



Segunda Lista de Exercícios de Engenharia de Software



Data de Liberação: 27 de maio de 2008.

Prof.: Fernando Castor Filho

1. O que é refatoração? Forneça alguns exemplos de refatorações comuns.
2. Qual a principal característica das refatorações, que a distingue de mudanças mais gerais na implementação de um programa?
3. Descreva alguns (pelo menos quatro) fatores de qualidade que servem como indicativos da complexidade de um sistema de software (e, conseqüentemente, da dificuldade em mantê-lo). Não é necessário ater-se apenas ao código do sistema.
4. Do ponto de vista da manutenção de um sistema de software, o que são métricas de complexidade e de processo? Forneça alguns exemplos de tais métricas. Como essas métricas podem ajudar um desenvolvedor a estimar o quão difícil é realizar a evolução de um sistema?
5. Explique a relação entre processo de desenvolvimento de software e qualidade de software.
6. No contexto do desenvolvimento de software, qual o propósito fundamental do gerenciamento de qualidade de software? Forneça exemplos de características de um sistema que são monitoradas pelo gerenciamento de qualidade.
7. O que são métricas dinâmicas? Forneça alguns exemplos de tais métricas e dos atributos de qualidade que elas quantificam. Tais atributos são internos ou externos ao sistema-alvo da medição? E as métricas estáticas? Estão mais relacionadas com atributos internos ou externos?
8. Quais os objetivos das atividades de Verificação & Validação?
9. Imagine que você está trabalhando num projeto envolvendo a integração de uma interface Web baseada em uma tecnologia emergente, como Ajax, com um sistema pré-existente escrito em Java. Sua tarefa é garantir que o sistema nunca falhe de maneira catastrófica e sempre exiba mensagens de erro significativas para os usuários quando problemas ocorrerem. O sistema pré-existente consiste em mais de um milhão de linhas de código Java (excluindo-se comentários e linhas em branco). Exceções são a principal maneira de se notificar usuários sobre a ocorrência de problemas e a interface Web precisa saber quais exceções a parte pré-existente do sistema lança de modo a poder tratá-las de forma adequada. O sistema foi construído, porém, sem usar as chamadas “exceções checadas” de Java (exceções que são verificadas estaticamente pelo compilador de Java). Que técnica(s) você usaria para descobrir quais são as exceções que o sistema pode lançar? Sua abordagem deve ser tão econômica quanto possível em termos de custo financeiro/esforço da equipe de desenvolvimento.
10. Quais os significados dos termos “falha”, “erro” e “defeito” na terminologia de tolerância a falhas?
11. Por que detecção e recuperação de erros são as atividades mais importantes da tolerância a falhas? Por que a remoção de falhas não é considerada tão primordial?
12. Explique os conceitos de redundância e diversidade em tolerância a falhas.

13. Suponha que você precisa tornar tolerante a falhas um sistema de software responsável obter o valor fornecido por um sensor de altitude e realizar alguns cálculos a partir desse valor. O resultado desses cálculos deve ser repassado para o sub-sistema responsável por controlar a descida de um avião. Descreva, em linhas gerais, como você tornaria esse sistema tolerante a falhas, considerando o seguinte modelo de falhas:

- O sensor de altitude pode parar de funcionar abruptamente e não responder mais.
- O sensor de altitude pode, em raras ocasiões, fornecer valores espúrios e incorretos sobre a altitude do avião.
- O sistema de software responsável por ler o valor do sensor e realizar os cálculos a partir desse valor pode vir a “errar” nos cálculos, já que ele usa tanto o valor atual do sensor quanto valores anteriores obtidos a partir de outras leituras.
- O sistema responsável por controlar a descida do avião não falha.

14. Que arquitetura tolerante a falhas você usaria para construir um sistema capaz de tolerar falhas de alguns de seus componentes de hardware? Por que? Usar blocos de recuperação produziria o mesmo efeito?

15. Testes são uma técnica de verificação ou validação? Justifique sua resposta.

16. O que são inspeções? São atividades de verificação ou validação? Explique como elas são realizadas.

17. Forneça exemplos de itens que podem aparecer na *checklist* de uma inspeção.

18. Explique em linhas gerais como funciona a técnica de análise estática. Que tipo de verificações é possível fazer usando essa técnica?

19. Por que é necessário testar no nível do componente e do sistema? Forneça exemplos práticos da necessidade de cada um dos dois tipos de teste.

20. Por que os testes de *release*, ou de aceitação, são testes caixa-preta? Por que não faz sentido usar uma abordagem caixa branca?

21. Explique os conceitos de teste de desempenho, de carga e de estresse. Quais deles você usaria para testar um sistema Web com requisitos estritos de disponibilidade e escalabilidade?

22. Descreva todas as partições que você conseguir encontrar para testar um procedimento que recebe como entrada um *array* de números inteiros e fornece como saída um *array* onde os elementos do *array* de entrada estão posicionados em ordem crescente. Por exemplo, se a entrada do programa for o *array* $A = \{5, 8, 3, 6, 1, 15\}$, sua saída será $B = \{1, 3, 5, 6, 8, 15\}$. Note que o *array* de saída é diferente do de entrada e este último permanece inalterado.

23. Por que sistemas de software orientados a objetos devem ter baixo acoplamento entre seus módulos e alta coesão?

24. O que é uma “decisão de projeto”? Por que é útil documentar tais decisões durante o desenvolvimento de um sistema de software?

25. Na documentação de um padrão de projeto, o que significam os campos “Problema” e “Usos Conhecidos”.

26. Em quais situações é benéfico utilizar o padrão Observer? Justifique sua resposta. Você acha que ele seria útil, por exemplo, para notificar um sistema responsável por gravar em disco o estado atual de uma árvore binária de busca de que essa estrutura de dados mudou? Novamente, justifique sua resposta.

27. Por que, no projeto de software, é recomendável que classes e pacotes dependam, tanto quanto possível, “na direção da estabilidade”?