. previsão de falhas ✖

Correto

Notas relativas a este envio: 1/1.

16

Notas:
1

Avizienis classifica os meios para alcançar os atributos de dependabilidade em quatro categorias. Associe a definição a categoria correspondente.

O objetivo é evitar defeitos de serviço na presença de falhas.

Tolerância a falhas ▼

O objetivo é estimar a quantidade atual de falhas, a incidência futura e a suas possíveis consequências.

Previsão de falhas ▼

O objetivo é prevenir a ocorrência ou introdução de falhas.

Prevenção de falhas ▼

O objetivo é reduzir o número e a severidade das falhas.

Remoção de falhas ▼

Correto

Notas relativas a este envio: 1/1.

17

Notas:
1

O item 3.1 do artigo do Avizienis inicia falando do ciclo de vida de um sistema. Os autores distinguem duas fases. Associe a definição ou característica ao nome da fase:

Consiste dos seguintes elementos: o mundo físico, os desenvolvedores, as ferramentas de desenvolvimento, os recursos para produção e teste.

fase de desenvolvimento ▼

Inclui todas as atividades desde a concepção inicial do sistema até o sistema estar testado e pronto para prover o serviço desejado.

fase de desenvolvimento ▼

Inicia quando o sistema é
aceito e começa a fornecer o
serviço ao usuário.

fase de uso

Alterna períodos de
oferecimento do serviço
correto, interrupção de
serviço (outage) e
desligamento (shutdown).

fase de uso

Correto

Notas relativas a este envio: 1/1.

18

Notas:
1

Um sistema alterna períodos de fornecimento correto de serviço, de interrupção de serviço (*outage*) e de desligamento (*shutdown*). Considerando a diferença entre interrupção de serviço e desligamento, associe cada evento a interrupção de serviço ou a desligamento.

Causado por defeito de
serviço.

interrupção de serviço (outage)

Serviço incorreto ou mesmo
nenhum serviço é fornecido.

interrupção de serviço (outage)

Parada intencional e
autorizada do serviço.

desligamento (shutdown)

Correto

Notas relativas a este envio: 1/1.

19

Notas:
1

Qual o significado do termo "manutenção" no artigo?

Escolher
uma
resposta.

- ☐ a. apenas reparos na fase de uso do sistema ✗
- ☐ b. apenas modificações na fase de desenvolvimento ✗
- ☒ c. reparos e modificações na fase de uso do sistema ✓
- ☐ d. troca de componentes de hardware ✗
- ☐ e. reparos e modificações em qualquer fase do sistema ✗

Correto

Notas relativas a este envio: 1/1.

20

Notas:
1

Considerando os conceitos apresentados no artigo de Avizienis, responda verdadeiro ou falso.

Reparo é parte da remoção de falhas durante a
fase de uso.

Verdadeiro

Reparo e tolerância a falha são conceitos
relacionados.

Verdadeiro

Uma distinção entre manutenção e tolerância a falha é que tolerância a falhas envolve a participação de um agente externo, como por exemplo, equipamento de teste e reinstalação de software.

falso

Correto

Notas relativas a este envio: 1/1.

21 Responda com números decimais inteiros.

Notas:

1 No item em que Avizienis e os demais autores tratam da taxonomia de falhas (3.2.1), os autores classificam as falhas de acordo com 8 pontos de vista, também chamados de classes de falhas elementares. Se todas as classes pudessem ser combinadas, teríamos 256 diferentes classes de falhas combinadas. Entretanto, os autores identificaram apenas 31 combinações que fazem sentido, por enquanto.

Correto

Notas relativas a este envio: 1/1.

22 Assinale SIM para as classes de falha elementares e NÃO para os termos que não correspondem as classes de falhas elementares sugeridas por Avizienis e demais autores.

1

persistência

SIM

deterioração física

NÃO

limites

SIM

capacidade

SIM

interferência física

NÃO

objetivo

SIM

causa

SIM

errata

NÃO

fase

NÃO

intenção

NÃO

dimensão

SIM

permanência

NÃO

Parcialmente correta

Notas relativas a este envio: 0.83/1.

23 Todas as combinações razoáveis de classes de falhas foram agrupadas em 3 grupos principais (com intersecção parcial). Quais são esses grupos?

Notas:

1

Escolher
uma
resposta.

- ☐ a. falhas de especificação, de desenvolvimento e de operação ✖
- ☐ b. falhas físicas, humanas e ambientais ✖
- ☐ c. falhas internas, externas e correlacionadas ✖
- ☒ d. falhas de desenvolvimento, físicas e de interação ✔
- ☐ e. falhas simples, múltiplas simultâneas e múltiplas em cascata ✖

Correto

Notas relativas a este envio: 1/1.

24 Falhas internas e externas são agrupadas na classe elementar de falhas denominada:

Notas:

1

Escolher
uma
resposta.

- ☐ a. dimensão ✖
- ☐ b. intenção ✖
- ☐ c. persistência ✖
- ☐ d. causa ✖
- ☒ e. limites ✔

Correto

Notas relativas a este envio: 1/1.

25 Falhas permanentes e transientes são agrupadas na classe elementar de falhas denominada:

Notas:

1

Escolher
uma
resposta.

- ☒ a. persistência ✔
- ☐ b. causa ✖
- ☐ c. dimensão ✖
- ☐ d. fase ✖
- ☐ e. limites ✖

Correto

Notas relativas a este envio: 1/1.

26 Falhas de hardware e de software são agrupadas na classe elementar de falhas denominada:

Notas:

1

Escolher
uma
resposta.

- ☐ a. persistência ✖
- ☐ b. fase ✖

- ☒ c. dimensão ✓
- ☐ d. causa ✗
- ☐ e. limites ✗

Correto

Notas relativas a este envio: 1/1.

27

Notas:
1

As falhas humanas, ou falhas resultantes da ação de pessoas, são distinguíveis pelo objetivo. Considerando o **objetivo** da interação humana com o sistema, quais são as duas classes básicas de falhas humanas?

Escolher
uma
resposta.

- ☐ a. permanente e transiente ✗
- ☒ b. maliciosa e não-maliciosa ✓
- ☐ c. deliberada e não-deliberada ✗
- ☐ d. acidental e incompetência ✗
- ☐ e. desenvolvimento e operacional ✗

Correto

Notas relativas a este envio: 1/1.

28

Notas:
1

Falhas de desenvolvimento não maliciosas podem ocorrer em hardware ou em software. Em hardware, principalmente em microprocessadores, essas falhas, quando descobertas depois que a produção começou, recebem um nome especial. De acordo com Avizienis, como se chamam?

Escolher
uma
resposta.

- ☐ a. desvio da especificação ✗
- ☐ b. revisão ✗
- ☐ c. errata ✓
- ☒ d. incompetência ✗
- ☐ e. update da especificação ✗

Errado

Notas relativas a este envio: 0/1.

29

Notas:
1

As classes combinadas de falhas aparecem agrupadas em grupos parcialmente sobrepostos. Um desses grupos compreende as falhas de interação. Falhas de interação ocorrem durante a fase de , portanto elas são todas do tipo . Elas são causadas por elementos do ambiente do sistema em uso e são, então, todas falhas . Um exemplo de falhas de interação que consiste em erro na inicialização de parâmetros do sistema são as falhas de .

Correto

Notas relativas a este envio: 1/1.

30

Notas:

1

Uma característica comum às falhas de interação é que, para terem sucesso, elas necessitam da presença prévia de:

Escolher

uma

resposta.

- ☒ a. uma vulnerabilidade ✓
- ☐ b. uma falha externa ✗
- ☐ c. uma falha humana ✗
- ☐ d. uma falha maliciosa ✗
- ☐ e. uma tentativa de intrusão ✗

Correto

Notas relativas a este envio: 1/1.

Terminar revisão

Você acessou como [João Luiz Grave Gross \(Sair\)](#)

FTF 2012/2

Dependabilidade: defeitos e erros

Revisão da tentativa 1

Iniciado em	terça, 4 setembro 2012, 18:49
Completado em	domingo, 9 setembro 2012, 21:17
Tempo empregado	5 dias 2 horas
Notas	30.3/31
Nota	97.74 de um máximo de 100(98%)

Question1

Notas: 1

Traduzimos *fault* por falha e *failure* por defeito. Considere a situação em que um sistema foi projetado de acordo com sua especificação e se comporta rigorosamente de acordo com essa especificação. Mesmo assim acontece um desastre inaceitável do ponto de vista do usuário. De acordo com o item 3.3 do artigo do Avizienis, assinale **verdadeiro** ou **falso** para as sentenças que seguem:

O sistema apresenta defeito porque não está fornecendo o serviço esperado mesmo seguindo a sua especificação.

Verdadeiro



A especificação tem uma ou mais falhas.

Verdadeiro



Um defeito devido a uma falha de especificação é um conceito muito vago e subjetivo e por isso não se deve levar muito a sério uma especificação.

Falso



O sistema não apresenta defeito porque está de acordo com a especificação.

Falso



O sistema apresenta defeito e a especificação não está adequadamente descrevendo a função do sistema.

Verdadeiro



Correto

Notas relativas a este envio: 1/1.

Question2

Notas: 1

De acordo com Avizienis, um defeito de serviço é um evento que ocorre quando o serviço fornecido desvia do serviço correto. As diferentes formas nas quais um sistema se desvia de um serviço correto são conhecidas como:

Escolher uma resposta.



a. severidades de defeito de serviço ✗



b. falhas de especificação ✗



c. modos de defeito de serviço ✓



d. inconsistências do serviço ✗



e. erros de serviço ✗

Correto

Notas relativas a este envio: 1/1.

Question3

Notas: 1

De acordo com Avizienis e demais autores, falhas de especificação podem ser de dois tipos. Assinale os dois tipos na lista a seguir:

Escolha pelo menos uma resposta.

- | | |
|-------------------------------------|---------------------------|
| <input type="checkbox"/> | a. inconsequência ✗ |
| <input checked="" type="checkbox"/> | b. omissão ✓ |
| <input type="checkbox"/> | c. inadequação ✗ |
| <input checked="" type="checkbox"/> | d. comissão (commision) ✓ |
| <input type="checkbox"/> | e. incorreção ✗ |

Correto

Notas relativas a este envio: 1/1.

Question4

Notas: 1

Considerando falhas de especificação, responda verdadeiro ou falso para as afirmações a seguir.

- | | |
|---|--------------|
| Falsa interpretação, suposições infundadas, inconsistências, erros tipográficos são exemplos de falhas de comissão. | verdadeiro ▼ |
| Um defeito pode ser subjetivo e questionável. | verdadeiro ▼ |
| Um defeito é facilmente identificável e caracterizável. | falso ▼ |

Correto

Notas relativas a este envio: 1/1.

Question5

Notas: 1

Modos de defeito de serviço caracterizam serviço incorreto de acordo com 4 pontos de vista. Assinale o ponto de vista que **não** pertence aos 4 listados no artigo de Avizienis.

Escolher uma resposta.



a. detectabilidade do defeito ✗



b. consequência do defeito ✗



c. domínio do defeito ✗



d. consistência do defeito ✗



e. custo do defeito ✓

Correto

Notas relativas a este envio: 1/1.

Question6

Notas: 1

Considerando o ponto de vista do domínio do defeito, quais são os dois tipos de defeitos citados por Avizienis?

Escolher uma resposta.



a. silencioso e severo ✗

<input checked="" type="radio"/>	b. conteúdo e temporização ✓
<input type="radio"/>	c. parada e errático ✗
<input type="radio"/>	d. consistentes e bizantinos ✗
<input type="radio"/>	e. sinalizado e não sinalizado ✗

Correto

Notas relativas a este envio: 1/1.

