

## Programação e Desenvolvimento de Software 2

### Projeto de Software

Prof. Helton Fábio de Matos  
hfmato@dcc.ufmg.br

## O Software

- O Software está incorporado em praticamente todos os aspectos da vida contemporânea.
- Para se desenvolver uma boa solução, é necessário fazer um esforço para compreender o problema a ser resolvido.
- Levantar uma boa definição dos requisitos é um dos primeiros passos.

## Engenharia de Software

- Aplicação sistemática de conhecimento científico, tecnológico, métodos e experiência para projetar, implementar, testar e documentar software.
- Aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software.

(ISO/IEC/IEEE 24765:2010, System and software engineering – Vocabulary)

## Engenharia de Software

- Abrange um processo, um conjunto de métodos e um leque de ferramentas que possibilitam o desenvolvimento de software de alta qualidade.



## Desenvolvimento de software

### Processo

- É o conjunto de atividades, ações e tarefas realizadas para criação de um software.
- Há várias sugestões de modelos para processos de desenvolvimento de software.
- O processo envolve:
  - a identificação das necessidades do usuário e a representação destas como requisitos de software;
  - a transformação dos requisitos em projeto de software;
  - a implementação do projeto em código;
  - teste do código;
  - implantação e verificação de uso operacional.

## Complexidade

- Complexidade de um software é uma propriedade inerente que impacta seu desenvolvimento.
- Ela deriva de pelo menos quatro fontes:
  - Da complexidade do domínio do problema.
  - Da complexidade do processo de desenvolvimento.
  - Das flexibilidades permitidas pela tecnologia de software.
  - Dos problemas decorrentes do comportamento de sistemas discretos ao longo do tempo.

## Complexidade

- A questão é: o ser humano tem uma capacidade limitada para lidar simultaneamente com todas as informações de um sistema complexo.
- Em 1956, G. March Miller concluiu que um ser humano consegue processar simultaneamente em torno de sete, mais ou menos dois, blocos de informação.
- O próprio Miller sugeriu que uma forma de superar esta limitação seria organizar blocos de informação em níveis, tal que em cada nível as informações sejam organizadas em número de blocos administrável.



## Complexidade

- Uma propriedade importante de sistemas complexos é que em geral eles possuem uma estrutura modular hierárquica.
- As técnicas básicas para se lidar com a complexidade são:
  - Abstração.
  - Decomposição.
  - Hierarquização.



## Abstração

- Abstração é a técnica de se idealizar o modelo de um objeto complexo descrevendo de forma genérica apenas as propriedades mais importantes, ignorando as não essenciais.



## Decomposição

- Qualquer problema complexo pode ser tratado mais facilmente se for subdividido em partes que possam ser resolvidos independentemente.
- Separação por interesse é um conceito de projeto que sugere tratar um problema complexo subdividindo-o em problemas menores para tratamento ou otimização.
- Programação Modular é a técnica que enfatiza organização de um programa em módulos funcionais.
- Em princípio, módulo é a unidade básica para a decomposição de um sistema de software.
- Aplicação recursiva: módulos subdivididos em submódulos.
- Facilita divisão de trabalho, o desenvolvimento, a composição e a reutilização.



## Encapsulamento

- Como decompor uma solução de software para obter o melhor conjunto de módulos?
- Princípio do encapsulamento de informações: projetar módulos para conter informações (algoritmos e dados) que sejam inacessíveis a outros módulos que não necessitem de tais informações.
- O acesso a informações de um módulo deve ser feito através de uma interface.

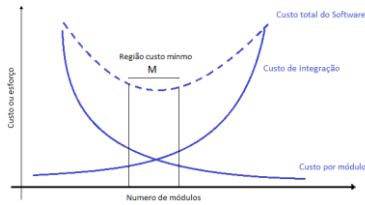


## Independência Funcional

- Módulos devem ter uma função bem definida.
- Módulos devem realizar suas funções com dependência mínima de outros.
- A independência é avaliada por dois critérios:
  - Coesão – robustez funcional relativa ao módulo.
  - Acoplamento – indica a interdependência relativa a outros módulos.
- Objetivos em um bom projeto: alta coesão e baixo acoplamento.



