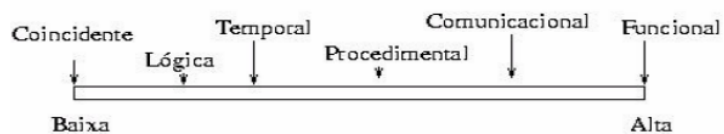


Design de Software

Introdução ao Design de Software

- O projeto de um sistema de software tem muitas características em comum com projetos de engenharia, do ponto de vista do mapeamento dos requisitos funcionais em soluções tecnológicas.
- Projeto é a única maneira com a qual se consegue traduzir com precisão os requisitos do cliente em um produto de software ou sistema finalizado.
- Nos modelos de ciclo de vida, a atividade de projeto recebe o resultado da atividade de requisitos e gera um modelo para a atividade de implementação ou codificação.
- Durante o projeto de software é criada a estrutura interna de um sistema de software. Esta estrutura interna é chamada de arquitetura.
- Existem diretrizes para a qualidade de um projeto de software, e entre elas estão:
 - Um projeto deve ser modular
 - Um projeto deve conter representações distintas para dados, arquitetura, interfaces e componentes
 - Um projeto deve levar a componentes que possuam características de independência funcional
 - Um projeto deve levar a interfaces que reduzam a complexidade das conexões entre os componentes e o ambiente externo.
- Projeto Geral ou Preliminar: fase que traduz a especificação do sistema em termos da arquitetura de dados e de módulos
- Projeto Detalhado: refinamento visando à codificação e especificação dos programas
- Fundamentos (Princípios)
 - Particionamento (decomposição)
 - Um problema P complexo deve ser decomposto em subproblemas menores que devem ser entendidos e solucionados separadamente. A combinação das sub soluções resolve o problema original.
 - Abstração
 - Recomenda-se que um elemento de projeto deve ser representado apenas por suas características essenciais, ou seja, detalhes desnecessários são descartados, de forma a permitir sua distinção de outros elementos por parte do observador, uma representação o mais simples possível e facilidade de entendimento, comunicação e avaliação.
 - Encapsulamento
 - Módulos devem ser especificados e projetados de modo que a informação (dados ou procedimento) nele contida seja inacessível a outros módulos que não tenham necessidade daquela informação.
 - Modularidade
 - Característica do software que permite a sua inteligibilidade, permite dividi-lo em componentes menores, chamados módulos.
 - Alta coesão e baixo acoplamento
 - Coesão

- Coesão é o quanto uma classe deve ter apenas uma única responsabilidade (não deve assumir responsabilidades que não são suas).
- Uma classe com baixa coesão faz muitas coisas não relacionadas e leva aos seguintes problemas: difícil de entender, difícil de reusar, difícil de manter e constantemente sendo afetada por outras mudanças
- Processual: associação a partir dos relacionamentos algorítmicos
- Comunicação: toda operação de um módulo se comunica entre si
- Sequencial: saída do elemento serve como entrada para o próximo
- Funcional: quando as operações de um módulo podem ser descritas em uma frase



○ Acoplamento

- O acoplamento significa o quanto um objeto depende do outro para funcionar
- Com uma classe possuindo forte acoplamento, temos os seguintes problemas: mudanças em uma classe relacionada força mudanças locais à classe, a classe é mais difícil de entender isoladamente e a classe é mais difícil de ser reusada, já que depende da presença de outras classes
- Dados: saída do elemento serve como entrada para o próximo
- Controle: flags de controle entre objetos para que um objeto controle as etapas de processamento de outro