Question7

Notas: 1

Considerando o domínio do defeito, quais são as duas classes de defeitos quando tanto a informação quanto a temporização são incorretas?

Escolher uma resposta.

<input checked="" type="radio"/>	a. defeitos de parada e errático ✓
<input type="radio"/>	b. adiantamento ou atraso ✗
<input type="radio"/>	c. defeitos silenciosos e severos ✗
<input type="radio"/>	d. defeitos sinalizados e não sinalizados ✗
<input type="radio"/>	e. defeitos consistentes e bizantinos ✗

Correto

Notas relativas a este envio: 1/1.

Question8

Notas: 1

Considerando o domínio de defeitos, associe a descrição a classe de defeito:

caso especial onde nenhum serviço é fornecido na interface do sistema, ou no caso de sistemas distribuídos, nenhuma mensagem é enviada

defeito silencioso ▼

o estado externo fica imutável, ou seja, a atividade do sistema não é mais percebida pelo usuário

defeito de parada ▼

o serviço é fornecido mas é incoerente

defeito errático ▼

o serviço é fornecido muito cedo ou muito tarde

defeito de temporização ▼

Correto

Notas relativas a este envio: 1/1.

Question9

Notas: 1

Quais são os dois tipos de defeitos, considerando o ponto de vista da detectabilidade do defeito?

Escolher uma resposta.



a. sinalizado e não sinalizado ✓



b. conteúdo e temporização ✗



c. consistentes e bizantino ✗



d. silencioso e severo ✗



e. parada e errático ✗

Correto

Notas relativas a este envio: 1/1.

Question10

Notas: 1

Considerando o ponto de vista da detectabilidade do defeito, os mecanismos de detecção apresentam dois modos de defeito. Associe as descrições aos termos apropriados:

Sinalização de perda de função quando nenhum defeito efetivamente ocorreu.

Falso alarme

Nenuma indicação de perda de função quando o defeito ocorre.

Defeito não sinalizado

Correto

Notas relativas a este envio: 1/1.

Question11

Notas: 1

Quais são os dois tipos de defeitos, considerando a consistência do defeito?

Escolher uma resposta.



a. parada e errático ✗



b. sinalizado e não sinalizado ✗



c. consistentes e bizantinos ✓



d. conteúdo e temporização ✗



e. silencioso e severo ✗

Correto

Notas relativas a este envio: 1/1.

Question12

Notas: 1

Alguns ou todos os usuários percebem o serviço incorreto de maneira diferente, alguns até podem perceber o serviço como correto. Esse tipo de defeito é chamado de:

Escolher uma resposta.



a. errático ✗



b. detectável ✗



c. consistente ✗



d. bizantino ✓



e. silencioso ✗

Correto

Notas relativas a este envio: 1/1.

Question13

Notas: 1

Graduar de defeitos sobre o ambiente de um sistema permite definir do defeito. Aos seus estão geralmente associadas probabilidades máximas aceitáveis de ocorrência. A quantidade, o nome e a definição dos níveis de

severidade de defeitos, assim como os níveis aceitáveis de probabilidades de ocorrência, são relacionados e envolvem os atributos de dependabilidade e segurança (security) para a aplicação considerada.

Correto

Notas relativas a este envio: 1/1.

Question14

Notas: 1

Considere os exemplos de critérios para determinação das classes de severidade de defeitos mencionados no artigo do Avizienis. Associe o critério ao atributo:

duração da interrupção do serviço

possibilidade de perdas de vidas humanas

extensão da corrupção de dados e a capacidade de recuperar essa corrupção

Correto

Notas relativas a este envio: 1/1.

Question15

Notas: 1

Sistemas que são projetados e implementados de tal forma que apresentam defeitos apenas em modos específicos descritos na especificação de dependabilidade e segurança (*security*) e apenas em uma extensão aceitável são sistemas com controle de defeito (*fail-controlled systems*). Seguindo a nomenclatura apresentada no artigo de Avizienis, associe a descrição ao termo apropriado:

sistema onde os defeitos são todos (ou numa extensão aceitável) apenas defeitos silenciosos

sistema onde os defeitos são todos (ou numa extensão aceitável) apenas defeitos menores (minor failures)

fail-safe

sistema onde os defeitos são todos (ou numa extensão aceitável) apenas defeitos da classe de parada

fail-halt ou fail-stop

Correto

Notas relativas a este envio: 1/1.

Question16

Notas: 1

Falhas de desenvolvimento podem levar a defeitos parciais ou completos de desenvolvimento ou permanecer indetectáveis até a fase de uso do sistema. De acordo com Avizienis (item 3.3.2) existem dois aspectos relacionados a defeitos de desenvolvimento, são eles:

Escolher uma resposta.



a. especificação com erro e alterações na especificação ✗



b. tecnologia obsoleta e defeitos de desempenho ✗



c. erros humanos e defeitos de produção ✗



d. defeitos de orçamento e defeitos de escalonamento ✓



e. defeitos de ferramentas e interferências do ambiente ✗

Correto

Notas relativas a este envio: 1/1.

Question17

Notas: 1

Avizienis e os demais autores citam várias causas para defeitos de desenvolvimento (item 3.3.2). Segundo eles, todas as causas citadas são geralmente devidas a:

Escolher uma resposta.

- ☐ a. falta de gente para compor uma equipe de desenvolvedores ✗
- ☐ b. funcionalidade do sistema desconhecida a priori ✗
- ☒ c. complexidade do sistema a ser desenvolvido subestimada ✓
- ☐ d. ausência de especificação para o sistema ✗
- ☐ e. carência de formação acadêmica em tolerância a falhas ✗

Correto

Notas relativas a este envio: 1/1.

Question18

Notas: 1

Assinale **verdadeiro** ou **falso** considerando o item 3.3.3 do artigo do Avizienis e demais autores.

Uma classe de falha de especificação de dependabilidade é a escolha injustificada de exagerados níveis de demanda para um ou mais atributos de dependabilidade, o que eleva os custos e pode levar a defeito de desenvolvimento.

Falso

Não é necessária uma especificação de dependabilidade, os desenvolvedores devem garantir por contrato que falhas não ocorram.

Falso

Uma especificação de dependabilidade e segurança (security) é um contrato que estabelece os objetivos de cada atributo: confiabilidade, disponibilidade, safety, confidencialidade, integridade e facilidade de manutenção.

Verdadeiro

Um defeito de dependabilidade ou segurança ocorre quando o sistema sofre defeitos de serviço mais frequentes ou mais severos do que aceitável.

Verdadeiro



Uma especificação de dependabilidade e segurança jamais contém falhas.

Falso



Parcialmente correta

Notas relativas a este envio: 0.8/1.

Question19

Notas: 2

Localize-se no item 3.4 do artigo do Avizienis. Um erro é definido como parte do estado global do sistema que pode conduzir a , que ocorre quando o erro causa o desvio do serviço correto. A causa de um erro recebe o nome de .

Correto

Notas relativas a este envio: 2/2.

Question20

Notas: 1

Escolha o termo apropriado:

quando a presença do erro é indicada por uma mensagem de erro ou por um sinal de erro dizemos que o erro é

detectado



quando o erro está presente mas não é sinalizado dizemos que o erro é

latente



Correto

Notas relativas a este envio: 1/1.

Question21

Notas: 1

Assinale verdadeiro ou falso de acordo com as opiniões expressas por Avizienis e demais autores.

Um erro sempre provoca um defeito de serviço.

Falso

A estrutura de um sistema e o comportamento do sistema definem se um erro vai conduzir a um defeito de serviço ou não.

Verdadeiro

Um defeito de serviço não ocorre enquanto o estado externo de um componente afetado por um erro não se torna parte do estado externo do sistema.

Verdadeiro

Sempre que um erro se torna parte do estado externo de um componente, ocorre um defeito de serviço do componente, mas não necessariamente do sistema que contém o componente.

Verdadeiro

Correto

Notas relativas a este envio: 1/1.

Question22

Notas: 1

A natureza da redundância que existe na estrutura de um sistema pode ser de dois tipos, segundo Avizienis:

Escolher uma resposta.



a. explícita ou inesperada ✗



b. passiva ou ativa ✗



c. homogênia ou diversitária ✗



d. de software ou de hardware ✗



e. protetora ou não intencional ✓

Correto

Notas relativas a este envio: 1/1.

Question23

Notas: 1

Considerando possíveis classificações de erros citadas por Avizienis, assinale verdadeiro se a classificação aparece no artigo, falso caso contrário.

Erros podem ser classificados, considerando persistência, como erros permanentes ou erros transientes.

Verdadeiro



Erros podem ser classificados, em função dos danos financeiros provocados, como erros maliciosos ou não maliciosos.

Falso



Erros podem ser classificados em termos dos defeitos de serviço elementares que eles provocam.

Verdadeiro



Erros de conteúdo podem ser classificados, na área de códigos de controle de erros, em termos do padrão de bits afetados.

Falso



Parcialmente correta

Notas relativas a este envio: 0.5/1.

Question24

Notas: 1

Considerando uma possível categorização de erros sugerida por Avizienis, associe o termo ao conceito:

afetam apenas um componente

erro simples



são provocados por falhas que causam problemas simultâneos em mais do que um componente

erros múltiplos relacionados

Correto

Notas relativas a este envio: 1/1.

Question25

Notas: 1

Localize-se no item 3.5 do artigo do Avizienis sob análise. Relacione o termo ao conceito.

falha que produz um erro

falha ativa

falha que não produziu um erro (ainda)

falha dormente

Correto

Notas relativas a este envio: 1/1.

Question26

Notas: 1

Relacione o termo ao conceito.

aplicação de uma entrada a um componente que faz uma falha dormente se tornar ativa

ativação de falha

processo computacional que faz com que um erro seja sucessivamente transformado em outros erros

propagação de erro

Correto

Notas relativas a este envio: 1/1.

Question27

Notas: 1

A habilidade de identificar o padrão de ativação de uma falha que causa um ou mais erros, de acordo com Avizienis, é chamada de:

Escolher uma resposta.

- ☐ a. estímulo de ativação da falha ✗
- ☒ b. reprodutibilidade da ativação da falha ✓
- ☐ c. probabilidade de ativar a falha ✗
- ☐ d. identificação do padrão de ativação ✗
- ☐ e. inevitabilidade de ativação de falha ✗

Correto

Notas relativas a este envio: 1/1.

Question28

Notas: 1

Falhas podem ser categorizadas de acordo com sua reprodutibilidade de ativação. Falhas cuja ativação é reproduzível são chamadas de:

Escolher uma resposta.

- ☐ a. intermitentes ✗
- ☒ b. sólidas ou hard ✓
- ☐ c. múltiplas ✗



d. evasivas ou soft ✗



e. simples ✗

Correto

Notas relativas a este envio: 1/1.

Question29

Notas: 1

Falhas podem ser categorizadas de acordo com sua reprodutibilidade de ativação. Falhas cuja ativação não é sistematicamente reproduzível são chamadas de:

Escolher uma resposta.



a. evasivas ou soft ✓



b. residuais ✗



c. independentes ✗



d. intermitentes ✗



e. sólidas ou hard ✗

Correto

Notas relativas a este envio: 1/1.

Question30

Notas: 1

Devido a similaridade de manifestação, dois tipos de falhas foram agrupadas com o nome de falhas intermitentes. São elas:

Escolher uma resposta.

- | | | |
|----------------------------------|--|---|
| <input type="radio"/> | a. falhas de software e falhas de desenvolvimento de hardware ✗ | |
| <input checked="" type="radio"/> | b. falhas evasivas de desenvolvimento e falhas transientes de hardware ✓ | falhas permanentes e falhas transientes |
| <input type="radio"/> | c. falhas soft e falhas hard ✗ | |
| <input type="radio"/> | d. falhas naturais e falhas humanas ✗ | |
| <input type="radio"/> | e. falhas permanentes e falhas transientes ✗ | |

Correto

Notas relativas a este envio: 1/1.