## Modularização: custo



Projeto de Software

14

## Conceitos de projeto

- O objetivo de um projeto é desenvolver um modelo que apresente a solução do problema a partir dos requisitos, das restrições e das metas de qualidade estabelecidas.
- Um projeto deve apresentar os recursos necessários para implementação da solução.
- Elementos básicos em um projeto:
  - Arquitetura da solução.
  - Estrutura de dados.
  - Interface (com o ambiente e usuários).
  - Componentes (dados, procedimentos e interface).

Projeto de Software

15

## Arquitetura do software

- É a descrição dos módulos componentes da solução.
- É o modelo que descreve a hierarquia software em termos de:
  - Módulos componentes (dados, funções e interface)
  - Relações entre estes módulos;
  - Estruturas de dados.

Projeto de Software

16

## Questões práticas

- Como identificar e documentar módulos?
- Quais os elementos um módulo deve possuir?
- Como módulos organizam dados e procedimentos?
- Qual é a interface de acesso ao módulo?
- O paradigma de Orientação a Objetos oferece boas soluções para estas questões.

Projeto de Software

17

## Exercicio

Projetar um diagrama de classes para um sistema que atenda à demanda descrita a seguir.

*Uma editora possui cinco publicações e deseja projetar um sistema para controlar a venda de assinaturas.*

*Cada assinatura vendida deve ser lançada no sistema com os seguintes dados: nome e endereço do cliente, nome da publicação e número de edições assinadas.*

*Um cliente pode assinar mais de uma publicação, mas cada uma é lançada individualmente no sistema.*

*O sistema deve rodar semanalmente e gerar as etiquetas a serem coladas nas publicações para envio.*

*Há publicações que são enviadas semanalmente enquanto outras são mensais.*

*Quando o número de edições assinadas for atingido, o registro da assinatura é eliminado.*

Projeto de Software

18

## Modelagem via CRC

Assinatura	
Responsabilidades	Colaboração
Nome da publicação	Publicação
Edição inicial	Cliente
Atualiza()	
Consulta()	
Cliente	
Responsabilidades	Colaboração
Nome	Publicação
endereço	Assinatura
Atualiza(enderec())	
Consulta()	
Publicação	
Responsabilidades	Colaboração
Nome	Assinatura
Edição atual	
AtualizaCampos()	
Lista()	

Projeto de Software

19

## Modelagem via CRC

Sistema	
Responsabilidades	Colaboração
inscreverPublicação()	Publicação
inscreverCliente()	Cliente
inscreverAssinatura()	Assinatura

Etiqueta	
Responsabilidades	Colaboração
Dados do cliente	Cliente
Dados da publicação	Publicação

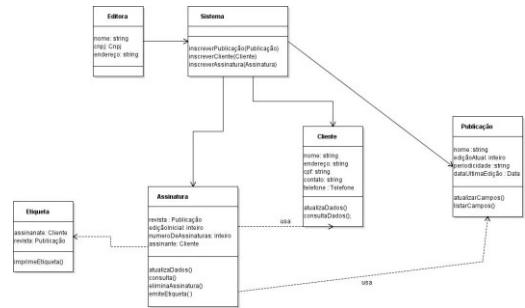


Projeto de Software

20

## Diagrama de Classes

Classes do domínio do problema

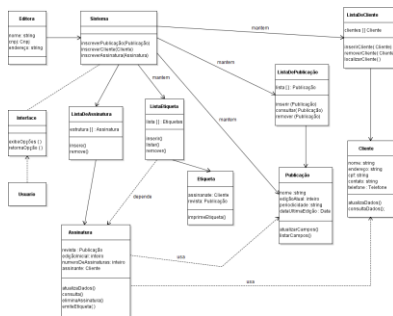


Projeto de Software

21

## Diagrama de Classes

Classes do projeto de implementação



Projeto de Software

22

UFMG  
UNIVERSIDADE FEDERAL  
DE MINAS GERAIS

Fim

DCC  
DEPARTAMENTO DE  
CIÊNCIA DA COMPUTAÇÃO