



Eixo Tecnológico Informação e Comunicação

**Princípios da Engenharia
do Software**
Rogério Antonio Schmitt



PRINCÍPIOS DA ENGENHARIA DO SOFTWARE

A engenharia de software classicamente apresenta um conjunto de princípios a serem utilizados ao realizar um projeto de desenvolvimento de software. Os princípios ajudam a se pensar numa prática segura de trabalho nesse tipo de projeto. Isso explica a sua importância.

Esses princípios servem como boas práticas ou lições aprendidas sobre como desenvolver software. Não se tratam de regras, mas uma filosofia de como desenvolver. Dentre os princípios, podemos citar alguns de grande relevância:

Decomposição:

- ▶ Como tratar a complexidade inerente a sistemas de software?

“A decomposição funcional é uma maneira de conceber o software como um conjunto de funções de alto nível (requisitos) que são decompostas em partes cada vez mais simples até chegar a comandos individuais de uma linguagem de programação.”
(Wazlawick, 2013)



**INSTITUTO
FEDERAL**
Santa Catarina

Abstração

- ▶ Descrever um elemento em uma linguagem de nível mais alto do que o necessário para sua construção.
- ▶ Ou seja: simplificar! Muitas vezes deixando escapar alguns detalhes propositalmente.
- ▶ É um auxílio importante para que todos os interessados no desenvolvimento possam entender estruturas grandes e complexas.
- ▶ Usar descrições mais abstratas e menos detalhadas.

Generalização

- ▶ Agrupar conceitos com atributos comuns em classes de objetos.
- ▶ A Orientação a Objetos é fruto da ideia de generalização.
- ▶ Permite a reutilização de definições em itens de software, classificando dados.

Padronização

- ▶ A padronização auxilia na elaboração de produtos com qualidade mais previsível.
- ▶ Padrões permitem capitalizar experiências de outros projetos.
- ▶ Erros já experimentados e que já possuem solução conhecida.

Rastreabilidade

- ▶ Software, por ser um elemento lógico, consiste de instruções, estruturas de dados e documentação.
- ▶ Software é baseado em uma série de documentos e especificações.
- ▶ Muitos deles derivam uns dos outros (decomposição).
- ▶ É necessário manter um registro dos rastros de um artefato em outro.

Por exemplo: Se um requisito muda, há um impacto em vários artefatos, como diagrama de casos de uso, diagrama de classes, de atividades, código fonte, casos de teste...

Gerenciamento de Requisitos

- ▶ É possível identificar todos os requisitos de um sistema desde o início do projeto?
- ▶ Requisitos costumam mudar com muita frequência.
- ▶ É necessário gerenciar suas mudanças/ sua evolução.
- ▶ Faz parte do processo de desenvolvimento e evolução do software.

Desenvolvimento iterativo

- ▶ É possível desenvolver software em um único ciclo com início, meio e fim?
- ▶ Atualmente trabalha-se com o entendimento de processo de desenvolvimento iterativo:
- ▶ Vários ciclos de desenvolvimento são realizados.
- ▶ Cada ciclo visa atender a um conjunto de objetivos (decomposição).
- ▶ Cada ciclo contribui para a geração e o amadurecimento do produto final.

Arquitetura baseada em Componentes

- ▶ A utilização de componentes de software é uma das formas de obter reusabilidade.
- ▶ Facilita o tratamento da complexidade do software.
- ▶ Torna mais ágil a manutenção e evolução de pequenas partes do produto.

Gerenciamento de Mudanças

- ▶ É imprescindível manter o controle da evolução e alterações em produtos de software.
- ▶ Há situações nas quais é necessário voltar atrás.
- ▶ Há situações nas quais é necessário desenvolver duas versões de um mesmo componente paralelamente.
- ▶ Um bom sistema de gerenciamento de configuração e mudança permite efetividade nessa tarefa.



Referências

Wazlawick, Raul. Engenharia do Software conceitos e práticas.
Florianópolis: Elsevier 1ª ed, 2013.