Dependabilidade: meios

Revisão da tentativa 1

Iniciado em	segunda, 10 setembro 2012, 19:01
Completado em	segunda, 10 setembro 2012, 19:42
Tempo empregado	41 minutos 1 segundo
Notas	29.6/30
Nota	98.67 de um máximo de 100(99%)

Question1

Notas: 1

Considerando o item 5 do artigo "Basic Concepts and Taxonomy of Dependable and Secure Computing", de Avizienis e demais autores, assinale **verdadeiro** para os termos que correspondem aos "meios" que foram mencionados no artigo para alcançar dependabilidade e **falso** caso contrário.

remoção de falhas

Verdadeiro



confidencialidade

Falso



disponibilidade	Falso
prevenção de falhas	Verdadeiro
previsão de falhas	Verdadeiro
redundância	Falso
confiabilidade	Falso
tolerância a falhas	Verdadeiro

Correto

Notas relativas a este envio: 1/1.

Question2

Notas: 1

Melhorar o processo de desenvolvimento para reduzir o número de falhas introduzidas no sistema durante sua produção é um exemplo de:

Escolher uma resposta.

<input type="radio"/>	Perda de tempo porque falhas são inevitáveis ✗
<input type="radio"/>	Previsão de falhas ✗
<input checked="" type="radio"/>	Prevenção de falhas ✓



Tolerância a falhas ✗



Remoção de falhas ✗

Correto

Notas relativas a este envio: 1/1.

Question3

Notas: 1

Tolerância a falhas visa principalmente:

Escolher uma resposta.



aprender a conviver com defeitos ✗



evitar falhas ✗



evitar defeitos ✓



prevenir que falhas ocorram ✗



remover falhas ✗

Correto

Notas relativas a este envio: 1/1.

Question4

Notas: 1

De acordo com Avizienis, tolerância a falhas emprega duas estratégias básicas. São elas:

Escolher uma resposta.



redundância de hardware e isolamento de falhas ✗



compensação de erros e reinicialização ✗



detecção de erro e recuperação do sistema ✓



diagnóstico de falhas e reconfiguração do sistema ✗



metodologias para o desenvolvimento de software e regras para o projeto de hardware ✗

Correto

Notas relativas a este envio: 1/1.

Question5

Notas: 1

Considerando as técnicas de tolerância a falhas, associe o conceito à técnica apropriada.

Previne nova ativação de falhas.

Tratamento de falhas ▼

Identifica a presença de um erro.

Detecção de erro ▼

Transforma um estado que contém um ou mais erros e falhas em um estado sem os erros detectados e sem falhas que possam ser novamente ativadas.

Recuperação ▼

Elimina erros do estado do sistema.

Tratamento de erros ▼

Correto

Notas relativas a este envio: 1/1.

Question6

Notas: 1

Assinale as técnicas para tratamento de erros:

Escolha pelo menos uma resposta.

<input checked="" type="checkbox"/>	recuperação por retorno ✓
<input type="checkbox"/>	diagnóstico ✗
<input type="checkbox"/>	reconfiguração ✗
<input checked="" type="checkbox"/>	recuperação por avanço ✓
<input type="checkbox"/>	reinicialização ✗
<input checked="" type="checkbox"/>	compensação ✓
<input type="checkbox"/>	isolamento ✗

Correto

Notas relativas a este envio: 1/1.

Question7

Notas: 1

Assinale as técnicas para tratamento de falhas:

Escolha pelo menos uma resposta.

<input type="checkbox"/>	recuperação por retorno ✗
--------------------------	---------------------------

<input checked="" type="checkbox"/>	diagnóstico ✓
<input checked="" type="checkbox"/>	isolamento ✓
<input checked="" type="checkbox"/>	reconfiguração ✓
<input type="checkbox"/>	compensação ✗
<input type="checkbox"/>	recuperação por avanço ✗
<input checked="" type="checkbox"/>	reinicialização ✓

Correto

Notas relativas a este envio: 1/1.

Question8

Notas: 1

Associe o conceito à técnica de tratamento de erros adequada.

Leva o sistema a um estado salvo que ocorreu antes da ocorrência do erro:

recuperação por retorno ▼

O novo estado é um estado sem erros detectados:

recuperação por avanço ▼

O estado com erros contém redundância suficiente para permitir que erros sejam mascarados:

compensação ▼

Correto

Notas relativas a este envio: 1/1.

Question9

Notas: 1

Considere a detecção de erros. Associe o conceito ao termo apropriado.

Ocorre durante a prestação normal de serviço.

detecção concorrente

Ocorre quando o serviço é suspenso e se presta a verificar a presença de erros latentes e falhas dormentes.

detecção preemptiva

Correto

Notas relativas a este envio: 1/1.

Question10

Notas: 1

Qual dos conceitos abaixo se aplica a diagnóstico?

Escolher uma resposta.



Identificação e registro da causa do erro em termos de localização e tipo. ✓



Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. ✗



Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✗



Exclusão física ou lógica do componente com falha, ou seja, transformação da falha ativa em dormente. ✗

Correto

Notas relativas a este envio: 1/1.

Question11

Notas: 1

Qual dos conceitos abaixo se aplica a isolamento de falha?

Escolher uma resposta.

- ☐ Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✗
- ☐ Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. ✗
- ☐ Identificação e registro da causa do erro em termos de localização e tipo. ✗
- ☒ Exclusão física ou lógica do componente com falha, ou seja, transformação da falha em dormente. ✓

Correto

Notas relativas a este envio: 1/1.

Question12

Notas: 1

Qual dos conceitos abaixo se aplica a reconfiguração?

Escolher uma resposta.

- ☐ Exclusão física ou lógica do componente com falha, ou seja, transformação da falha em dormente. ✗
- ☐ Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✗
- ☒ Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. ✓



Identificação e registro da causa do erro em termos de localização e tipo. ✗

Correto

Notas relativas a este envio: 1/1.

Question13

Notas: 1

Qual dos conceitos abaixo se aplica a reinicialização?

Escolher uma resposta.



Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. ✗



Identificação e registro da causa do erro em termos de localização e tipo ✗



Exclusão física ou lógica do componente com falha, ou seja, transformação da falha em dormente. ✗



Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✓

Correto

Notas relativas a este envio: 1/1.

Question14

Notas: 1

Considere as técnicas de tratamento de erros. Todas elas podem ser aplicadas por demanda, ou seja, quando ocorre um erro, mas apenas uma pode ser aplicada sistematicamente, independente da presença ou não do erro. Assinale a técnica que pode ser aplicada sistematicamente.

Escolher uma resposta.



a. compensação ✓



b. recuperação por retorno ✗



c. nenhuma das técnicas de tratamento de erro pode ser aplicada sistematicamente ✗



d. recuperação por avanço ✗

Correto

Notas relativas a este envio: 1/1.

Question15

Notas: 1

Avizienis não lista mascaramento de falhas como uma técnica de tolerância a falhas. No texto, mascaramento de falhas é definido como:

Escolher uma resposta.



a. resultado do uso sistemático de compensação podendo levar progressivamente ou eventualmente à perda da redundância de proteção ✓



b. resultado do uso sistemático de recuperação por retorno podendo levar progressivamente ao estado inicial do sistema (estado no momento do power up) ✗



c. resultado do uso sistematico de tratamento de falhas seguida imediatamente por recuperação por avanço e postergação da manutenção ✗



d. resultado do uso de recuperação por retorno seguida imediatamente por recuperação por avanço e um passo posterior de tratamento de erro ✗

Correto

Notas relativas a este envio: 1/1.

Question16

Notas: 1

Assinale **verdadeiro** ou **falso** de acordo com os conceitos do artigo do Avizienis.

Após a detecção de um erro, sempre deve seguir-se o tratamento do erro e só então, depois disso, o tratamento da falha.

Falso



As técnicas de detecção de erros preemptiva e tratamento de erros, possivelmente acompanhadas de tratamento de falha, são comumente executadas no power up do sistema.

Verdadeiro



Um fator que distingue tolerância a falhas de manutenção é que manutenção é realizada por um agente externo.

Verdadeiro



Recuperação por avanço (rollforward) e recuperação por retorno (rollback) são mutuamente exclusivas, ou seja, só pode ser usada uma ou outra.

Falso



Falhas intermitentes não necessitam isolamento ou reconfiguração.

Verdadeiro



Correto

Notas relativas a este envio: 1/1.

Question17

Notas: 1

Assinale **verdadeiro** ou **falso** de acordo com o artigo do Avizienis (item 5.2.2).

Para falhas sólidas de projeto, canais redundantes podem apresentar projeto idêntico pois componentes de hardware sempre apresentam defeitos de forma independente uns dos outros.

Falso



As classes de falha que podem realmente ser toleradas em um sistema dependem da hipótese de falha que foi

Verdadeiro



considerada no processo de desenvolvimento do sistema.

A escolha das técnicas de tolerância a falhas a serem empregadas depende do custo do projeto e dos recursos disponíveis e não da hipótese de falha considerada.

Falso

Para falhas físicas, canais redundantes podem apresentar projeto idêntico pois podemos nos basear na hipótese que componentes de hardware apresentam defeitos independentemente uns dos outros.

Verdadeiro

Um método popular de alcançar tolerância a falhas é realizar múltiplas computações através de múltiplos canais, seja sequencialmente ou concorrentemente.

Verdadeiro

Correto

Notas relativas a este envio: 1/1.

Question18

Notas: 1

A existência de recursos internos ao componente, seja de hardware ou software, para que o componente, além de executar sua função, também possa realizar detecção concorrente de erros está relacionada ao conceito de:

Escolher uma resposta.



área de confinamento de erro ✗



cobertura ✗



recuperação por retorno (rollback) ✗



projeto diversitário ✗



componente auto-verificável (self-checking) ✓

Correto

Notas relativas a este envio: 1/1.

Question19

Notas: 1

Nem todas as técnicas de tolerância a falhas são igualmente efetivas. A medida de eficiência de uma dada técnica de tolerância a falhas é chamada de (escreva o nome em português):

Resposta:

Cobertura

Correto

Notas relativas a este envio: 1/1.

Question20

Notas: 1

As imperfeições de tolerância a falhas, segundo Avizienis e demais autores, são devidas a dois fatores. São eles:

Escolher uma resposta.



problemas com confinamento de falhas e projeto diversitário mal especificado ✗



ausência de projeto diversitário e de componentes auto-verificáveis ✗



deficiência na cobertura do tratamento de erros e falhas e deficiências nas hipóteses sobre falhas ✓



presença de falhas dependentes e ausência de isolamento de falhas ✗

Correto

Notas relativas a este envio: 1/1.

Question21

Notas: 1

Quando trabalhamos com múltiplos componentes, uma questão importante é prevenir a propagação de erros para os componentes livres de falhas, principalmente quando um componente precisa comunicar alguma informação para outros componentes. Todos os componentes que recebem a informação devem alcançar um estado consistente entre si. O problema de alcançar um acordo entre todos os componentes sobre um valor é conhecido como:

Escolher uma resposta.

- | | |
|----------------------------------|---|
| <input checked="" type="radio"/> | a. problema do consenso ✓ |
| <input type="radio"/> | b. problema do acordo recursivo ✗ |
| <input type="radio"/> | c. problema da exclusão mútua ✗ |
| <input type="radio"/> | d. problema da consistência mútua ✗ |
| <input type="radio"/> | e. problema da informação de origem única ✗ |

Correto

Notas relativas a este envio: 1/1.

Question22

Notas: 1

Avizienis e os demais autores do artigo sob análise afirmam que tolerância a falhas é um conceito recursivo. Qual o significado desta afirmação?

Escolher uma resposta.

