



Instituto Federal de Educação, Ciência e Tecnologia da Bahia - IFBa
Grupo de Sistemas Distribuídos, Otimização, Redes e Tempo-Real - GSORT
Especialização em Computação Distribuída e Ubíqua
INF612 - Aspectos Avançados em Engenharia de Software
Prof.: Sandro Santos Andrade

LISTA DE EXERCÍCIOS

Parte I: Arquitetura de Software

- 01) Desenvolva três arquiteturas alternativas para um mesmo problema (por exemplo, um jogo simples tipo Tetrix ou uma agenda). O que torna estas arquiteturas diferentes ? Como você chegou a estas arquiteturas ?
- 02) Escolha um sistema utilizado no seu trabalho ou apresentado nas aulas e identifique os componentes, conectores e topologia que formam a sua arquitetura. Qual o grau de acoplamento entre componentes e conectores no sistema implementado (arquitetura descritiva) ?
- 03) Selecione uma aplicação distribuída de sua escolha e informe como você transformaria as diretrizes arquiteturais do estilo *Client-Server* em decisões específicas de projeto. Essas diretrizes foram fáceis de aplicar ? Elas são facilmente violadas ? Dê um exemplo de violação de uma das diretrizes e analise o seu impacto.
- 04) Suponha um sistema que tem estado em uso por 20 anos em uma empresa. O sistema tem sido modificado e ampliado, durante este período, de modo a acomodar um fluxo contínuo de novas demandas. Os engenheiros de *software* responsáveis pela aplicação, entretanto, estão apreensivos sobre a realização de mais modificações no sistema. Quais os critérios para que a equipe decida, com segurança, se deve continuar modificando o sistema original ou construir uma nova aplicação ?
- 05) Descreva a arquitetura de um compilador em relação aos estilos arquiteturais adotados. Um único estilo arquitetural é utilizado ou um conjunto de estilos são criteriosamente aplicados ?
- 06) Escolha um aplicação que você seja familiar e que utilize primariamente conectores do tipo *Procedure Call*. Substitua todos os conectores *Procedure Call* por conectores explícitos e unicamente identificados, de modo a separar completamente o *concern* de processamento e armazenamento de dados do *concern* de interação entre componentes. Discuta os méritos relativos das arquiteturas destas duas aplicações (antes e depois da substituição).
- 07) Repita o exercício acima substituindo um ou mais conectores que suportam interação síncrona por conectores que suportam interação assíncrona. Quais desafios você encontrou para realizar isso ? A aplicação modificada possui o mesmo comportamento da aplicação original ? Porque ?

Parte II: Padrões de Projeto e Idiomas de Programação

- 08) Apresente um exemplo que viole o Princípio do Aberto-Fechado. Como você modificaria esta aplicação para que este princípio fosse plenamente atendido ?
- 09) Apresente um exemplo (não apresentado em sala) que viole o Princípio de Substituição de Barbara Liskov. Como você modificaria esta aplicação para que este princípio fosse plenamente atendido ?
- 10) Faça uma análise comparativa, sob o ponto de vista de projeto (e não de implementação), entre herança de implementação, herança de interface e agregação. Analise em relação aos seguintes critérios: flexibilidade, custo e risco.

11) Apresente um exemplo de como os Idiomas de Programação podem melhorar a robustez de um código-fonte.

Parte III: Visualização de Software

12) Cite um exemplo prático onde a utilização de técnicas de Visualização de Software facilitaria as atividades de evolução da aplicação. No exemplo citado, que tipo de visualização (estrutural, comportamental ou de evolução) estaria sendo utilizada ? Qual paradigma de visualização você poderia propor para realizar esta tarefa ?

13) A Visualização de Evolução nos permite obter informações interessantes sobre o processo de desenvolvimento de *software* sendo utilizado. Proponha critérios para, através da Visualização de Evolução, diferenciar um Processo Ágil de um Processo Clássico (por exemplo, em cascata).

Parte IV: Gerência de Qualidade e Integração Contínua

14) Escolha e justifique três atividades/padronizações por você julgados como prioritários na implantação de um Processo de Gerência de Qualidade. Como isto seria implantado no dia-a-dia organizacional ?

15) Explique onde as métricas de produtos se inserem dentro de um Processo de Gerência de Qualidade. Cite exemplos.

16) A sua empresa adota algum princípio de Desenvolvimento Ágil ? Se sim, escolha três princípios e indique como eles são implantados. Se não, proponha três princípios mais urgentes a serem utilizados com o objetivo de melhorar a gerência da qualidade dos sistemas sendo desenvolvidos.

17) A sua empresa adota alguma ferramenta de Integração Contínua ? Se sim, indique as atividades, realizadas no *build script*, que mais contribuem para a Gerência de Qualidade. Se não, indique as principais atividades, a serem realizadas no *build script*, com o objetivo de melhorar a Gerência da Qualidade dos sistemas sendo desenvolvidos.

Parte V: Engenharia de Software Experimental

18) Qual a importância da utilização de Intervalos de Confiança durante as suas medições ? Dê um exemplo de como isto pode ser aplicado em experimentos com *software*.

19) Porque pode-se chegar nestes Intervalos de Confiança a partir de somente uma amostra da população (por exemplo, realizando medições em apenas um *software* de um domínio de aplicação) ?

20) Qual a importância da utilização de (Quasi-)Experimentos na Engenharia de Software ? Quais critérios devem ser verificados para decidir sobre a viabilidade de execução de tais experimentos ?

21) Cite um exemplo, aplicável no seu local de trabalho ou em um cenário de pesquisa acadêmica, onde a realização de um (Quasi-)Experimento é justificada ? Explique quais seriam os objetivos deste experimento.