

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Programação e Desenvolvimento de Software 2

Conceito de especificação de software

Prof. Douglas G. Macharet
douglas.macharet@dcc.ufmg.br

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

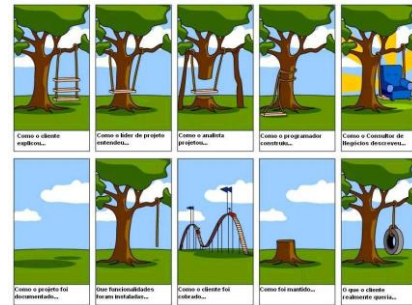
Introdução

- Introdução à Orientação a Objetos
 - O que é um objeto?
 - Como criar esses objetos? (linguagem)
- Requisitos/Análise/Modelagem/Design
 - Quais objetos devem ser criados?
 - Quais características eles devem possuir?
 - Atributos (dados), comportamentos, ...

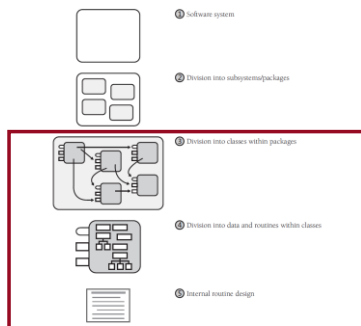
Introdução

- Análise e Projeto
 - Quais as reais necessidades do cliente?
 - Requisitos → Software
- Software complexo → Planejamento
- Sistema mal projetado
 - Prejudica manutenção/extensão
 - Tempo → Dinheiro

Introdução



Níveis de modelagem



Análise e Modelagem

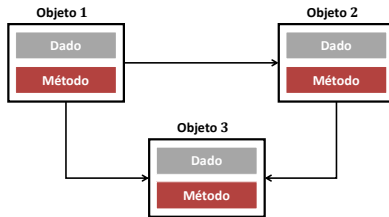
■ Estruturada

Vantagens	Desvantagens
Mais intuitiva (top-down)	Fases sequenciais
Baseada em funcionalidade	Custo inicial muito alto
Especificação mais simples	Baixa reusabilidade

■ Orientada a Objetos

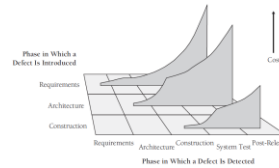
Vantagens	Desvantagens
Baseada nos dados	Funcionalidades nos objetos
Modularidade	Difícil identificar objetos
Extensibilidade	Difícil visualizar as ligações

Modelo Orientado a Objetos



Especificação de requisitos

- Identificação e especificação de requisitos
 - Identificar e formalizar as (reais) necessidades
 - Lidar com domínios desconhecidos



Especificação de requisitos

- Requisitos
 - Objetivos/restrições estabelecidas por clientes e usuários que definem as propriedades desejadas
- Tipos
 - Funcionais
 - Funcionalidades que o software deve oferecer
 - Não-funcionais
 - Qualidades globais de um software

Histórias de usuário (User Stories)

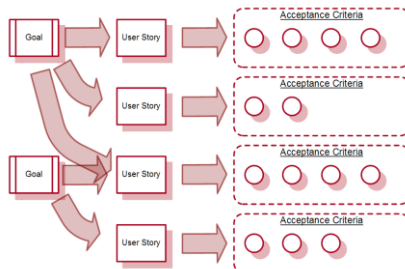
- User Story
 - Descrição informal em linguagem natural de um ou mais recursos de um sistema de software
 - Funcionalidades que são importantes (desejadas) para os usuários ou clientes do sistema
 - Conta uma história acerca de como alguém utiliza o produto para alcançar um objetivo

User Stories \neq Requisitos

User Stories \rightarrow Requisitos

<http://www.extremeprogramming.org/rules/userstories.html>

Histórias de usuário (User Stories)



Histórias de usuário (User Stories)

Etapas

- Cartão
 - Descrição por escrito da história
- Conversação
 - Discussão sobre a história para explicar detalhes
- Confirmação
 - Testes que confirmam a conclusão da história



<https://torjeffries.com/blog/articles/epiccard-conversation-confirmation/>

Histórias de usuário (User Stories)

Cartão



Como [Quem?] eu quero [O quê?] para [Por quê?].

Exemplo: Como um cliente da operadora de saúde eu quero procurar um médico pelo nome para obter o endereço do seu consultório.

