

Coesão e Acoplamento

Revisão sobre o Projeto

1. O projeto deve ser rastreável para o modelo de análise;
2. Sempre considere a arquitetura do sistema a ser construído;
3. O projeto dos dados é tão importante quanto o projeto das funções de processamento;
4. As interfaces (ambas internas e externas) devem ser projetadas com cuidado;
5. A interface com o usuário deve estar sintonizada com as necessidades do usuário final;
6. O projeto, no nível dos componentes, deve ser funcionalmente independente;
7. Componentes devem ser fracamente acoplados entre eles e com o ambiente externo;
8. As representações (modelos) de projeto devem ser entendidas facilmente;
9. O projeto deve ser desenvolvido iterativamente.

Independência Funcional

- Para tentar administrar a complexidade inerente de sistemas de software, emergiu o conceito chamado “Independência Funcional”.
- Este conceito está ligado à modularidade, ocultação de informações e abstração.
- Em sistemas de software, a Independência Funcional pode ser medida através de dois critérios: **coesão e acoplamento**.

Independência Funcional

- As métricas de coesão e acoplamento tratam de avaliar como os componentes dependem uns dos outros.
- Neste caso, não são considerados como os requisitos funcionais, que levaram a criação destes componentes, são dependentes uns dos outros e nem se o conjunto de requisitos funcionais escolhido é apropriado.
- Quando se escolhe um conjunto de requisitos funcionais interdependente, os componentes da solução gerada tendem a ser interdependentes também.

Coesão

- Coesão é uma medida da força funcional relativa de um módulo;
- A coesão mede o grau com que as tarefas executadas por um único módulo se relacionam entre si;
- Para software orientado a objetos, coesão também pode ser conceituada como sendo o quanto uma classe encapsula atributos e operações que estão fortemente relacionados uns com os outros;

Coesão

Uma classe com baixa coesão faz muitas coisas não relacionadas e leva aos seguintes problemas:

- Difícil de entender
- Difícil de reusar
- Difícil de manter
- "Delicada": constantemente sendo afetada por outras mudanças

Tipos de Coesão

- Coincidente
- Lógico
- Temporal
- Procedural
- De comunicação
- Sequencial
- Funcional

Tipos de Coesão

Coesão Coincidental

- Há nenhuma (ou pouca) relação construtiva entre os elementos de um módulo
- No linguajar OO:
 - Um objeto não representa nenhum conceito OO
 - Uma coleção de código comumente usado e herdado através de herança (provavelmente múltipla);

Coesão Lógica

- Um módulo faz um conjunto de funções relacionadas, uma das quais é escolhida através de um parâmetro ao chamar o módulo;
- Semelhante a acoplamento de controle;

Tipos de Coesão

Coesão Temporal

- Elementos estão agrupados no mesmo módulo porque são processados no mesmo intervalo de tempo
- Exemplos comuns:
 - Método de inicialização que provê valores defaults para um monte de coisas diferentes
 - Método de finalização que limpa as coisas antes de terminar

Tipos de Coesão

Coesão Procedural

- Associa elementos de acordo com seus relacionamentos procedurais ou algorítmicos;
- Um módulo procedural depende muito da aplicação sendo tratada;
- Junto com a aplicação, o módulo parece razoável;
- Sem este contexto, o módulo parece estranho e muito difícil de entender;

Coesão de Comunicação

- Todas as operações de um módulo operam no mesmo conjunto de dados e/ou produzem o mesmo tipo de dado de saída;
- Cura: isole cada elemento num módulo separado;
- "Não deveria" ocorrer em sistemas OO usando polimorfismo (classes diferentes para fazer tratamentos diferentes nos dados);

Tipos de Coesão

Coesão Sequencial

- A saída de um elemento de um módulo serve de entrada para o próximo elemento;
- Cura: decompor em módulos menores;

Coesão Funcional

- Um módulo tem coesão funcional se as operações do módulo puderem ser descritas numa única frase de forma coerente;
- Num sistema OO:
 - Cada operação na interface pública do objeto deve ser funcionalmente coesa;
 - Cada objeto deve representar um único conceito coeso;

Acoplamento

- Acoplamento é uma medida da interdependência relativa entre os módulos;
- Para sistemas de software orientados a objetos pode se definir acoplamento como sendo o grau com o qual classes estão conectadas entre si;
- Existem métricas definidas na literatura para coesão de um módulo e acoplamento entre módulos;

Acoplamento

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;
- A classe é mais difícil de ser reusada, já que depende da presença de outras classes;

Tipos de acoplamento:

- Acoplamento de dados;
- Acoplamento de controle;

Acoplamento de Dados

- **Situações**

- Saída de um objeto é entrada de outro;
- Uso de parâmetros para passar itens entre métodos;

- **Ocorrência comum:**

- Objeto a passa objeto x para objeto b;
- Objeto x e b estão acoplados;

- Uma mudança na interface de x pode implicar em mudanças a b;

Acoplamento de Controle

Passar flags de controle entre objetos de forma que um objeto controle as etapas de processamento de outro objeto

Ocorrência comum:

Objeto a manda uma mensagem para objeto b

- b usa um parâmetro da mensagem para decidir o que fazer
-

Tipos de Acoplamento

Acoplamento de dados globais

- Dois ou mais objetos compartilham estruturas de dados globais;
- É um acoplamento muito ruim pois está escondido;
- Uma chamada de método pode mudar um valor global e o código não deixa isso aparente;

Acoplamento de dados internos

- Um objeto altera os dados locais de um outro objeto;

Ocorrência comum:

- Dados públicos, package visibility ou mesmo protected em java;