

## **ENGENHARIA DE SOFTWARE – ATIVIDADE TEÓRICA 2**

**2.1)** Justifique 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 como 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 reúso 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 o desenvolvimento de requisitos de sistema.

**2.5)** Descreva as principais atividades do processo de projeto de software e as saídas dessas atividades. Usando um diagrama que mostre as possíveis relações entre as saídas dessas atividades.

**2.6)** 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 (desconsiderando prototipação e entrega incremental).

**2.7)** Explique por que os sistemas desenvolvidos como protótipos normalmente não devem ser usados como sistemas de produção.

**2.8)** Explique por que o modelo em espiral de Boehm é um modelo adaptável, que apoia tanto as atividades de prevenção de mudanças quanto as de tolerância a mudanças quanto as de tolerância a mudanças. Na prática, esse modelo não tem sido amplamente usado. Sugira as possíveis razões para isso.

**2.9)** Quais são as vantagens de proporcionar visões estáticas e dinâmicas do processo de software, assim como no Rational Unified Process?

2.10) 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?

## RESPOSTAS

**2.1)**

- **Sistema de antibloqueio de frenagem (ABS):** Devido à necessidade de alta confiabilidade, segurança e testes rigorosos, o modelo mais adequado é o modelo cascata. Ele permite um processo sistemático com validações em cada etapa, essencial em sistemas embarcados de tempo real.
- **Sistema de realidade virtual para manutenção de software:** Devido à natureza experimental e interativa, o modelo de prototipação é o mais apropriado, pois permite refinar funcionalidades com base em testes práticos com os usuários.
- **Sistema de contabilidade para universidade:** Como se trata de uma substituição de sistema, onde é necessária compatibilidade com processos existentes, o modelo incremental permite integrar módulos gradualmente, com menor risco.
- **Sistema de planejamento de viagens com foco ambiental:** Pela interatividade, inovação e ajustes com base na experiência do usuário, o modelo incremental com prototipagem permite evoluir o sistema com feedback contínuo.

**2.2)** O desenvolvimento incremental é eficaz em sistemas de negócios por permitir entregas rápidas e parciais, facilitando a adaptação às mudanças dos requisitos e o feedback frequente dos usuários. Em sistemas de tempo real, como controle de processos industriais, essa abordagem é menos adequada porque esses sistemas exigem integração completa e testes rigorosos desde o início para garantir confiabilidade e segurança.

**2.3)** No processo baseado em reuso, há duas atividades distintas de engenharia de requisitos porque é necessário, primeiro, entender os requisitos gerais do sistema e, depois, analisar os requisitos com base nos componentes reutilizáveis. Isso garante que o sistema seja adaptado ao que está disponível para reuso e evita retrabalho.

**2.4)** Os **requisitos do usuário** descrevem o que os usuários esperam do sistema, geralmente em linguagem natural. Já os **requisitos do sistema** são técnicos, detalhados e guiam os desenvolvedores. Separar esses níveis evita confusão, facilita a comunicação com os clientes e permite uma implementação mais precisa.

**2.5)** As principais atividades são traçadas como:

- **Projeto de Arquitetura:** Define os principais componentes e suas interações.
- **Projeto de Interface:** Especifique como os componentes e os usuários interagem.
- **Projeto de Componentes:** Detalhe a implementação interna dos módulos.
- **Projeto de Banco de Dados:** Estruture a persistência de dados.

No entanto, as saídas são designadas da seguinte forma:

- **Projeto de Arquitetura → Projeto de Interface ↔ Projeto de Componentes  
↔ Projeto de Banco de Dados**

**2.6)** Em sistemas complexos, mudanças ocorrem por alterações de mercado, políticas ou tecnológicas. Atividades como **engenharia de requisitos com validação constante**, **documentação modularizada**, e **teste baseado em regressão** ajudam a lidar com essas mudanças.

**2.7)** Protótipos são criados para explorar ideias, não para desempenho ou segurança. Usá-los como sistemas finais pode comprometer a manutenção, estabilidade e escalabilidade.

**2.8)** O modelo em espiral é adaptável pois integra planejamento de riscos, validação contínua e revisões em cada ciclo. Na prática, não é muito usado por ser complexo, exigir maior especialização em gerenciamento de riscos e demandar mais tempo e custo.

**2.9)** Ter visões estáticas (ex: documentos, estrutura) e dinâmicas (ex: fluxos, interações) ajuda a entender o sistema por diferentes perspectivas. O RUP usa isso para melhorar a comunicação entre equipes e garantir consistência entre as fases do projeto.

**2.10)** A automação pode eliminar tarefas repetitivas, mas também cria novas funções mais criativas e analíticas. Não é ético resistir à tecnologia se ela trouxer melhorias sociais. A solução é se requalificar, adaptar-se e participar do desenvolvimento dessas novas tecnologias.