

1. São atributos importantes que todo software profissional deve possuir:

Manutenabilidade; confiança e proteção; eficiência; aceitabilidade;

Heterogeneidade; confiança e proteção; eficiência; aceitabilidade;

Segurança e confiança; disponibilidade; manutenabilidade; aceitabilidade;

Aceitabilidade; segurança e confiança; heterogeneidade; facilidade de uso;

Facilidade de uso; manutenabilidade; proteção; eficiência;

2. Existem muitos processos de software diferentes, mas todos devem incluir quatro atividades fundamentais para a engenharia de software:

Especificação de software; projeto e implementação; validação de software; evolução de software.

Especificação de software; análise e projeto; Implementação; teste.

Modelagem de negócios; análise e projeto; implementação; teste.

Requisitos; gerenciamento de configuração e mudança; validação de software; evolução de software.

Especificação de software; projeto e implementação; gerenciamento de configuração e mudança; evolução de software.

3. Modelos de processo de software são representações simplificadas de um ciclo de vida de software. A abordagem que intercala as atividades de especificação, desenvolvimento e validação em forma de versões é o:

Desenvolvimento incremental

Modelo em cascata

Engenharia de software orientado à reuso

Rational Unified Process (RUP)

Rapid Application Development (RAD)

4. Atualmente, as empresas de software tendem a utilizar modelos de desenvolvimento de software iterativos, em contraposição a modelos mais antigos, como o modelo em cascata. Sobre a comparação entre os modelos iterativo e cascata, pode-se afirmar que:

Um modelo iterativo permite entregas periódicas, antecipando problemas de integração logo no início do desenvolvimento, enquanto um modelo cascata adiará as questões que seriam levantadas num período de integração.

O modelo cascata é muito utilizado em conjunto com metodologias ágeis de desenvolvimento, por permitir um feedback constante do cliente, em contraponto ao modelo iterativo, que permite um contato maior com o cliente somente nos períodos de teste.

A principal característica do modelo cascata é a repetição (daí o nome cascata) de processos como análise, desenvolvimento e testes, enquanto um modelo iterativo de desenvolvimento evita alterações repetitivas no projeto.

A vantagem principal do modelo iterativo sobre o cascata é a não existência de uma etapa de análise, o que permite que o software possa ser concluído em menor tempo.

No modelo em cascata, os custos de acomodar mudanças nos requisitos do cliente é reduzido, enquanto que no modelo incremental, os gerentes precisam de entregas regulares para mensurar o progresso.

5. A respeito dos requisitos funcionais e não funcionais em engenharia de *software*, assinale a opção correta.

Um único requisito não funcional, como um requisito de proteção, pode gerar uma série de requisitos funcionais relacionados que definam os serviços necessários no novo sistema.

“Cada usuário do sistema deve ser identificado apenas por sua matrícula de sete dígitos” constitui um exemplo de requisito não funcional de segurança.

“O sistema deve ser de fácil uso pelos usuários finais” configura um exemplo de requisito funcional individual.

Os requisitos não funcionais, por lidarem com desempenho, proteção ou disponibilidade, não são críticos em relação aos requisitos funcionais.

Os requisitos não funcionais, por estarem desvinculados dos serviços específicos oferecidos pelo sistema, não afetam a arquitetura geral desse sistema.

6. A gerência de requisitos compõe um conjunto de atividades que tem como objetivo ajudar a equipe de projeto a:

Identificar, controlar e rastrear requisitos e modificações de requisitos em qualquer época, à medida que o projeto prossegue.

Utilizar ferramentas de engenharia de software para modelar os requisitos do sistema, através da UML.

Construir um modelo técnico refinado de funções, características e restrições do software.

Negociar com os clientes os conflitos de prioridade de requisitos e identificar e analisar os riscos associados a cada requisito.

Avaliar os requisitos quanto à qualidade, garantindo que ambiguidades, inconsistências, omissões e erros tenham sido detectados e corrigidos.

7. São exemplos de requisitos não-funcionais:

I - Controladores experientes devem ser capazes de usar todas as funções do sistema depois de duas horas de treinamento.

II - O sistema de controle de aeronave deve ser fácil de ser usado por controladores experientes e deve estar organizado de tal maneira que os erros dos usuários sejam minimizados.

III - O sistema deve fornecer telas apropriadas para o usuário ler documentos disponíveis no repositório de documentos.

IV - O sistema deve utilizar os dados obtidos a partir dos sensores e interpretá-los para realizar a navegação.

I e II

III e IV

I, II, III e IV

N.D.A

8. Considere o seguinte cenário: “Um sistema automático de emissão de passagens vende passagens de trem. A partir de uma lista de possíveis destinos, os usuários escolhem seu destino e apresentam um cartão de crédito e um número de identificação pessoal. Os destinos possíveis devem ser organizados de modo a facilitar a escolha. Após a escolha do destino, o sistema deve responder prontamente se há espaço disponível no trem. A passagem é emitida e o custo dessa passagem é incluído em sua conta do cartão de crédito. Quando o usuário pressiona o botão para iniciar, uma tela de menu com os possíveis destinos é ativada, juntamente com uma mensagem para que o usuário selecione um destino. Uma vez selecionado um destino, pede-se que os usuários insiram seu cartão de crédito. A validade do cartão é checada e o usuário então deve fornecer um número de identificação pessoal. Quando a transação de crédito é validada, a passagem é emitida. O formato do bilhete de passagem deve seguir ao padrão definido pelo Sistema Nacional de Tráfego Ferroviário”.

São exemplos de requisitos funcionais e não funcionais, respectivamente, EXCETO:

formato do bilhete de passagem deve seguir ao padrão definido pelo Sistema Nacional de Tráfego Ferroviário; checar a validade do cartão e receber número de identificação pessoal;

emitir passagem e debitar custo no cartão de crédito; o tempo de resposta sobre vaga no trem deve ser adequado.

listar os possíveis destinos; o formato do bilhete de passagem deve seguir ao padrão definido pelo Sistema Nacional de Tráfego Ferroviário”.

verificar se existem vagas no destino escolhido; as telas devem facilitar a escolha do destino.

listar os possíveis destinos; emitir passagem e debitar custo no cartão de crédito;

9. Acerca do seguinte conceito “é o processo de transformação das ideias que estão na mente dos clientes (a entrada) em um documento formal (saída)”. O conceito é referente à fase de:

Extração (elicitação) de requisitos

Análise de requisitos

Validação de requisitos

Gerenciamento de requisitos

Todas as anteriores

10. A análise de requisitos no desenvolvimento de sistemas especifica as funcionalidades (requisitos funcionais) e as propriedades (requisitos não-funcionais). É considerado um requisito não-funcional:

Disponibilidade do sistema

Informação na interface de entrada

Notificação de inconsistência de dado na tela

Histograma na interface gráfica do usuário

Emissão de relatórios fiscais.

Comentário: Os requisitos não funcionais mapeiam os aspectos qualitativos de um software, por exemplo: performance (tempo de resposta); segurança (restrições de acesso, privilégios); perspectiva do usuário (padrão das cores, disposição dos objetivos na tela); comunicabilidade (e-mail, VoIP, Browser); usabilidade e portabilidade (a aplicação deve rodar em vários tipos de aplicativos: móveis, desktop, note). Como qualquer outro tipo de requisito, ele deve ser levantado, analisado, especificado e validado.