- | | |
|-----------------------|---|
| <input type="radio"/> | a. Conceito recursivo significa neste contexto que os mecanismos de tolerância a falhas não podem ser implementados nem em software nem em hardware pois sempre resta um componente |
|-----------------------|---|

não protegido. ✖



b. Conceito recursivo significa neste contexto que os mecanismos que implementam tolerância a falhas são extremamente complexos e só podem ser implementados em software por linguagens de programação que permitem recursão. ✖



c. Conceito recursivo significa neste contexto que os mecanismos que implementam tolerância a falhas também devem ser protegidos contra as falhas que possam afetá-los. ✔



d. Conceito recursivo significa neste contexto que os mecanismos de tolerância a falhas não podem ser implementados em hardware, pois componentes de hardware não são recursivos. ✖



e. Conceito recursivo significa neste contexto que os mecanismos que implementam tolerância a falhas são auto-suficientes, auto-contidos e não sofrem interferência de falhas. ✖

Correto

Notas relativas a este envio: 1/1.

Question23

Notas: 1

Segundo Avizienis e demais autores, os termos auto-reparo, auto-cura e resiliência são:

Escolher uma resposta.



relativos a técnicas de tratamento de defeitos ✖



sinônimos de tolerância a falhas ✔



sinônimos de dependabilidade ✖



referentes a técnicas complementares a tolerância a falhas ✖



relativos a técnicas específicas de tratamento de erros ✖

Correto

Notas relativas a este envio: 1/1.

Question24

Notas: 1

Um dos quatro meios para atingir dependabilidade é a remoção de falhas (5.3). Segundo Avizienis, a remoção de falhas durante a fase de desenvolvimento de um sistema consiste em 3 passos. São eles:

Escolher uma resposta.



detecção de erro, tratamento de erro e tratamento de falha ✖



verificação estática, verificação dinâmica e execução simbólica ✖



rollback, rollforward e compensação ✖



teste dinâmico, teste estático e injeção de falhas ✖



verificação, diagnóstico e correção ✔

Correto

Notas relativas a este envio: 1/1.

Question25

Notas: 1

As técnicas de verificação podem ser classificadas levando em consideração se executam ou não o sistema. De acordo com essa classificação, a verificação é chamada de:

Escolher uma resposta.

- ☐ a. determinística ou randômica ✗
- ☐ b. formal ou estatística ✗
- ☒ c. dinâmica ou estática ✓
- ☐ d. regressiva ou não regressiva ✗
- ☐ e. numérica ou simbólica ✗

Correto

Notas relativas a este envio: 1/1.

Question26

Notas: 1

Remoção de falhas durante a fase de uso (ou operação) de um sistema compreende duas técnicas. São elas:

Escolher uma resposta.

- ☒ manutenção corretiva e manutenção preventiva ✓
- ☐ verificação e correção ✗
- ☐ recuperação e compensação ✗
- ☐ isolamento e remoção ✗
- ☐ detecção e tratamento de erro ✗

Correto

Notas relativas a este envio: 1/1.

Question27

Notas: 1

Realizar uma avaliação do comportamento de um sistema com respeito a ocorrência de falhas e sua ativação é uma tarefa relacionada a:

Escolher uma resposta.

<input type="radio"/>	tolerância a falhas ✗
<input type="radio"/>	remoção de falhas ✗
<input checked="" type="radio"/>	previsão de falhas ✓
<input type="radio"/>	prevenção de falhas ✗

Correto

Notas relativas a este envio: 1/1.

Question28

Notas: 1

Considerando avaliação do comportamento do sistema com respeito a ocorrência de falhas, associe o conceito ao tipo de avaliação:

visa identificar, classificar e ordenar os modos de defeito que podem levar a defeitos do sistema	avaliação qualitativa ▼
visa avaliar, em termos de probabilidades, a extensão em que certos atributos são satisfeitos; os atributos são então vistos como "medidas"	avaliação quantitativa ou probabilística ▼

Correto

Notas relativas a este envio: 1/1.

Question29

Notas: 1

Avizienis cita alguns métodos para avaliação quantitativa (probabilística) e qualitativa no contexto de previsão de falhas. Para os métodos listados a seguir, assinale se são adequados para avaliação qualitativa, quantitativa, ambas ou nenhuma:

redes de Petri estocásticas

avaliação quantitativa

FMEA - failure modes and effect analyses

ambas

diagramas de bloco de confiabilidade

ambas

cadeias de Markov

avaliação qualitativa

árvores de falha

ambas

Parcialmente correta

Notas relativas a este envio: 0.6/1.

Question30

Notas: 1

Avizienis cita duas abordagens principais e complementares para previsão de falhas probabilística. São elas:

Escolher uma resposta.



a. modelagem e teste de avaliação ✓



b. teste de mutação e injeção de falhas ✗



c. teste funcional e teste estrutural ✖



d. teste de caixa branca e teste de caixa preta ✖



e. teste baseado em falhas e teste baseado em critérios ✖

Correto

Notas relativas a este envio: 1/1.

INF01209 - Fundamentos de Tolerância a Falhas

Você acessou como João Luiz Grave Gross (Sair)

Moodle do INF ► FTF 2012/2 ► Questionários ► Tolerância a falhas na aplicação ► Revisão da tentativa 1

Tolerância a falhas na aplicação

Revisão da tentativa 1

Terminar revisão

Iniciado em	sexta, 14 setembro 2012, 03:47
Completado em	segunda, 24 setembro 2012, 13:37
Tempo empregado	10 dias 9 horas
Notas	27.58/28
Nota	98.49 de um máximo de 100(98%)

- 1
Notas:
1
- No item 2.1 do artigo: V. D. Florio e C. Blondia, "A survey of linguistic structures for application-level fault tolerance", *ACM Comput. Surv.*, vol. 40, no. 2, p. 6:1–6:37, maio 2008, quando argumentam sobre a necessidade de tolerância a falhas de software, os autores comentam sobre tolerância a falhas de hardware. Assinale **verdadeiro** ou **falso** segundo as opiniões expressas no texto.

Durante muitos anos, a pesquisa na área de tolerância a falhas foi concentrada em desenvolver estruturas de hardware engenhosas e eficientes para lidar com falhas.

Verdadeiro ▼

Durante algum tempo, tolerância a falhas de software foi considerada a única necessária para alcançar a disponibilidade e a integridade de dados demandadas por computadores modernos.

Falso ▼

Componentes de hardware são apenas uma das fontes de não confiabilidade em sistemas computacionais, decrescendo em importância conforme a confiabilidade desses componentes aumenta.

Verdadeiro ▼

A prevalência de falhas de hardware está aumentando continuamente.

Falso ▼

Correto

Notas relativas a este envio: 1/1.

2

Notas:
1

Ainda no item 2.1 do artigo, os autores comentam sobre a complexidade dos sistemas computacionais. Assinale **verdadeiro** ou **falso** segundo as opiniões expressas no texto.

Quanto maior o nível de abstração, maior é a complexidade dos algoritmos que entram em jogo e a consequente propensão a erro das máquinas envolvidas (sejam máquinas reais ou abstratas).

Verdadeiro ▼

A maior parte da complexidade dos modernos sistemas computacionais reside no hardware.

Falso ▼

Dividindo a complexidade do sistema em camadas, falhas ficam confinadas a uma única camada, tornando mais fácil gerenciar a complexidade crescente dos sistemas.

Verdadeiro ▼

Uma forma conceitual eficiente e flexível de gerenciar complexidade é considerar um sistema como uma hierarquia de máquinas abstratas sofisticadas.

Verdadeiro ▼

Mesmo que a divisão em camadas torne a complexidade transparente, a complexidade ainda é parte do sistema global sendo desenvolvido e falhas se propagando por várias camadas podem ocorrer.

Verdadeiro ▼

Parcialmente correta

Notas relativas a este envio: 0.8/1.

3

Notas:
1

Florio e Blondia, no item 2.2, justificam prover tolerância a falhas de software no nível da aplicação baseados no fato de que a maioria dos defeitos experimentados por computadores modernos são devidos a:

Escolher
uma
resposta.

- ☒ a. falhas de software incluindo as falhas no nível de aplicação ✓
- ☐ b. falhas em device drivers e kernel de sistemas operacionais ✗
- ☐ c. quedas de serviço ✗
- ☐ d. falhas de hardware ✗
- ☐ e. falhas de interação com o usuário ✗

Correto

Notas relativas a este envio: 1/1.

4

Notas:
1

Um dos argumentos apresentados por Florio e Blondia para justificar tolerância a falhas de software no nível da aplicação é a adoção popular de componentes de software reusáveis. Muitas aplicações orientadas a objetos são construídas com componentes reusáveis. Considerando reuso de componentes, podemos afirmar, de acordo com a opinião dos autores, que:

Escolher
uma
resposta.

- ☐ a. o reuso de componentes aumenta os custos de desenvolvimento mas contribui para aumento de confiabilidade porque os componentes já foram exaustivamente testados ✖
- ☐ b. o desenvolvedor da aplicação não conhece o desenvolvedor do componente reusável e por essa razão não pode confiar na qualidade do componente ✖
- ☒ c. o reuso de componentes diminui os custos de desenvolvimento mas a confiabilidade desses componentes e o seu impacto na confiabilidade da aplicação são frequentemente desconhecidos ✔
- ☐ d. o reuso de componentes aumenta a complexidade do projeto de sistemas e, além disso, a confiabilidade desses componentes é geralmente inadequada ✖

Correto

Notas relativas a este envio: 1/1.

5

Notas:
1

Um dos argumentos apresentados por Florio e Blondia para justificar tolerância a falhas de software no nível da aplicação, que eles consideram como provavelmente o mais convincente, é chamado de argumento fim-a-fim. Considerando esse argumento e a opinião dos autores, assinale verdadeiro e falso.

frequentemente funções só podem ser completamente e corretamente implementadas com o conhecimento e auxílio da aplicação que se encontra nas extremidades do sistema de suporte

verdadeiro ▼

soluções de tolerância a falhas baseadas puramente em hardware ou no sistema operacional não são capazes de prover tolerância a falhas completa e fim-a-fim na aplicação do usuário

verdadeiro ▼

o argumento implica que todas as tarefas de tolerância a falhas podem ser realizadas no nível da aplicação

falso ▼

como exemplo é citado que um canal de comunicação extraordinariamente confiável livra a aplicação da obrigação de prover confiabilidade

falso ▼

Correto

Notas relativas a este envio: 1/1.

6

Notas:
1

No item 2.3, os autores listam problemas de projeto relacionados a tolerância a falhas no nível da aplicação. Quais foram os problemas mencionados?

Escolher
uma
resposta.

- ☐ a. como incorporar tolerância a falhas no nível da aplicação, como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, e como gerenciar o código tolerante a falhas ✖
- ☐ b. como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, quais mecanismos de tolerância a falhas devem ser suportados, e como gerenciar o código tolerante a falhas ✖
- ☒ c. como incorporar tolerância a falhas no nível da aplicação, desenvolvedores para quais recursos de tolerância a falhas devem ser suportados, e como gerenciar o código tolerante a falhas ✓
como treinar a equipe de aplicar tolerância a falhas no nível da aplicação, quais mecanismos de tolerância a falhas devem ser suportados, e como gerenciar o código tolerante a falhas
- ☐ d. como reduzir os custos de projeto redundante, como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, e como gerenciar o código tolerante a falhas ✖

Correto

Notas relativas a este envio: 1/1.

7

Notas:
1

No item 2.3, os autores listam problemas de projeto relacionados a tolerância a falhas no nível da aplicação. Qual problema foi proposto primeiramente por Randell?

