

Ciclo de vida:

1.1) Explique quais são as etapas do ciclo de vida de um software segundo rezende em seu livro engenharia de software e sistemas de informação. Faça uma breve correlação entre as etapas, destacando artefatos que podem ser gerados em cada uma.

1.2) Explique os tipos de correções que podem existir em um sistema de software segundo rezende em seu livro engenharia de software e sistemas de informação.

1.3) Conceitue erro, defeito e falha, e apresente a diferença entre os termos.

Processos de Software:

- 2.1** Justificando sua resposta com base no tipo de sistema a ser desenvolvido, sugira o modelo genérico de processo de software mais adequado para ser usado como base para a gerência do desenvolvimento dos sistemas a seguir:

Um sistema para controlar o antibloqueio de frenagem de um carro.

Um sistema de realidade virtual para dar apoio à manutenção de software.

Um sistema de contabilidade para uma universidade, que substitua um sistema já existente.

Um sistema interativo de planejamento de viagens que ajude os usuários a planejar viagens com menor impacto ambiental.

- 2.2** Explique por que o desenvolvimento incremental é o método mais eficaz para o desenvolvimento de sistemas de software de negócios. Por que esse modelo é menos adequado para a engenharia de sistemas de tempo real?

- 2.3** Considere o modelo de processo baseado em reuso da Figura 2.3. Explique por que, nesse processo, é essencial ter duas atividades distintas de engenharia de requisitos.

- 2.4** Sugira por que é importante, no processo de engenharia de requisitos, fazer uma distinção entre desenvolvimento dos requisitos do usuário e desenvolvimento de requisitos de sistema.

2.5 Explique os motivos para desenvolver um software no modelo cascata.

métodos ágeis:

- 3.1** Explique por que, para as empresas, a entrega rápida e implantação de novos sistemas frequentemente é mais importante do que a funcionalidade detalhada desses sistemas.
- 3.2** Explique como os princípios básicos dos métodos ágeis levam ao desenvolvimento e implantação de software acelerados.
- 3.3** Quando você não recomendaria o uso de um método ágil para o desenvolvimento de um sistema de software?
- 3.4** Extreme Programming expressa os requisitos dos usuários como histórias, com cada história escrita em um cartão. Discuta as vantagens e desvantagens dessa abordagem para a descrição de requisitos.
- 3.5** Explique por que o desenvolvimento *test-first* ajuda o programador a desenvolver um melhor entendimento dos requisitos do sistema. Quais são as potenciais dificuldades com o desenvolvimento *test-first*?
- 3.6** Sugira quatro razões pelas quais a taxa de produtividade de programadores que trabalham em pares pode ser mais que a metade da taxa de produtividade de dois programadores que trabalham individualmente.
- 3.7** Compare e contraste a abordagem Scrum para o gerenciamento de projetos com abordagens convencionais dirigida a planos, como discutido no Capítulo 23. As comparações devem ser baseadas na eficácia de cada abordagem para o planejamento da alocação das pessoas nos projetos, estimativa de custos de projetos, manutenção da coesão da equipe e gerenciamento de mudanças no quadro da equipe do projeto.

- 3.8** Você é um gerente de software em uma empresa que desenvolve softwares críticos de controles para aeronaves. Você é responsável pelo desenvolvimento de um sistema de apoio ao projeto de software que dá suporte para a tradução de requisitos de software em uma especificação formal de software (discutido no Capítulo 13). Comente sobre as vantagens e desvantagens das estratégias de desenvolvimento a seguir:
- a) Coletar dos engenheiros de software e *stakeholders* externos (como a autoridade regulatória de certificação) os requisitos para um sistema desse tipo e desenvolver o sistema usando uma abordagem dirigida a planos.
 - b) Desenvolver um protótipo usando uma linguagem de *script*, como Ruby ou Python, avaliar esse protótipo com os engenheiros de software e outros *stakeholders* e, em seguida, revisar os requisitos do sistema. Desenvolver novamente o sistema final, usando Java.
 - c) Desenvolver o sistema em Java, usando uma abordagem ágil com um usuário envolvido na equipe de desenvolvimento.
- 3.9** Tem-se sugerido que um dos problemas de se ter um usuário participando de uma equipe de desenvolvimento de software é que eles 'se tornam nativos', ou seja, adotam a perspectiva da equipe de desenvolvimento e perdem de vista as necessidades de seus colegas usuários. Sugira três maneiras de evitar esse problema e discuta as vantagens e desvantagens de cada abordagem.

3.10 Como XP preconiza que devem ser os contratos de desenvolvimento de software?

3.11 Quais as diferenças entre XP e Scrum?

3.12 Times Scrum são ditos cross-funcionais e auto-organizáveis. Por quê? Defina esses termos.

3.13 Em Scrum, qual a diferença entre as histórias do topo e do fundo do Backlog do Produto?

3.14 O que são e para que servem story points?

3.15 Em Scrum, qual a diferença entre uma sprint review e uma retrospectiva?

3.16 Quais são as principais diferenças entre Scrum e Kanban?