

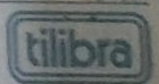
-> Questionário:

1) 1º) Mariner (1962), um foguete com uma sonda espacial para Vênus, foi desviado de sua procura de novo logo após o lançamento. Um programador passou errado uma fórmula para o software. Prejuízo de 28,5 milhões de dólares.

2º) Crash na Wall Street (1987), o índice Dow Jones caiu 508 pontos. Houve um número muito grande de ordens de venda gerados pelas computadores, quebrando sistemas e deixando os investidores efetivamente cegos. Prejuízo de 500 bilhões de dólares.

3º) Bug do milênio (1999), para economizar espaço de armazenamento, usava-se o armazenamento de anos como 2 dígitos, como 99 para 1999. Quando o ano 2000 chegou eles interpretaram 00 como 1900, gerando uma série de bugs. Prejuízo de 500 bilhões de dólares.

2) O objetivo da Garantia de Qualidade (SQA) é prover o gerenciamento, como discutido no resumo deste trabalho, com a adequada responsabilidade do processo que está sendo utilizado pelo projeto de software e dos produtos que estão sendo construídos. O objetivo é subsidiar o projeto de software e os gerentes envolvidos com o resultado de revisões e auditoria. Entre suas vantagens podemos citar um software com menos bugs, menos erros e



/ /

mais facilmente corrigível.

3) Necessidades implícitas são aquelas que se inserem na expectativa ou desejo do cliente. Exemplos: incentivar preços mais baratos e uma boa aparência da interface do software. Já as explícitas são os benefícios ofertados pelo software, ou seja, tudo quanto se estabelece em contrato ou na especificação de projetos, material de propaganda. Exemplos: marketing do software e fazer o que é proposto no protótipo de telas do projeto inicial.

4) Numere a segunda coluna de acordo com a primeira:

1) Teste de funcionalidade 2) Teste de Usabilidade 3) Teste de Carga (Stress) 4) Teste de Regressão 5) Teste de Volume 6) Teste de Configuração 7) Teste de Compatibilidade 8) Teste de Segurança 9) Teste de Desempenho 10) Teste de Instalação 11) Teste de Confiabilidade 12) Teste de Disponibilidade 13) Teste de Recuperação 14) Teste de Contingência.

- 2 () tem por objetivo verificar a facilidade que o software possui de ser claramente compreendido e manipulado pelo usuário na sua navegação (utilização).
- 7 () teste que executa alguns ou todos os casos de testes já verificados, a fim de detectar erros gerados por alterações ou correções feitas durante o processo de desenvolvimento ou manutenção do software.
- 6 () teste que verifica todas as configurações necessárias para garantir um bom funcionamento do software.
- 13 () teste também conhecido como plano de continuidade de negócios ou plano de recuperação de desastres, tem como intuito descrever as medidas a serem tomadas para restabelecer o sistema a seu estado funcional ou estado minimamente aceitável em caso de paralisação do sistema.
- 7 () teste que garante a compatibilidade com todos artefatos
- 12 () teste que verifica o nível de disponibilidade do software, esta fase tem como objetivo garantir que o sistema estará disponível mesmo com pequenas falhas no ambiente de produção.
- 5 () teste que avalia um sistema simulando uma situação predeterminada, geralmente um grande volume de carga ou dados. Tem por objetivo identificar gargalos originados por baixo dimensionamento de link, falta de otimização da infraestrutura ou implementação ineficiente do aplicativo.
- 8 () teste de segurança visa verificar se todos os mecanismos de proteção embutidos no sistema eliminam as vulnerabilidades do software.
- 9 () teste que avalia o tempo de resposta, os recursos utilizados e eficiência de cada função.
- 1 () teste que avalia um sistema ou componente executando de modo que a demanda de recursos ocorre em quantidade frequente ou volume irregular.
- 10 () teste que verifica se o processo de instalação está executando todos os passos requeridos para implementação do sistema no ambiente de utilização.
- 14 () teste que valida a confidencialidade das informações, geralmente este teste é aplicado em conjunto com o teste de segurança.
- 3 () teste de sistema que força o software a falhar de diversas maneiras e verifica se a recuperação é adequadamente executada.

5) Um software de qualidade é relativo, como vimos no resumo deste trabalho, o que agrada um cliente, pode ou não agradar outro, entretanto plataformas diferentes possuem necessidades diferentes. Por exemplo, um site e-commerce, para ser de qualidade deve garantir que as compras sejam feitas de forma segura, já um serviço de streaming deve garantir que os dados da transmissão cheguem a todos os usuários, e por fim a cloud computing deve assegurar que os dados jamais serão perdidos, através de backups, por exemplo.

6) A qualidade de produto se relaciona diretamente com as necessidades e desejos do cliente, já a qualidade de processos o que se busca é definir como aquele item ou serviço deve ser fabricado ou prestado da melhor forma possível.

7) A ISO 9126 é uma norma para qualidade de produto de software. Ela define um conjunto de parâmetros com o objetivo de padronizar a avaliação da qualidade de software. As características definidas são aplicáveis a todo tipo de software, incluindo programas de computadores e dados contidos em firmware.

8) Manutenibilidade é uma característica inerente a um projeto de sistema ou produto, e se refere à facilidade, precisão, segurança e economia na execução de ações de manutenção nesse



sistema ou produto.

9) A portabilidade de um software é a sua capacidade de ser compilado ou executado em diferentes arquiteturas. O termo também pode ser usado para se referir a re-escrita de um código fonte para outra linguagem de programação.

10) O six sigma é um conjunto de práticas aplicadas para eliminar defeitos a partir da redução da variação dos processos. Ele trás uma estratégia gerencial para melhorar processos, reduzindo a variabilidade e otimizando os resultados. O grande problema da implementação do six sigma no âmbito de desenvolvimento de software diz respeito à retrabalho ou mesmo em alguns processos pode não funcionar, por exemplo, um teste de software não pode ser simplificado porque qualquer tipo de erro que surgir depois será muito pior.

11) • Qualidade Funcional é aquela que assegura a execução consistente e a melhoria contínua das atividades funcionais de rotina satisfazendo as necessidades e expectativas dos clientes.

• Qualidade estrutural diz respeito à qualidade do código escrito e o quão bem estruturado ele é.

• Qualidade de processo é o conjunto de melhores práticas no desenvolvimento de projetos (adesão de normas e padrões).