Escolher
uma

- ☐ a. como reduzir os custos de projeto ✖
- ☐ b. como treinar a equipe de desenvolvedores para aplicar

- resposta. tolerância a falhas no nível da aplicação ✖
- ☐ c. como gerenciar o código tolerante a falhas ✖
 - ☐ d. quais mecanismos de tolerância a falhas devem ser suportados ✖
 - ☒ e. como incorporar tolerância a falhas no nível da aplicação ✓

Correto

Notas relativas a este envio: 1/1.

- 8 Randell afirmou ser necessário o uso de técnicas de estruturação
Notas: apropriadas para a incorporação de tolerância a falhas no nível da aplicação.
1 Soluções inadequadas a esse problema, segundo o artigo de Florio e
Blondia, podem provocar .

Correto

Notas relativas a este envio: 1/1.

- 9 A falta de estruturação adequada, segundo o artigo de Florio e Blondia,
Notas: resulta na mistura do código funcional com o código não-funcional. Assinale
1 **verdadeiro** para as consequências ao desenvolvimento do projeto
resultantes dessa mistura, e **falso** quando a alternativa não estiver
relacionada a mistura dos códigos:

- | | |
|---|---|
| o custo e o tempo de desenvolvimento aumentam | <input type="text" value="verdadeiro"/> |
| o tamanho do código resultante é menor | <input type="text" value="falso"/> |
| o código resultante é mais fácil de ser mantido | <input type="text" value="falso"/> |
| a complexidade do sistema diminui | <input type="text" value="falso"/> |
| as tarefas do desenvolvedor ficam significativamente mais complicadas | <input type="text" value="verdadeiro"/> |

Correto

Notas relativas a este envio: 1/1.

- 10 Com relação ao problema de como incorporar tolerância a falhas no nível da
Notas: aplicação, Florio e Blondia concluem que:

1

- Escolher uma resposta.
- ☐ a. a incorporação neste nível é inadequada pois é impossível conciliar interesses ✖
 - ☐ b. os aspectos de tolerância a falhas devem prevalecer em relação aos aspectos funcionais no código da aplicação ✖
 - ☐ c. equipes diferentes de desenvolvedores devem ser encarregadas da parte funcional do código e da parte de

suporte a tolerância a falhas ✖

- ☒ d. uma estrutura ideal de sistema deve garantir uma adequada separação entre os interesses funcionais e os relativos a tolerância a falhas no código da aplicação ✔

Correto

Notas relativas a este envio: 1/1.

11

Notas:
1

Segundo Florio e Blondia, uma resposta inadequada para o problema de qual recurso de tolerância a falhas deve ser suportado pelo nível da aplicação pode apresentar duas características inconvenientes:

Escolher
uma
resposta.

- ☐ a. requerer algoritmos muito sofisticados e estratégias de verificação de alto custo ✖
- ☐ b. requerer desenvolvedores altamente especializados e ferramentas de teste sofisticadas ✖
- ☐ c. exigir equipamento especial e linguagens de programação específicas ✖
- ☒ d. requerer alto grau de redundância e rapidamente consumir a quantidade de redundância disponível ✔

Correto

Notas relativas a este envio: 1/1.

12

Notas:
1

Florio e Blondia definem 3 atributos estruturais para tolerância a falhas no nível da aplicação (ALTF) chamados de SC, SA e A. Associe:

separação de interesses funcionais e não funcionais

SC ▼

suporte direto a um grande conjunto de mecanismos de tolerância a falhas

SA ▼

facilidade do código tolerante a falhas se adaptar ao ambiente onde se encontra no momento

A ▼

Correto

Notas relativas a este envio: 1/1.

13

Notas:
1

Considerando o conceito de redundância como apresentado no item 2.3.1 do artigo de Florio e Blondia, é possível afirmar que:

Tolerância a falhas é, em geral, o resultado de alguma estratégia que efetivamente explora alguma forma de redundância.

Verdadeiro ▼

É impossível acrescentar mais confiabilidade a um canal não confiável apenas aumentando o grau de redundância de informação.

Falso ▼

Redundância pode consumir uma grande quantidade de recursos extras e rapidamente exaurir esses recursos.

Verdadeiro ▼

Redundância requer grande quantidade de recursos extras, mas isso não implica em necessariamente em aumento de custo.

Falso ▼

Correto

Notas relativas a este envio: 1/1.

14

Notas:
1

Florio e Blondia concluem o item 2.3.1 com algumas observações. Assinale verdadeiro para as alternativas que correspondem a conclusões dos autores e falso caso contrário.

Isolar a complexidade fora da aplicação do usuário provoca aumento de custos para a verificação, validação e teste.

Falso ▼

Para qualquer dado nível de redundância, balancear a complexidade do modo de defeito contra a quantidade e tipos de falhas que podem ser toleradas é uma característica importante para uma estrutura tolerante a falhas eficaz.

Verdadeiro ▼

A complexidade é um importante fonte de falhas de projeto.

Verdadeiro ▼

Correto

Notas relativas a este envio: 1/1.

15

Notas:
1

Florio e Blondia afirmam que para um sistema ser tolerante a falhas, recursos de tolerância a falhas devem também ser providos no nível da aplicação. Assinale com X as 6 classes de métodos, mencionadas no item 3 do artigo, que podem ser usadas para prover tolerância a falhas no nível de aplicação.

tolerância a falhas em software de versão única

X ▼

baseada em monitoração da tarefa de recuperação

- ▼

tolerância a falhas em software com múltiplas versões

X ▼

baseada em protocolos para metaobjetos

X ▼

programação orientada a aspectos

X ▼

redundância temporal

- ▼

tolerância a falhas embarcada

- ▼

novas linguagens de programação distribuídas

customizadas

redundância espacial

Parcialmente correta

Notas relativas a este envio: 0.78/1.

16 Qual o requisito chave para o desenvolvimento de sistemas tolerantes a falhas (de acordo com Florio e Blondia, item 3.1)?

Notas:

1

- Escolher uma resposta.
- ☐ a. Replicação de dados ✗
 - ☒ b. Recursos replicados em hardware ou em software ✓
 - ☐ c. Recursos replicados em hardware ✗
 - ☐ d. Replicação temporal ✗
 - ☐ e. Recursos replicados em software ✗

Correto

Notas relativas a este envio: 1/1.

17 A replicação é usada como método fundamental para alcançar tolerância a falhas com múltiplas computações. Segundo Florio e Blondia, quais os seus domínios ?

Notas:

1

- Escolher uma resposta.
- ☐ a. Espaço, paralelismo e codificação de dados ✗
 - ☐ b. Tempo, concorrência e paralelismo ✗
 - ☒ c. Tempo, espaço e informação ✓
 - ☐ d. Codificação de dados, tempo e concorrência ✗

Correto

Notas relativas a este envio: 1/1.

18 Considerando os domínios de replicação para alcançar tolerância a falhas, segundo Florio e Blondia, associe a explicação ao domínio:

Notas:

1

Repetição da computação

Adoção de múltiplos canais de hardware

Adoção de múltiplas versões de software

Correto

Notas relativas a este envio: 1/1.

19 Considerando os domínios de replicação sugeridos por Florio e Blondia (item

Notas: 3.1 do artigo sob análise), associe o significado aos termos na notação
1 nT/mH/pS:

número de programas

número de canais de hardware

número de execuções

Correto

Notas relativas a este envio: 1/1.

20 Um sistema 1T/1H/1S, de acordo com Florio e Blondia, é chamado de:

Notas:

1

Escolher
uma
resposta.

- ☐ a. tolerante a falhas ✖
- ☐ b. intolerante a falhas ✖
- ☐ c. duplex ✖
- ☒ d. simplex ✔
- ☐ e. replicado ✖

Correto

Notas relativas a este envio: 1/1.

21 Tolerância a falhas em versão única de software implica em:

Notas:

1

Escolher
uma
resposta.

- ☒ a. Embutir em um sistema simplex (ou seja, sem réplicas) técnicas de detecção e recuperação de erros. ✔
- ☐ b. Construir programas que não possam ser copiados nem replicados. ✖
- ☐ c. Executar software em ambientes que inibam múltiplos processos e múltiplos usuários simultâneos. ✖
- ☐ d. Construir programas aplicativos sem erros. ✖
- ☐ e. Usar externamente aos programas aplicativos as técnicas de ações atômicas, pontos de recuperação e recuperação por retorno e tratamento de exceções. ✖

Correto

Notas relativas a este envio: 1/1.

22 Segundo Florio e Blondia, a adoção de tolerância a falhas em versão única de software apresenta como desvantagens:

Notas:

1

Escolher
uma
resposta.

- ☐ a. necessidade de optar entre o sistema EFTOS e o sistema SwIFT ✖
- ☐ b. necessidade de desenvolver bibliotecas e frameworks para suprir os recursos de tolerância a falhas ✖

- ☒ c. aumento do tamanho da aplicação com consequente perda de transparência, da facilidade de manutenção e da portabilidade, acompanhados do aumento do tempo de desenvolvimento e dos custos ✓
- ☐ d. obrigação do uso estrito de processos de engenharia de software ✗

Correto

Notas relativas a este envio: 1/1.

23 Considerando o conceito de SV como apresentado por Florio e Blondia, responda verdadeiro ou falso:

Notas:

1

A adoção do SV na camada de aplicação requer concentrar no código fonte da aplicação tanto a especificação do que fazer como a estratégia de tolerância a falhas.

Verdadeiro ▼

SV baseia-se na hipótese que todos os recursos de tolerância a falhas são providos transparentemente pelo hardware, o que gera um código leve, facilmente portátil, de baixo custo e baixa necessidade de manutenção.

Falso ▼

A principal vantagem de incluir em um único código tanto a funcionalidade do sistema como as estratégias de tolerância a falhas é que essa abordagem aumenta a portabilidade do código pois o mesmo carrega consigo todos os recursos necessários.

Falso ▼

Como exemplos de técnicas de detecção e recuperação, que podem ser embutidas em software de versão única, os autores citam ações atômicas, pontos de recuperação e rollback e tratamento de exceções.

Verdadeiro ▼

Correto

Notas relativas a este envio: 1/1.


24 Uma solução parcial, segundo os autores, para a perda de portabilidade e aumento dos custos associados a embutir estratégias de tolerância a falhas em software de versão única seria:

Notas:

1

Escolher uma resposta.

- ☐ a. obrigação do uso estrito de processos de engenharia de software ✗
- ☐ b. o uso do sistema EFTOS e do sistema SwiFT ✗
- ☒ c. o uso de bibliotecas e frameworks criados seguindo criteriosamente processos de engenharia de software ✓
- ☐ d. uso eficiente de linguagens de baixo nível ✗

- ☐ e. o uso de estratégias como ações atômicas, recuperação por retorno e tratamento de exceções 

Correto

Notas relativas a este envio: 1/1.

25 Considerando o sistema SwiFT, assinale as sentenças verdadeiras e falsas.

Notas:

1

SwiFT inclui uma série de componentes reutilizáveis de software adequados para incluir técnicas de tolerância a falhas em software de versão única

Verdadeiro 

SwiFT é o nome de um computador tolerante a falhas que foi popular na década de 90

Falso 

SwiFT introduziu apenas um pequeno overhead em custo na maioria dos casos

Verdadeiro 

SwiFT foi usado com sucesso para aumentar a tolerância a falhas em sistemas de software onde falhas residuais estavam presentes

Verdadeiro 

SwiFT é o nome de um projeto que visava construir componentes de hardware para facilitar a execução de software tolerante a falhas

Falso 

Correto

Notas relativas a este envio: 1/1.

26 Florio e Blondia definem no item 2.3 os conceitos de SC (separação de interesses funcionais e de tolerância a falhas), SA (adequação sintática) e A (adaptabilidade). Em relação a software de versão única, SV, como os autores avaliam esses atributos? Indique suficiente ou insuficiente.

