Projeto de Desenvolvimento de Software

Princípios da Engenharia de Software

Msc. Eliezio Soares
eliezio.soares@ifrn.edu.br
http://docente.ifrn.edu.br/elieziosoares

NBR ISO 9000-3
Definições:

- A ISO 9000 define critérios de garantia de qualidade.
- A ISO 9000-3 fornece uma interpretação da ISO 9000 para aplicação em empresas de desenvolvimento de software.

  - **Software:** Criação intelectual compreendendo os programas, procedimentos, regras e qualquer documentação correlata à operação de um sistema de processamento de dados.
  - **Produto de Software:** Conjunto completo de programas de computador, procedimentos e documentação correlata, assim como dados designados para entrega a um usuário.
  - **Item de Software:** Qualquer parte identificável de um produto de software em

Engenharia de Software
PRINCÍPIOS

- Recebemos como herança e missão propagar boas práticas no desenvolvimento de novos projetos.
- Lições aprendidas ao longo do tempo sobre COMO desenvolver software.
- Não se trata de simples regras, mas de filosofia / princípios.

Decomposição

"Divide et Impera" – "Dividir para Reinar"

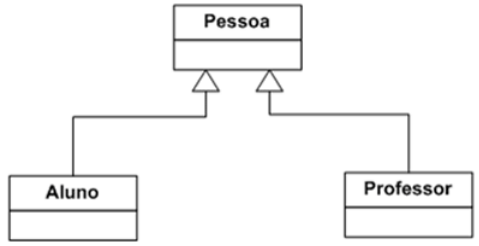
- Como tratar a complexidade inerente a sistemas de software ?
- "A decomposição funcional é uma maneira de conceber o software como um conjunto de funções de alto nível (requisitos) que são decompostas em partes cada vez mais simples até chegar a comandos individuais de uma linguagem de programação." (Wazlawick, Raul Sidnei. 2013)
  - A decomposição algorítmica (a solução estruturada)
  - A decomposição Orientada a objetos (OO)

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.
  - Descrições mais abstratas:
    - Algoritmos -> Classes
    - Classes -> Diagrama de Classes
    - Diagramas -> Descrição Textual

Generalização

- Agrupar conceitos com atributos comuns.
- Permite a reutilização de definições em itens de software.
- A Orientação a Objetos é fruto da idéia de generalização:
  - Professores e Alunos têm em comum o fato de serem Pessoas, eles possuem atributos comuns.



Herança: Aluno e Professor herdam de Pessoa

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 é um elemento de sistema lógico, e não físico. Consistem de Instruções, estruturas de dados e documentação.” (Pressman, 2000)
  - 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...

Gerenciamento de Requisitos

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

Desenvolvimento Interativo

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

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.

Gerenciamento de Riscos

- Sempre existem riscos em um projeto.
  - Eles precisam ser previstos.
  - É necessário prevenir-se.
  - É preciso definir plano de ação.
- Exemplos:
  - Equipe com alta rotatividade.
  - Equipe inexperiente.
  - Pressão pela redução de prazos.
  - Exceder limite orçamentário.

Referências

- Wazlawick, Raul Sidnei. Engenharia de software: conceitos e práticas. Rio de Janeiro: Elsevier, 2013.