

CS.1 Quais são os 5 Fundamentos de CS?

R: 1. Minimizar complexidade, 2. Antecipar mudanças, 3. Construção para verificação, 4. Reuso e 5. Padrões na Construção.

CS.2 O que é refatoração?

R: Otimização do sistema através de uma melhoria da estrutura interna do código sem modificar seu comportamento externo. Quebra o código para diminuir a complexidade e torna-lo mais compreensível, trazendo muitas vantagens. Um exemplo de uma técnica de refatoração é a Extract Method, que se trata de decompor uma rotina grande em métodos com códigos menores.

CS.3 Considerações Práticas

1. O que é um Framework?

R: É um conjunto de códigos comuns a vários sistemas e reutilizáveis que podem ser estendidos para disponibilizar uma porção de uma funcionalidade pronta e testada sem ter a necessidade de criá-la do zero, sendo muito útil para tarefas repetitivas.

CS.4 Qual ou quais dos 5 fundamentos de CS é ou são os mais importantes? Desenvolva o porquê.

R: Todos são claramente essenciais para a construção de um software, mas considero que sejam os fundamentos de “padrões na construção” e “antecipar mudanças”. Os motivos por ter escolhido esses dois fundamentos é que considero necessário seguir padrões e especificações para dar validade ao sistema e a visão estratégica é muito importante para a nossa área pois um software também tem que ser realizado pensando em todo seu ciclo de vida para ser utilizado por muito tempo, mesmo com as mudanças rápidas.

CS.5 Como você diferenciaria complicado de complexo? Dê um exemplo.

R: Algo que é complicado possui uma difícil compreensão sobre seu funcionamento, enquanto algo complexo possui um comportamento imprevisível.

Um exemplo de complicado seria a ação de aprender a dirigir um carro e algo complexo seria o fluxo de vários carros em uma cidade, pois mudam a cada momento.

CS.6 Testes automatizados auxiliam em quais fundamentos de CS?

[X] Construção para Verificação

CS.7 O que é reuso? Como podemos alcançar reuso na construção de software?

R: É utilizar um recurso já existente para resolver um problema diferente. Pode ser alcançada através dos seguintes recursos:

- Grupos de classes – Bibliotecas (externas ao projeto): coleção de software controlada.

Módulos: interno ao projeto

- Código Fonte
- Commercial Off-the-Self (COTS) – bibliotecas compradas para serem utilizadas no projeto.

CS.8 O que é TDD? Você concorda ou discorda sobre TDD?

R: Test Driven Programming, uma prática de desenvolvimento guiada para testes que influencia na criação do próprio software, pois os testes já são construídos antes mesmo do software em si e eles são utilizados para orientar a sua implementação. Acredito ser muito útil pois previne (não evita totalmente) de serem criadas coisas desnecessárias, consequente perdendo tempo no processo de construção do software.