Notas:

1

separação de interesses

insuficiente 

adequação sintática

insuficiente 

adaptabilidade

insuficiente 

Correto

Notas relativas a este envio: 1/1.


27 Na conclusão do item 3.1.1, Florio e Blondia, após reforçarem uma série de desvantagens da abordagem de versão única, alertam para uma característica importante dessa abordagem. Qual é ela?

Notas:

1

Escolher uma resposta.

☐ a. apresenta suficiente adaptabilidade 

☐ b. apresenta um alto grau de portabilidade e facilidade de manutenção 

- ☒ c. não apresenta qualquer restrição quanto a classe de aplicação que pode se beneficiar da abordagem ✓
- ☐ d. não apresenta impacto nos custos de desenvolvimento e manutenção ✗

Correto

Notas relativas a este envio: 1/1.

28

Notas:
1

Na conclusão do item 3.1.1, Florio e Blondia, após reforçarem uma série de desvantagens, falam das vantagens de utilizar, em software de versão única, ferramentas que permitem ao usuário tratar com átomos de tolerância a falhas sem se preocupar com sua implementação. Em relação a essas ferramentas podemos afirmar que:

- Escolher uma resposta.
- ☐ a. só podem ser usadas em programas escritos em C ou C++ ✗
 - ☐ b. introduzem grande overhead no desempenho para apenas um pequeno aumento nos atributos de dependabilidade ✗
 - ☐ c. não apresentam qualquer impacto no custo ou no desempenho da aplicação ✗
 - ☐ d. prejudicam a dependabilidade do sistema ✗
 - ☒ e. permitem ao projetista reusar peças de software sofisticadas e amplamente testadas ✓

Correto

Notas relativas a este envio: 1/1.

Terminar revisão

Você acessou como [João Luiz Grave Gross \(Sair\)](#)

FTF 2012/2

Tolerância a falhas: multiversões e protocolos de meta objetos

Revisão da tentativa 1

Iniciado em	segunda, 24 setembro 2012, 13:38
Completado em	segunda, 24 setembro 2012, 14:30
Tempo empregado	52 minutos 23 segundos
Notas	28/28
Nota	100 de um máximo de 100(100%)

Question1

Notas: 1

O item 3.1.2 do artigo de Florio e Blondia (Artigo: FLORIO, V.; BLONDIA. C. *A Survey of Linguistic Structures for Application-Level Fault Tolerance*, ACM Computing Surveys, Vol. 40, No. 2, April 2008.) trata de tolerância a falhas em software multi-versão (MV). Esse tipo de abordagem requer um número mínimo de versões do mesmo programa, que é igual a:

Escolher uma resposta.



a. qualquer número ímpar para não haver empate na votação ✖



b. 2 ✓



c. 1 ✗



d. sT/yH/nS ✗



e. 3 ✗

Correto

Notas relativas a este envio: 1/1.

Question2

Notas: 1

No item 3.1.2, Florio e Blondia afirmam que a abordagem MV exige N versões de software com certa particularidade importante. Identifique essa particularidade:

Escolher uma resposta.



a. Todas as N versões devem apresentar o mesmo desempenho. ✗



b. Todas as N versões devem executar sincronizadamente sobre canais idênticos de hardware. ✗



c. O número N de versões deve ser ímpar para não haver empate na votação ✗



d. Todas as versões devem ser idênticas. ✗



e. Cada uma das N versões deve ter sido projetada e desenvolvida por um time diferente e independente. ✓

Correto

Notas relativas a este envio: 1/1.

Question3

Notas: 1

A abordagem MV tem por objetivo:

Escolher uma resposta.

- ☐ a. tolerar falhas transitórias de hardware ✗
- ☐ b. facilitar a detecção de erros ✗
- ☐ c. facilitar a manutenção do software ✗
- ☒ d. reduzir os efeitos de falhas de projeto devido a erros humanos cometidos durante o projeto ✓
- ☐ e. aumentar a testabilidade do software ✗

Correto

Notas relativas a este envio: 1/1.

Question4

Notas: 1

A abordagem MV é subdividida, segundo Florio e Blondia, em duas abordagens principais. Assinale **sim** para as abordagens abaixo que correspondem às citadas no artigo.

tratamento de exceções

não



bloco de recuperação

sim



ações atômicas distribuídas

não

programação N-versões

sim

programação concorrente

não

programação distribuída

não

Correto

Notas relativas a este envio: 1/1.

Question5

Notas: 1

A abordagem conhecida com o nome de blocos de recuperação, que foi proposta por Randell em 1975, pode ser implementada da forma NT/1H/NS ou 1T/NH/NS, sendo a primeira a mais frequente. Conhecendo essas expressões podemos afirmar que:

Escolher uma resposta.



a. A abordagem opera com N programas executados ou em série no mesmo hardware ou em paralelo sobre N canais de hardware. ✓



b. N se refere ao número de recuperações necessárias para garantir o correto funcionamento dos módulos, também chamados de blocos, do programa. ✗



c. A abordagem opera com 1 programa executado N vezes em série e N vezes em paralelo. ✗



d. A abordagem opera com N programas encadeados em série formando um único bloco de execução para fins de recuperação. ✗



e. A abordagem opera com N programas executados em paralelo no mesmo hardware. ✗

Correto

Notas relativas a este envio: 1/1.

Question6

Notas: 1

Na abordagem conhecida com o nome de blocos de recuperação, cada um dos N componentes funcionais, programas ou versões é chamado alternativa. Assinale a opção mais próxima do conceito da abordagem.

Escolher uma resposta.



a. Cada alternativa executa toda computação; os resultados são então avaliados por um teste de aceitação que calcula a média entre os resultados para compor o resultado final. ✗



b. Cada alternativa colabora com as demais executando parte da computação; as diferentes partes são então combinadas após passar por um teste de aceitação. ✗



c. O resultado de cada alternativa, começando pela primária, é avaliado por um teste de aceitação; caso não seja aceito, o resultado da próxima alternativa é avaliada e assim sucessivamente até que um dos resultados passe no teste de aceitação. ✓



d. Cada alternativa executa toda computação; os resultados são então comparados por um teste de aceitação que escolhe o melhor. ✗



e. A cada execução é escolhida uma nova alternativa para que os programas não acumulem erros residuais. ✗

Correto

Notas relativas a este envio: 1/1.

Question7

Notas: 1

A eficácia da abordagem de blocos de recuperação reside na cobertura do mecanismo de detecção de erros adotado, portanto o componente mais crucial da abordagem é:

Escolher uma resposta.

- ☐ a. a alternativa primária ✗
- ☒ b. o teste de aceitação ✓
- ☐ c. a cache de recuperação ✗
- ☐ d. o chaveamento entre as alternativas ✗
- ☐ e. o suporte para o mecanismo de recuperação ✗

Correto

Notas relativas a este envio: 1/1.

Question8

Notas: 1

Florio e Blondia, analisando blocos de recuperação, apontam alguns pontos positivos e negativos relacionados a abordagem. Assinale verdadeiro se a opção corresponde a uma afirmação encontrada no artigo e falso se não corresponde.

A abordagem de blocos de recuperação foi validada com sucesso através de experimentos estatísticos e modelagem matemática.

verdadeiro



Blocos de recuperação não são uma abordagem satisfatória considerando intrusividade no código, portabilidade e facilidade de manutenção.

verdadeiro



A abordagem de blocos de recuperação consiste em uma estratégia rígida sem possibilidade de configuração offline e com baixa adaptabilidade.

verdadeiro



Um aspecto positivo da abordagem de blocos de recuperação é que o custo de manutenção e desenvolvimento independe do número de réplicas, apesar de depender do número de canais de hardware.

falso



Não existem registros que a abordagem tenha sido usada com sucesso em algum campo de aplicação.

falso



Correto

Notas relativas a este envio: 1/1.

Question9

Notas: 1

Segundo Florio e Blondia, sistemas de programação N-versões (NVP) são baseados em:

Escolher uma resposta.



a. execução paralela e replicação de código ✗



b. redundância e recuperação ✗



c. N versões do mesmo programa executados sequencialmente ✗



d. redundância e consenso ✓



e. concorrência e teste de aceitação ✗

Correto

Notas relativas a este envio: 1/1.

Question10

Notas: 1

O autor que introduziu o conceito de programação N-versões foi:

Escolher uma resposta.

- ☐ a. Laprie ✗
- ☒ b. Avizienis ✓
- ☐ c. Randell ✗
- ☐ d. Florio ✗
- ☐ e. Blondia ✗

Correto

Notas relativas a este envio: 1/1.

Question11

Notas: 1

Na abordagem conhecida como programação N-versões, NVP, cada um dos N componentes funcionais é chamado versão. O autor da abordagem define que as versões devem ser geradas de forma . O número N de versões deve ser . As versões funcionalmente equivalentes devem ser desenvolvidas a partir .

Correto

Notas relativas a este envio: 1/1.

Question12

Notas: 1

Na abordagem conhecida como programação N-versões, cada um dos N componentes funcionais é diferente dos demais e é chamado versão. Assinale a opção mais próxima do conceito da abordagem.

Escolher uma resposta.



a. Cada versão colabora com as demais executando parte da computação; as diferentes partes são então combinadas após passar por consenso. ✗



b. Cada versão executa toda computação; os resultados são então processados por um algoritmo genérico de decisão e é escolhido o resultado final por consenso ou maioria. ✓



c. Cada versão executa toda computação; os resultados são então avaliados por um teste de aceitação que escolhe o mais correto com o resultado final. ✗



d. A cada execução é escolhida uma nova versão para que os programas não acumulem erros residuais. ✗



e. O resultado de cada versão, começando pela primária, é avaliado por um teste de consenso; caso não seja aceito, o resultado da próxima versão é avaliado e assim sucessivamente até que um dos resultados passe no teste. ✗

Correto

Notas relativas a este envio: 1/1.

Question13

Notas: 1

A abordagem conhecida como programação N-versões é baseada em uma conjectura fundamental, qual seja:

Escolher uma resposta.



a. as N versões diferentes executadas em paralelo replicam as falhas residuais de software que se encontram dormentes do código fonte ✗



b. projetos independentes se traduzem em defeitos randômicos dos componentes, ou seja, independência estatística dos defeitos ✓



c. projetos independentes resultam em defeitos correlacionados de componentes, que rapidamente exaurem a redundância disponível ✗



d. usar N cópias do mesmo programa ou N versões diferentes não influencia a tolerância a falhas de um sistema ✗

Correto

Notas relativas a este envio: 1/1.

Question14

Notas: 1

Considerando a abordagem de programação N-versões, assinale verdadeiro ou falso de acordo com o artigo de Florio e Blondia.

Avizienis afirma que a geração das versões de forma independente reduz significativamente os defeitos correlacionados.

Verdadeiro



O uso de N cópias da mesma versão evita a replicação de falhas dormentes residentes no código da versão.

Falso



Vários experimentos e estudos teóricos mostram que nem sempre é correto assumir que a geração das versões de forma independente reduza significativamente os defeitos correlacionados.

Verdadeiro



Correto

Notas relativas a este envio: 1/1.

Question15

Notas: 1

Comparando as duas abordagens, blocos de recuperação e programação N-versões, selecione os recursos adequados.

Mecanismo usado para seleção de resposta na abordagem conhecida como blocos de recuperação:

teste de aceitação

Mecanismo usado para seleção de resposta na abordagem conhecida como programação N-versões:

consenso ou algoritmo de votação

Correto

Notas relativas a este envio: 1/1.

Question16

Notas: 1

Comparando as duas abordagens, blocos de recuperação e programação N-versões, em relação ao mecanismo usado para seleção de resultado, podemos afirmar que a primeira abordagem usa um mecanismo específico dependente da aplicação enquanto a segunda usa um mecanismo genérico independente da aplicação.

