

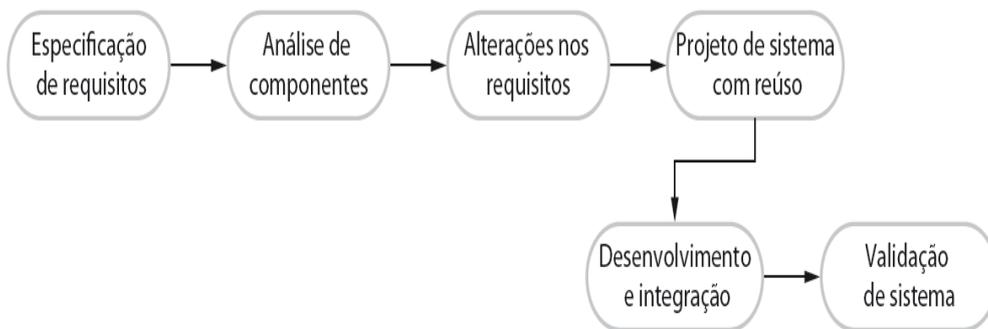
Exercícios de fixação extraídos do livro **Sommerville, Ian. Engenharia de Software. 9ª ed.** Recomenda-se a leitura antecipada dos capítulos indicados.

Capítulo 1 – Introdução à Engenharia de Software

1. Explique por que software profissional não é apenas os programas que são desenvolvidos para o cliente.
2. Qual a diferença mais importante entre o desenvolvimento de um produto genérico de software e o desenvolvimento de software sob demanda? O que isso pode significar na prática para usuários de produtos de software genérico?
3. Quais são os quatro atributos importantes que todo software profissional deve possuir? Sugira outros quatro atributos que, as vezes, podem ser significantes.
4. Além dos desafios de heterogeneidade, mudanças sociais e corporativas, confiança e proteção, identifique outros problemas e desafios que a engenharia de software provavelmente enfrentara no século XXI (Dica: pense no meio ambiente).
5. Baseado em seu conhecimento de alguns tipos de aplicações discutidos na Seção 1.1.2 (Sommerville, Ian. Engenharia de Software. 9ª ed.), explique, com exemplos, por que tipos de aplicações diferentes requerem técnicas especializadas de engenharia de software para apoiar seu projeto e desenvolvimento.
6. Explique por que existem ideias fundamentais na engenharia de software que se aplicam a todos os tipos de sistemas.
7. Explique como uso universal da Internet mudou os sistemas de software.
8. Discuta se os engenheiros profissionais devem ser certificados da mesma forma que médicos e advogados.
10. Para ajudar a combater o terrorismo, muitos países estão planejando desenvolver, ou já desenvolveram, sistemas computacionais que rastreiam grandes números de cidadãos e suas ações. Obviamente, isso tem implicações nas questões da privacidade. Discuta a ética de se trabalhar desenvolvendo esse tipo de sistema.

Capítulo 2 – Processos de Software

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. Explique por que o desenvolvimento incremental e 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?
3. Considere o modelo de processo baseado em reuso mostrado na figura abaixo. Explique por que, nesse processo, é essencial ter duas atividades distintas de engenharia de requisitos.



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.
5. Descreva as principais atividades do processo de projeto de software e as saídas dessas atividades. Usando um diagrama, mostre as possíveis relações entre as saídas dessas atividades.
6. Explique por que, em sistemas complexos, as mudanças são inevitáveis. Exemplifique as atividades de processo de software que ajudam a prever as mudanças e fazer com que o software seja desenvolvido mais tolerante a mudanças (desconsidere prototipação e entrega incremental).
7. Explique por que os sistemas desenvolvidos como protótipos normalmente não devem ser usados como sistemas de produção.
8. Explique por que o modelo em espiral de Boehm e um modelo adaptável, que apoia tanto as atividades de prevenção de mudanças quanta as de tolerância a mudanças. Na prática, esse modelo não tem sido amplamente usado. Sugira as possíveis razões para isso.
9. Quais são as vantagens de proporcionar visões estáticas e dinâmicas do processo de software, assim como no Rational Unified Process?
10. Historicamente, a introdução de tecnologia provocou mudanças profundas no mercado de trabalho e, pelo menos temporariamente, deixou muitas pessoas desempregadas. Discuta se a introdução da automação extensiva em processos pode vir a ter as mesmas consequências para os engenheiros de software. Se sua resposta for não, justifique. Se você acha que sim, que vai reduzir as oportunidades de emprego, é ética a resistência passiva ou ativa, pelos engenheiros afetados, à introdução dessa tecnologia?

Capítulo 3 – Desenvolvimento Ágil de Software

1. Explique por que, para as empresas, a entrega rápida e implantação de novos sistemas frequentemente mais importante do que a funcionalidade detalhada desses sistemas.
2. Explique como os princípios básicos dos métodos ágeis levam ao desenvolvimento e implantação de software acelerados.
3. Quando você não recomendaria o uso de um método ágil para o desenvolvimento de um sistema de software?
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.
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*?
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.
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.
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 suporte para a tradução de requisitos de software em uma especificação formal de software (discutido no Capítulo 13 do livro Sommerville, Ian. Engenharia de Software. 9ª ed.). 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.
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.
10. Para reduzir os custos e o impacto ambiental das viagens, sua empresa decide fechar uma série de escritórios e dar suporte ao pessoal para trabalhar em casa. No entanto, a gerência sênior que introduz essa política não está ciente de que o software é desenvolvido por métodos ágeis, que contam com equipe trabalhando no mesmo local, e a programação em pares. Discuta as dificuldades que essa nova política pode causar e como você poderia contornar esses problemas.