

Os 7 Princípios Básicos de Desenvolvimento de Software (David Hooker*)

*<http://wiki.c2.com/?SevenPrinciplesOfSoftwareDevelopment>

Primeiro Princípio - A razão pelo qual tudo existe



- Tem que existir uma razão para se fazer software!!
 - Senão melhor não fazer
 - Fazer software -> “agregar valor para o usuário”
 - É importante enxergar os reais requisitos do software!

Segundo Princípio - Mantenha as coisas simples (KISS - Keep It Simple ! Stupid)

- “um projeto deve ser o mais simples possível, mas não mais simples que isso”
- As soluções mais elegantes normalmente são simples
- Fazer algo simples usualmente demanda mais tempo do que fazer de forma complexa



Terceiro Princípio - Mantenha o estilo



- O projeto de um software deve seguir um único estilo (estilo de codificação, documentação, teste, um mesmo processo, ...)
- A combinação de diferentes estilos corretos pode levar a um software incorreto
- Padrões e estilos devem ser estabelecidos no início e seguidos por todos

Quarto Princípio - O que você produzir alguém vai consumir



- Sempre especifique, projete e codifique algo pensando que outros vão ler
- Sempre exija qualidade nos produtos que você consome e forneça qualidade nos produtos que você produz
- Alguém pode ter que depurar o código que você escreve, e isso faz dele um usuário de seu código.

Quinto Princípio - Esteja aberto para o futuro

- Sistemas de boa qualidade têm vida longa
- Projete desde o início pensando na manutenção



Sexto Princípio - Planeje com antecedência para Reutilização

- Pense no problema geral, e não só no problema específico
- Busque por soluções já existentes



Sétimo Princípio - Pense !

“Plano é desnecessário, mas planejar é indispensável” D. Eisenhower – General do exército americano

- Avalie alternativas
- Detalhe os riscos