Correto

Notas relativas a este envio: 1/1.

Question17

Notas: 1

Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite diferentes saídas corretas enquanto a segunda abordagem permite uma única saída correta.

Correto

Notas relativas a este envio: 1/1.

Question18

Notas: 1

Comparando blocos de recuperação e programação N-versões quanto a estratégia de execução, podemos afirmar que a primeira abordagem em sua forma original implementa uma estratégia de enquanto a segunda abordagem permite .

Correto

Notas relativas a este envio: 1/1.

Question19

Notas: 1

Considerando conclusões de Florio e Blondia a respeito de programação N-versões, podemos afirmar que (assinale **verdadeiro** ou **falso**):

A aplicabilidade da programação N-versões está restrita a um pequeno número de linguagens de programação como Java e C++.

Falso

O componente conhecido como EE (executivo do ambiente de programação N-versões) é simples, mas dependente da aplicação.

Falso

Programação N-versões não é adequada a sistemas de tempo-real.

Falso

Programação N-versões foi adotada com sucesso em várias áreas de aplicação incluindo aplicações de segurança crítica espaciais e aéreas.

Verdadeiro

Correto

Notas relativas a este envio: 1/1.

Question20

Notas: 1

Em relação a programação N-versões, assinale **verdadeiro** ou **falso** para as sentenças a seguir:

Portabilidade da programação N-versões é restrita pela portabilidade do executivo do ambiente N-versões e pela portabilidade de cada uma das versões.

Verdadeiro



A adoção de programação N-versões sempre implica em penabilidades em relação a manutenibilidade e portabilidade.

Verdadeiro



A programação N-versões nas arquiteturas 1T/NH/NS e NT/1H/NS tem uma função de custo que cresce linearmente com o número de versões.

Falso



A adoção de programação N-versões implica em um aumento substancial no custo de desenvolvimento e manutenção.

Verdadeiro



Correto

Notas relativas a este envio: 1/1.

Question21

Notas: 1

No item 3.2 do artigo de Florio e Blondia aparece a sigla MOP. MOP significa:

Escolher uma resposta.

☐

a. multiple object programming ✗

☐

b. meta operating system partitioning ✗

☐

c. maintainable object processing ✗

☒

d. metaobject protocol ✓

☐

e. multiple operating processes ✗

Correto

Notas relativas a este envio: 1/1.

Question22

Notas: 1

Segundo Florio e Blondia, técnicas de estruturação genéricas como o método MOPs (item 3.2 do artigo) permite, em alguns casos, alcançar graus adequados de:

Escolher uma resposta.



a. facilidade na implementação de técnicas de tolerância a falhas em qualquer linguagem de programação ✗



b. adaptabilidade e portabilidade ✗



c. desempenho e dependabilidade ✗



d. correção e desempenho ✗



e. flexibilidade, transparência e separação de interesses funcionais e não funcionais ✓

Correto

Notas relativas a este envio: 1/1.

Question23

Notas: 1

Segundo Florio e Blondia, a ideia básica do método MOPs é:

Escolher uma resposta.



a. adotar uma linguagem orientada a objetos como C++ ou Java e protocolos de comunicação certificados pelos maiores fabricantes para facilitar a interoperabilidade ✗

- ☐ b. abrir a implementação dos protocolos de comunicação inter-objetos de forma a inserir comandos específicos e padronizados de tolerância a falhas que os objetos são obrigados a executar ✗
- ☒ c. abrir a implementação do executivo de execução (runtime executive) de uma linguagem orientada a objetos de tal forma que o programador possa adotar semânticas diferentes e customizadas, ajustando assim a linguagem para as necessidades do usuário e do ambiente ✓
- ☐ d. abrir a implementação do compilador de uma linguagem de alto nível de tal forma que o programador possa adotar semânticas diferentes e customizadas, ajustando assim a linguagem para as necessidades do usuário e do ambiente ✗
- ☐ e. abrir a implementação do sistema operacional de tal forma que o programador possa desenvolver novas chamadas do sistema (sys call), ajustando assim o sistema operacional para as necessidades do usuário e do ambiente ✗

Correto

Notas relativas a este envio: 1/1.

Question24

Notas: 1

Usando MOPs o programador pode modificar o comportamento de características fundamentais como invocação de métodos, criação e destruição de objetos. Para fins de implementação de técnicas de tolerância a falhas, essa possibilidade de modificação permite:

Escolher uma resposta.

- ☐ a. a substituição dinâmica de programas errados do usuário por programas corretos do sistema ✗
- ☐ b. o aumento significativo do desempenho ✗
- ☐ c. a verificação automática da correção de programas ✗
- ☒ d. a gerência transparente de redundância espacial e temporal ✓

Correto

Notas relativas a este envio: 1/1.

Question25

Notas: 1

O conceito chave de suporte a MOPs (item 3.2 do artigo de Florio e Blondia) é:

Escolher uma resposta.



a. orientação a aspectos ✗



b. reflexão computacional ✓



c. objetos reais ✗



d. Java ✗



e. runtime executive ✗

Correto

Notas relativas a este envio: 1/1.

Question26

Notas: 1

MOPs oferecem ao programador de metanível uma representação de um sistema como um conjunto de elementos que representam e refletem propriedades de objetos reais. Esses elementos são chamados de:

Escolher uma resposta.



a. objetos tolerantes a falhas ✗

- ☐ b. objetos funcionais ✗
- ☐ c. objetos confiáveis ✗
- ☐ d. objetos de tempo real ✗
- ☒ e. metaobjetos ✓

Correto

Notas relativas a este envio: 1/1.

Question27

Notas: 1

De acordo com Florio e Blondia, item 3.2, preencha as lacunas:

O programador dos recursos de tolerância a falhas define uma quantidade de e os associa com . A cada vez que o programa funcional o/a correspondente é .

Correto

Notas relativas a este envio: 1/1.

Question28

Notas: 1

Segundo Florio e Blondia, a abordagem baseada em MOPs parece constituir uma técnica promissora para a adoção efetiva, coerente e transparente de mecanismos e técnicas de TF. Assinale verdadeiro ou falso:

Nenhuma evidência experimental ou analítica permite estimar a praticidade e generalidade da abordagem MOPs.

Alguns estudos na década de 90 comprovam a eficiência de MOPs em alguns casos.

Verdadeiro



Não há dúvidas que MOPs representa uma solução prática para a efetiva integração na aplicação de usuários da maioria dos mecanismos de TF existentes.

Falso



Correto

Notas relativas a este envio: 1/1.

INF01209 - Fundamentos de Tolerância a Falhas

Você acessou como João Luiz Grave Gross (Sair)

Moodle do INF ► FTF 2012/2 ► Questionários ► Redundância, confiabilidade e disponibilidade ► Revisão da tentativa 1

Redundância, confiabilidade e disponibilidade

Revisão da tentativa 1

Terminar revisão

Iniciado em	domingo, 7 outubro 2012, 12:34
Completado em	domingo, 7 outubro 2012, 16:07
Tempo empregado	3 horas 33 minutos
Notas	30.9/36
Nota	85.83 de um máximo de 100(86%)

- 1
- Notas: 1
- Johnson, B.W. no artigo: "*Fault Tolerance*", *The Electrical Engineering Handbook*, Ed. Richard C. Dorf Boca Raton: CRC Press LLC, 2000, apresenta inicialmente uma breve introdução aos conceitos de tolerância a falhas. Observe que a nomenclatura usada algumas vezes difere levemente da de outros autores.

De acordo com Johnson, o que significa tolerância a falhas?

- Escolher uma resposta.
- ☒ a. a habilidade do sistema de continuar a realizar corretamente suas tarefas depois da ocorrência de falhas de hardware ou software ✓
 - ☐ b. a habilidade do sistema de prover replicação física de hardware para garantir redundância passiva ✗
 - ☐ c. garantia de correção do serviço fornecido por um sistema ✗
 - ☐ d. garantia que um sistema não vai provocar danos a pessoas ou ao meio ambiente ✗
 - ☐ e. habilidade do sistema de reconhecer e sinalizar a ocorrência de um defeito ✗

Correto

Notas relativas a este envio: 1/1.

2 De acordo com Barry Johnson, o que é uma falha?

Notas:

1

Escolher
uma
resposta.

- ☐ a. A falha é o resultado de uma propagação de eventos que provoca a queda de um dado serviço. ✖
- ☒ b. A falha é simplesmente qualquer dano físico, imperfeição, ou problema que ocorre em hardware ou software. ✔
- ☐ c. A falha é um fenômeno físico provocado unicamente por desgaste de componentes de hardware. ✖
- ☐ d. Uma falha é caracterizada quando um serviço se desvia da sua especificação. ✖
- ☐ e. A falha é a consequência de um projeto mal especificado. ✖

Correto

Notas relativas a este envio: 1/1.

3

Notas:

1

Barry Johnson cita quatro áreas de aplicação de computação tolerante a falhas: longa vida, computação crítica, manutenção adiada e alta disponibilidade. Associe os exemplos à área de aplicação.

alguns tipos de controladores industriais

computação crítica ▼

estações de processamento remoto

manutenção adiada ▼

sistemas de controle de vôo

computação crítica ▼

automação bancária

alta disponibilidade ▼

satélites

longa vida ▼

Correto

Notas relativas a este envio: 1/1.

4

Notas:

1

Barry Johnson afirma que tolerância a falhas pode ser alcançada em um sistema incorporando várias formas de redundância. O autor cita quatro formas de redundância, são elas:

Escolher
uma
resposta.

- ☐ a. redundância de especificação, de documentação, de teste e de equipamentos de hardware ✖
- ☐ b. redundância espacial, gerencial, temporal e orçamentária ✖
- ☐ c. redundância gerencial, temporal, de software e de componentes de hardware ✖
- ☒ d. redundância de software, redundância gerencial,

informação, tempo e hardware ✓ temporal, de software e de componentes de hardware

- ☐ e. redundância de equipes de desenvolvimento, documentação, procedimentos e componentes de hardware ✗

Correto

Notas relativas a este envio: 1/1.

5

Notas: 3 Redundância passiva é uma forma de redundância de que usa o conceito de para esconder a ocorrência da falha e prevenir que a falha se manifeste como .

Correto

Notas relativas a este envio: 3/3.

6

Notas: 3 Segundo Barry Johnson, redundância ativa é uma forma de redundância de que usa o conceito de para, após a ocorrência da falha, remover a falha do sistema. Técnicas de redundância ativa requerem que o sistema realize para tolerar falhas.

Correto

Notas relativas a este envio: 3/3.

7

Associe a técnica à forma de redundância que a emprega.

Notas:

1

localização de falha	<input type="text" value="redundância ativa"/>
recuperação de falha	<input type="text" value="redundância ativa"/>
detecção de falha	<input type="text" value="redundância ativa"/>
mascaramento de falha	<input type="text" value="redundância passiva"/>

Parcialmente correta

Notas relativas a este envio: 0.5/1.

8

Associe o exemplo à forma de redundância.

Notas:

1

TMR	<input type="text" value="redundância passiva"/>
standby sparing (estepe)	<input type="text" value="redundância ativa"/>

Correto

Notas relativas a este envio: 1/1.

9

Notas:
1

Avizienis quando apresenta técnicas de tolerância a falhas as divide em dois grande grupos, detecção de erros e recuperação, essa última englobando tratamento do erro e tratamento da falha. A classificação de Barry Johnson é diferente. Johnson fala em detecção de falhas (o que seria equivalente a detecção de erros do Avizienis) e mascaramento de falhas. Onde o "mascaramento de falhas" se encaixa na classificação de Avizienis?

Escolher
uma
resposta.