Histórias de usuário (User Stories)

Crítérios de aceitação

- Critérios que precisam ser alcançados para que a User Story atenda os requisitos do usuário e seja aceita (dada como concluída)
 - Questões associadas a usabilidade, tratamento de erros, desempenho, funcionalidade, ...
 - Devem apresentar a intenção, não a solução
- 3-5 testes/critérios de aceitação por história

Histórias de usuário (User Stories)

Crítérios de aceitação

Como um operador de marcação de consulta eu gostaria de visualizar a agenda de consultas dos médicos para saber quais horários estão disponíveis para novas marcações.

- Critérios de aceitação
 - Visualização da agenda por dia, semana e mês
 - Visualizar a agenda de consultas do mês atual e do mês posterior
 - Cada consulta agendada deve mostrar o nome e o convênio do paciente

Histórias de usuário (User Stories)

Exemplos

Como titular de um cartão de crédito, eu quero ver o extrato da minha conta para poder pagar o saldo devedor.

- Critérios de aceitação
 - Exibir saldo total
 - Exibir o pagamento mínimo devido
 - Exibir data de vencimento do pagamento
 - Exibir mensagem de erro se o serviço não estiver respondendo

Como professor, eu gostaria de gerar um relatório de avaliação para poder avaliar o desempenho dos alunos.

- Critérios de aceitação
 - Mostrar a pontuação atual da avaliação de um aluno
 - Exibir a pontuação de avaliação passada do aluno
 - Fornecer uma opção para Imprimir/Salvar/Compartilhar

← Uma nova história?

Histórias de usuário (User Stories)

Exercícios

- Definir User Stories para os seguintes casos
 - Uber
 - Pedir um carro
 - Netflix
 - Buscar um filme
 - Waze
 - Procurar um caminho

Modelagem CRC

- Kent Beck e Ward Cunningham (1989)
 - Introdução à *object-oriented thinking (design)*
- Ferramenta/Metodologia/Processo
 - Análise, modelagem e projeto de sistemas OO
- Nenhum processo garante bons resultados, mas é possível torná-los mais prováveis
 - Utilizar POO não é garantia de qualidade!

Modelagem CRC

- Cartões CRC
 - Class-Responsibility-Collaboration
 - 10cm x 15cm (físicos)
- Por que utilizar cartões?
 - Barato, portátil, disponível, familiar, ...
 - Objetividade, simplicidade, clareza, ...
 - Participação de diferentes atores

Modelagem CRC

Classe:	
Responsabilidades	Colaborações

Modelagem CRC

- Processo iterativo e iterativo
 - Usuários, analistas, designers, ...



- Independente de linguagem de programação
- Auxilia profissionais com pouca experiência

Modelagem CRC

Classe

- Classe
 - Conjunto de elementos que compartilham uma mesma estrutura e comportamento comum
- Atributos
 - “Do que isso é feito?”
- Comportamentos
 - “O que isso pode fazer?”

Modelagem CRC

Responsabilidade

- Responsabilidade
 - Comportamento demandado de um objeto
 - Contrato/obrigação para com o sistema
- Tipos de responsabilidade
 - Fornecer informações (conhecimento)
 - Fazer alguma coisa (realização)

Modelagem CRC

Responsabilidade – Tipos

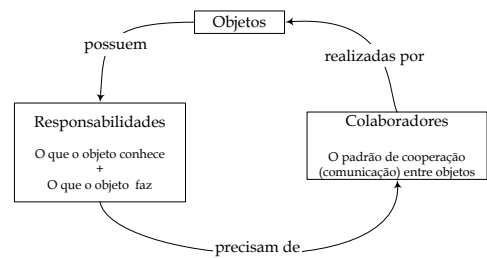
- Conhecimento
 - Objeto conhece e gerencia seus dados internos
 - Objeto conhece outros objetos relacionados
- Realização
 - Objeto faz algo sobre ele próprio
 - Conhece e coordena ações em outros objetos

Modelagem CRC

Colaboração

- Colaboração
 - Processo através do qual vários objetos interagem para fornecer um comportamento
 - Possui a responsabilidade, mas não a informação
- Relação cliente/servidor (serviço)

Modelagem CRC



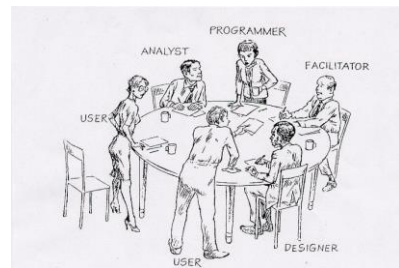
Modelagem CRC

Sessão

- Extração e modelagem dos requisitos
 - Simulação do sistema e execução de cenários
- Aproximadamente seis pessoas
 - Especialistas, projetistas, analistas e moderador
- Princípios para uma sessão de sucesso
 - Todas as ideias são potencialmente boas ideias
 - Flexibilidade (diferentes visões é bom!)

