Questionário sobre Construção de Software

Curso de Análise e Desenvolvimento de Sistemas – Engenharia de Software

Faculdades Integradas Camões – Curitiba, PR– Brazil

Fernando Kiala Marques

Definição de Construção de Software (CS)
É a referência aos detalhes sobre a criação de um software funcional através da combinação de programação, testes unitários, testes de integração e debugging.

CS.1 Quais são os 5 Fundamentos de CS?
1. Minimizar Complexidade 2. Antecipar Mudanças 3. Construção para Verificação 4. Reuso 5. Padrões na Construção

CS.2 O que é refatoração?
Refatoração (do inglês Refactoring) é o processo de modificar um sistema de software para melhorar a estrutura interna do código sem alterar seu comportamento externo.

CS.3 Considerações Práticas 1. O que é um Framework?
Um framework em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica.

CS.4 Qual ou Quais dos 5 fundamentos de CS é ou são mais importantes? Desenvolva o porque.
Acredito que com exceção do Reuso, que seria uma situação opcional que pode ou não ajudar no processo, todos os demais fundamentos são de suma importância para a construção de software. Pois cada um deles acabam contribuindo muito para que se alcance o objetivo de desenvolver um software de qualidade.

- O fundamento de **Minimizar Complexidade** deve ser levado em conta devido a situação de quanto mais complexo um software, mais difícil será o desenvolvimento do projeto, o que conseqüentemente vai demandar de mais tempo, recursos e ao final pode se tornar complicado para os usuários finais utilizarem, não sendo um software de fácil aprendizagem.
- O Fundamento de **Antecipar Mudanças** ajuda muito a pensar no futuro para evitar problemas e possíveis mudanças que podem afetar o software em partes ou ao todo, gerando retrabalho. E como o software modifica-se com o tempo a antecipação da mudança deve guiar muitas a decisões da construção de software.
- O fundamento de **Construção para Verificação** é uma forma de identificar falhas e corrigi-las antes de o software ser entregue e ocorrer essas falhas em produção.

- O fundamento de **Padrões na Construção** ajuda muito a ter uma equipe alinhada que vai trabalhar de uma forma similar, evitando que cada um desenvolva seu trabalho da forma que achar melhor. Além disso muitos padrões ajudam no desenvolvimento de softwares flexíveis, reutilizáveis e de fácil manutenção.

Em geral acredito que os quatro fundamentos citados contribuem para o desenvolvimento de um software de qualidade, que atenda as necessidades e tenha uma boa aceitação do cliente, evitando problemas como novos gastos e retrabalhos.

CS.5 Como você diferenciaria complicado de complexo? De um exemplo.
O complicado é uma situação ou algo confuso, difícil de se compreender ou se apreender. Já o complexo seria uma construção composta de numerosos elementos interligados ou que funcionam como um todo.
Um exemplo de complicado voltado para Software seria uma situação de um projeto de sistema contábil que envolve cálculos com fórmulas pequenas, mas difíceis de se compreender e aprender. Já um exemplo de complexo seria um sistema que deve atender uma ISO que possui vários critérios e padrões que devem ser rigorosamente atendidos.

CS.6 Testes Automatizados auxiliam em quais fundamentos de CS?
[ ] Minimizar Complexidade [ ] Antecipar Mudanças
[x] Construção para Verificação [ ] Reuso [ ]Padrões na Construção

CS.7 O que é reuso? Como podemos alcançar reuso na construção de software?
O reuso é utilizar um recurso existente para solucionar um problema diferente. Podemos utilizar na construção de software alguns recursos existentes como bibliotecas, códigos fontes ou módulos.

CS.8 O que é TDD? Você concorda ou discorda sobre TDD?
Test **D**riven **D**evelopment (TDD) ou em português Desenvolvimento guiado por testes é uma técnica de desenvolvimento de software que se relaciona com o conceito de verificação e validação e se baseia em um ciclo curto de repetições.
Concordo que o TDD pode ser uma ótima técnica para ser aplicada no desenvolvimento das funcionalidades do software. Acredito que funcionaria muito bem em projetos que usam metodologias ágeis como o Scrum, que divide o desenvolvimento do software em várias partes atribuindo elas como uma tarefa ao desenvolvedor , que poderia aplicar a técnica do TDD em conjunto que o ajudaria a chegar em um resultado melhor ao final de cada tarefa, contribuindo com um código mais limpo, identificando e corrigindo erros antes de chegar a uma fase de testes.