- ☐ a. detecção de erros concorrente ✖
- ☐ b. tratamento de falha por recuperação ✖
- ☐ c. tratamento de erro por rollforward ✖
- ☒ d. tratamento de falha por isolamento ✖
- ☐ e. tratamento de erro por compensação ✔

Errado

Notas relativas a este envio: 0/1.

10

Notas:
1

O que é a distância de Hamming entre duas palavras binárias?

Escolher
uma
resposta.

- ☐ a. número de bits em 1 nas duas palavras ✖
- ☐ b. quantidade de retransmissões necessárias para as palavras serem transmitidas sem erro ✖
- ☐ c. diferença entre as paridades da palavras ✖
- ☒ d. o número de posições em que os bits diferem ✔
- ☐ e. número de bits errados ✖

Correto

Notas relativas a este envio: 1/1.

11

Notas:
1

Determine a distância de Hamming das seguintes duplas

0101 e 1010

4 ▼

0011 e 0110

2 ▼

0000 e 1000

1 ▼

1111 e 0000

4 ▼

Correto

Notas relativas a este envio: 1/1.

12

Notas:
1

Determine a distância de código dos seguintes códigos binários (cada linha é um código completo):

000, 011, 101, 110

 ▾

00, 01, 10, 11

 ▾

001, 010, 100, 111

 ▾

0101, 1010

 ▾

Correto

Notas relativas a este envio: 1/1.

13 O que é a distância de um código formado de palavras binárias?

Notas:

1

Escolher
uma
resposta.

- ☒ a. menor distância de Hamming entre quaisquer duas palavras válidas do código ✓
- ☐ b. números de bits necessários para codificar as palavras ✗
- ☐ c. número máximo de bits errados que podem ser corrigidos ✗
- ☐ d. número máximo de erros que podem ser detectados ✗
- ☐ e. quantidade máxima de retransmissões necessárias para que qualquer palavra seja transmitida sem erro ✗

Correto

Notas relativas a este envio: 1/1.

14 O código de paridade, seja par ou ímpar, apresenta uma distância de

Notas: Hamming de ▾; consequentemente pode detectar ▾ bit(s) com
3 erro e corrigir ▾ bit(s) com erro.

Parcialmente correta

Notas relativas a este envio: 1/3.

15 O conceito fundamental associado a redundância no tempo, segundo Barry Johnson, é:

Notas:

1

Escolher
uma
resposta.

- ☒ a. repetir a computação em dois ou mais computadores e comparar os resultados para verificar se existe discrepância ✗
- ☐ b. repetir a computação duas ou mais vezes e comparar os resultados para verificar se existe discrepância ✓
- ☐ c. sincronizar os relógios internos de dois ou mais computadores para que os resultados sejam fornecidos exatamente no mesmo instante de tempo ✗
- ☐ d. repetir n vezes uma computação e realizar a média dos resultados ✗

- ☐ e. repetir exaustivamente uma computação até que o resultado seja correto ✖

Errado

Notas relativas a este envio: 0/1.

16 O problema principal associado a redundância no tempo, segundo Barry Johnson, é:

Notas:
1

- Escolher uma resposta.
- ☐ a. sincronizar os diversos computadores que operam em paralelo ✖
 - ☐ b. o número de repetições necessárias prejudica o desempenho do sistema ✖
 - ☐ c. não ser adequada para detecção de falhas transitórias ✖
 - ☐ d. não ser adequada para detecção de falhas permanentes ✖
 - ☒ e. garantir que o sistema tenha os mesmos dados para manipular a cada vez que ele realiza a computação redundante ✔

Correto

Notas relativas a este envio: 1/1.

17 Responda verdadeiro ou falso segundo as afirmações de Barry Johnson.

Notas:
1

Mesmo que uma falha transiente ocorra e corrompa completamente os dados de entrada, ainda assim repetir a computação no tempo é suficiente para tolerar falhas transientes.

falso ▼

Para tolerar falhas transientes usando redundância temporal é necessário codificar os dados antes da transmissão da segunda computação e decodificá-los no recebimento.

falso ▼

Um dos maiores potenciais de redundância temporal é a habilidade de detectar falhas permanentes usando apenas um mínimo de hardware extra.

verdadeiro ▼

Correto

Notas relativas a este envio: 1/1.

18 Barry Johnson apresenta 3 exemplos de redundância de software. São eles:

Notas:
1

- Escolher uma
- ☐ a. Programação auto-verificadora, programação ativa e programação passiva. ✖

- resposta.
- ☐ b. Programação modular, programação ágil e programação extrema. ✗
 - ☒ c. Programação N-versões, blocos de recuperação e programação N- auto-verificadora. ✓
 - ☐ d. Programação com estepe quente, programação com estepe frio e programação diversitária. ✗
 - ☐ e. Programação N- auto-verificadora, verificação de consistência e verificação de capacidade. ✗

Correto

Notas relativas a este envio: 1/1.

19

Notas:

1

As 3 abordagens de redundância de software mencionadas por Barry Johnson têm em comum:

Escolher
uma
resposta.

- ☒ a. exigir N versões diferentes de um mesmo programa ✓
- ☐ b. executar todas as versões do programa em paralelo ✗
- ☐ c. votar sobre o resultado de N versões ✗
- ☐ d. tolerar N-1 falhas ✗
- ☐ e. exigir um teste de aceitação para cada versão do programa ✗

Correto

Notas relativas a este envio: 1/1.

20

Notas:

1

Na técnica de redundância conhecida como programação N-versões, o número de falhas que podem ser toleradas com N versões de um programa é:

Escolher
uma
resposta.

- ☐ a. $N / 2$ ✗
- ☒ b. $(N - 1) / 2$ ✓
- ☐ c. N ✗
- ☐ d. $(N + 1) / 2$ ✗
- ☐ e. $N - 1$ ✗

Correto

Notas relativas a este envio: 1/1.

21

Notas:

1

Na técnica de redundância conhecida como programação N-auto-verificadora (N self-checking), o número de falhas que podem ser toleradas com N versões de um programa é:

Escolher
uma
resposta.

- ☐ a. $(N - 1) / 2$ ✗
- ☐ b. $(N + 1) / 2$ ✗

☒ c. N - 1 ✓

☐ d. N ✗

☐ e. N / 2 ✗

Correto

Notas relativas a este envio: 1/1.

22
Notas:
1

Responda segundo os conceitos de redundância de software do artigo de Barry Johnson. Associe a forma de seleção do resultado do sistema redundante à técnica de redundância de software.

votação entre os resultados produzidos por todas as versões

programação N-versões ▼

testes de aceitação para cada versão e uma lógica de seleção entre os resultados dos programas que passam no seu teste de aceitação

programação N auto-verificadora ▼

um único conjunto de testes de aceitação para todas as versões e versões executadas em sequência até que a primeira versão passe nos testes de aceitação

blocos de recuperação ▼

Correto

Notas relativas a este envio: 1/1.

23
Notas:
1

Considerando os conceitos de avaliação de dependabilidade do artigo de Barry Johnson, associe a definição ao termo. Observe que os conceitos podem ser diferentes dos enunciados por Avizienis.

qualidade de serviço provido por um sistema

dependabilidade ▼

probabilidade de que o sistema estará disponível para realizar suas tarefas no instante de tempo t

disponibilidade ▼

probabilidade condicional de que um componente funcione corretamente durante um intervalo de tempo $[t_0, t]$ dado que esteja operacional no tempo t_0

confiabilidade ▼

número esperado de defeitos de um tipo de dispositivo ou equipamento por um dado período de tempo

taxa de defeitos ▼

Correto

Notas relativas a este envio: 1/1.

24 Em relação a função de taxa de defeitos podemos afirmar que (responda verdadeiro ou falso):

Notas:

1

é uma função monotonicamente crescente

falso

é claramente constante

falso

é claramente dependente do tempo

verdadeiro

pode ser expressa em termos da função de confiabilidade

Escolher...

inicialmente descrece, fica aproximadamente constante por um longo período, e depois cresce exponencialmente

falso

Parcialmente correta

Notas relativas a este envio: 0.6/1.

25 A relação entre confiabilidade e tempo é conhecida como lei de defeitos exponenciais. Ela estabelece que:

Notas:

1

Escolher uma resposta.

- ☐ a. A confiabilidade de um componente permanece constante até o final da vida útil, quando então diminui exponencialmente. ✖
- ☐ b. A confiabilidade de um componente aumenta exponencialmente com o tempo. ✖
- ☐ c. A confiabilidade de um componente aumenta e diminui exponencialmente com o tempo seguindo uma curva inversa a curva da banheira. ✖
- ☒ d. A confiabilidade de um componente, para uma taxa de defeitos constante, diminui exponencialmente com o tempo. ✔
- ☐ e. A confiabilidade de um componente aumenta e diminui exponencialmente com o tempo seguindo a curva da banheira. ✖

Correto

Notas relativas a este envio: 1/1.

26 Associe a explicação ao termo.

Notas:

1

tempo médio entre defeitos de um sistema

MTBF

integral de 0 a infinito da confiabilidade de um sistema, considerando que a função de confiabilidade seja nula no tempo infinito

MTTF

tempo esperado no qual o sistema irá operar até que o primeiro defeito ocorra

MTTF

tempo médio requerido para reparar um sistema

MTTR

Correto

Notas relativas a este envio: 1/1.

27 Responda verdadeiro ou falso:

Notas:
1

(MTBF = MTTF + MTTR) é uma relação que vale apenas quando o reparo do sistema torna o sistema novamente perfeito, ou seja, assim como ele era quando novo

verdadeiro

em muitas aplicações práticas, o MTTR é apenas uma fração pequena do MTTF, por essa razão muitas vezes é considerado que o MTBF é uma aproximação do MTTF

verdadeiro

MTTF e MTBF são valores praticamente iguais e podem ser usados alternadamente, mesmo quando a taxa de reparos for desprezível

falso

Correto

Notas relativas a este envio: 1/1.

28 O que significa cobertura de falhas?

Notas:
1

Escolher uma resposta.

- ☐ a. cobertura de falhas é matematicamente definida como o tempo médio medido desde a ativação de uma falha até o sistema detectar o erro provocado pela falha ✗
- ☐ b. cobertura de falhas é matematicamente definida como o tempo médio de operação do sistema após a ocorrência de uma falha até o sistema apresentar o defeito decorrente daquela falha ✗
- ☐ c. cobertura de falhas é matematicamente definida como a diferença entre a taxa de reparos e da taxa de defeitos ✗
- ☒ d. cobertura de falhas é matematicamente definida a probabilidade condicional que dada a existência de uma falha o sistema se recupere ✓

Correto

Notas relativas a este envio: 1/1.

29 A confiabilidade de um sistema geralmente é calculada a partir da confiabilidade dos seus componentes individuais. Para isso são considerados dois modelos: série e paralelo. Responda:

1

em que modelo a confiabilidade do sistema pode ser calculada como a probabilidade de que nenhum dos componentes irá apresentar defeito?

série ▼

em que modelo se assume que os defeitos dos elementos são independentes entre si?

paralelo ▼

em que modelo é requerido que apenas um componentes do sistema opere corretamente para o sistema operar corretamente?

paralelo ▼

em que modelo é requerido que todos os componentes do sistema operem corretamente para o sistema operar corretamente?

série ▼

em que modelo a taxa de defeitos do sistema pode ser calculada como a soma das taxas de defeitos de todos os componentes?

paralelo ▼

Parcialmente correta

Notas relativas a este envio: 0.8/1.

30

Notas:
1

As taxas de defeito e de reparo estão associadas com duas medidas de dependabilidade expressas como tempos médios. Estabeleça qual a associação.

inverso da taxa de defeitos

MTTF ▼

inverso da taxa de reparos

MTTR ▼

Correto

Notas relativas a este envio: 1/1.

Terminar revisão

Você acessou como [João Luiz Grave Gross \(Sair\)](#)

FTF 2012/2