Modelagem CRC

Sessão



Modelagem CRC

Sessão

- Brainstorming
- Preenchimento dos cartões
- Determinar possíveis cenários
- Executar cenários para refinar os cartões

Modelagem CRC

Sessão – Brainstorming

- Escolher o time que irá participar
- Conversa geral sobre o sistema/requisitos
 - É realmente necessário? Por que? Qual ganho?
 - O que deve fazer? Quem vai utilizar? Como?
- Explicar como o processo CRC funciona

Modelagem CRC

Sessão – Preenchimento dos cartões

- **Classes**
 - Procurar pelos substantivos nas descrições
 - Dado o contexto, decidir os mais importantes
- **Responsabilidades**
 - Verbos → Comportamento
 - O que essa classe conhece e faz?
- **Colaboradores**
 - A classe consegue realizar isso sozinha?
 - Quais informações/serviços ela precisa?

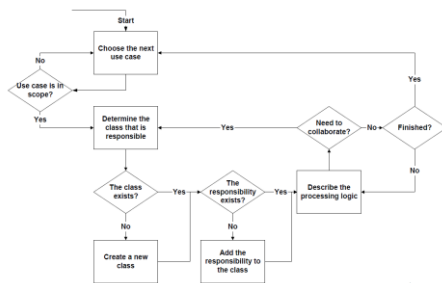
Modelagem CRC

Sessão – Cenários

- **Determinação de possíveis casos de uso**
 - Interação entre um usuário e o sistema
- **Verificar a aplicação e viabilidade**
- **Associar um cartão ao cenário descrito**
- **Descrever a lógica de execução**
 - “Algoritmo” descrito pelos especialistas
- **Realizar um teste/execução do cenário**

Modelagem CRC

Sessão – Cenários



Exemplo 1

- CRC para o SIGA da UFMG

Classe: Estudante	
Responsabilidades	Colaborações
ID	Curso
Nome	Disciplina
Fazer matrícula	
Trancar matrícula	

Exemplo 1

- CRC para o SIGA da UFMG

Classe: Disciplina	
Responsabilidades	Colaborações
Número	Departamento
Nome	Professor
Carga horária / Créditos	Estudante
Professor responsável	
Mostrar lista de alunos	

Exemplo 2

- Loja de e-commerce
- **Quais as possíveis classes?**
 - Cliente, Produto, Pedido, Carrinho, ...
- **Quais as responsabilidades de Pedido?**
 - Conhecer: Produtos que a integram
 - Fazer: Calcular o total da compra
- **Quais as colaborações?**
 - Cliente, Produto

Exemplo 3

- Sistema para uma oficina mecânica
- Quais as possíveis classes?
 - Cliente, Veículo, Orçamento, OS, ...
- Quais as responsabilidades de Orçamento?
 - Conhecer: Data, veículo, peças, ...
 - Fazer: Calcular o total do orçamento
- Quais as colaborações?
 - Cliente, Veículo, Peça

Considerações finais

- Cartões CRC
 - Clareza e objetividade
 - “Pensamento orientado à objetos”
 - Bons para discussão mas não documentação
- Quantas responsabilidades a classe deve ter?
 - Qual impacto isso terá no sistema?
 - Como distribuir essas responsabilidades?

Considerações finais

- Vantagens
 - Maior conhecimento do domínio do problema
 - Ótimo ambiente para aprendizado
- Desvantagens
 - Aplicação limitada em relação ao design
 - Pouco formal, ainda distante da implementação, ...
 - Outras alternativas → UML
 - Dificuldades em juntar toda a equipe

Considerações finais

Exercícios

- Sistema de gestão acadêmica
- Sistema bancário
- Sistema para um smartphone
 - Aplicativo (Whatsapp)
 - Jogo (Angry Birds)
- Sistema operacional