

Programação e Desenvolvimento de Software 2

Fundamentos de projeto e especificação de software

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

Introdução

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

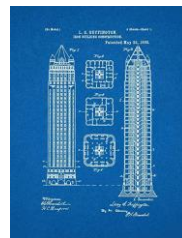
Introdução

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



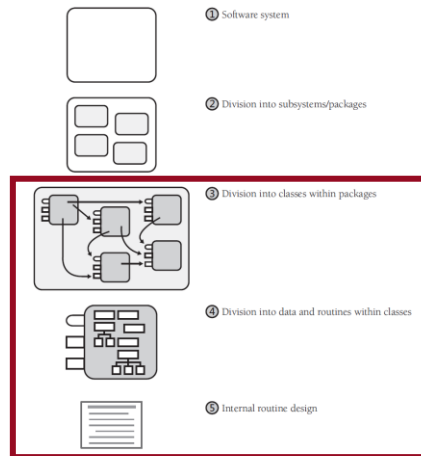
Burj Khalifa

Introdução



Acho que vou precisar reconstruir o primeiro andar do zero.

Níveis de modelagem



Fonte: Code Complete: A Practical Handbook of Software Construction, Second Edition



PDS 2 - Fundamentos de projeto e especificação de software

5

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

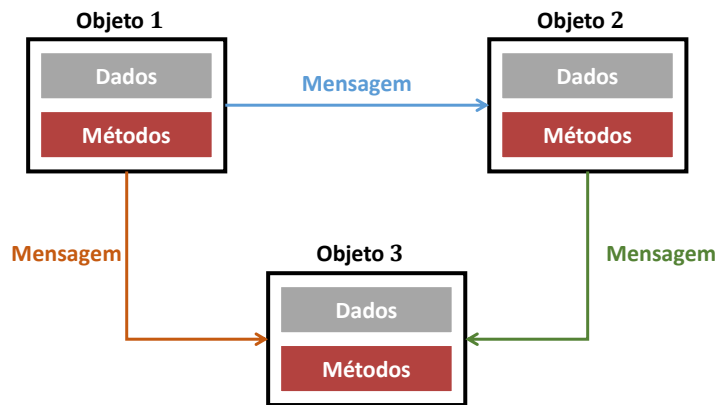
Vantagens	Desvantagens
Baseada nos dados	Funcionalidades nos objetos
Modularidade	Difícil identificar objetos
Extensibilidade	Difícil visualizar as ligações



PDS 2 - Fundamentos de projeto e especificação de software

6

Modelo Orientado a Objetos

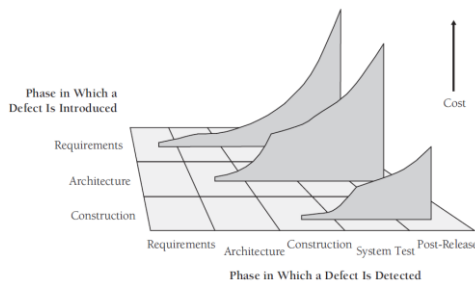


Especificação de requisitos

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

Especificação de requisitos

- Identificação e especificação de requisitos
 - Identificar e formalizar as (reais) necessidades
 - Falhas → Alto custo mais tarde para arrumar



Fonte: Code Complete: A Practical Handbook of Software Construction, Second Edition



PDS 2 - Fundamentos de projeto e especificação de software

9

Especificação de requisitos

A identificação e especificação de requisitos de software é uma atividade bastante desafiadora!

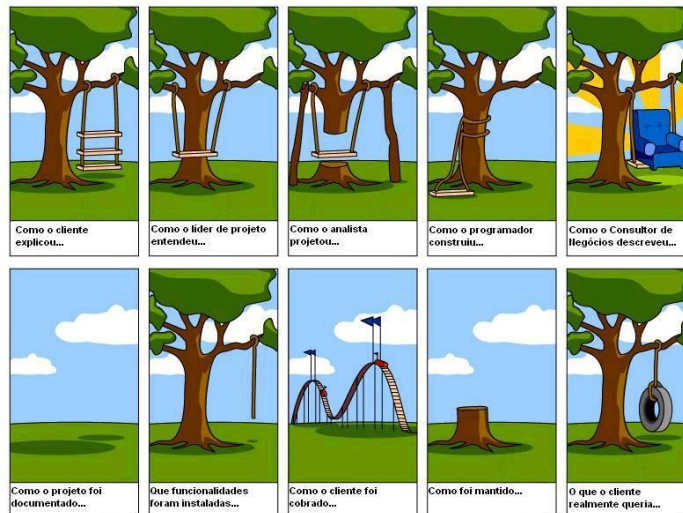
- Existe toda uma área voltada para isso
- Nosso objetivo é dar uma visão introdutória
 - Sistemas de tamanho pequeno/médio
 - Abordagem mais aprofundada → Engenharia de Software



PDS 2 - Fundamentos de projeto e especificação de software

10

Especificação de requisitos



Modelagem

- Definição (organização) dos componentes do sistema
 - Requisitos → Implementação
- Design Principles
 - KISS - “Keep It Simple, Stupid”
 - DRY - “Don’t Repeat Yourself”
 - YAGNI - “You Ain’t Gonna Need It”
 - SOLID
- Quanto mais experiente, mais “soluções” você conhece



https://en.wikipedia.org/wiki/KISS_principle
https://en.wikipedia.org/wiki/Don%27t_repeat_yourself
https://en.wikipedia.org/wiki/You_aren%27t_gonna_need_it
<https://en.wikipedia.org/wiki/SOLID>

Processo de desenvolvimento



Requisitos



Modelagem



Implementação

- **Histórias de usuário (User Stories)**
 - Identificação e “formalização” dos requisitos
- **Cartões CRC**
 - Representação básica dos elementos e interação entre eles

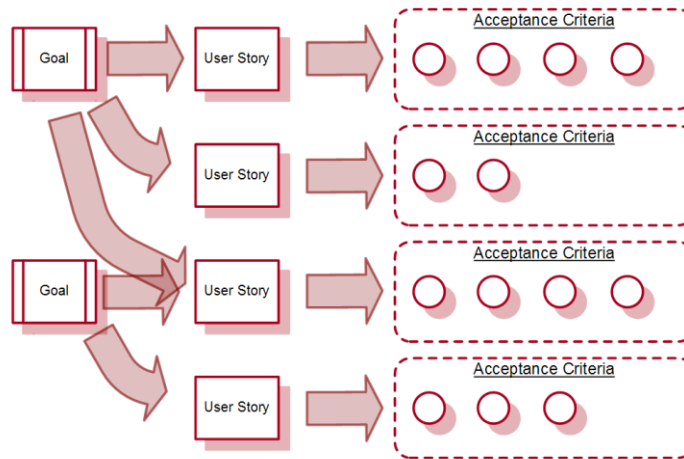
Histórias de usuário (User Stories)

- **User Story**
 - Descrição **informal** em linguagem natural de um ou mais recursos desejados/necessários para um sistema de software
 - Funcionalidades importantes para os usuários/clientes do sistema
 - Conta uma história acerca de como alguém irá utilizar o produto (sistema) para alcançar um determinado objetivo

User Stories \neq Requisitos

User Stories \rightarrow Requisitos

Histórias de usuário (User Stories)



<http://tynerblain.com/blog/2009/07/06/writing-complete-user-stories/>

DCC **111**

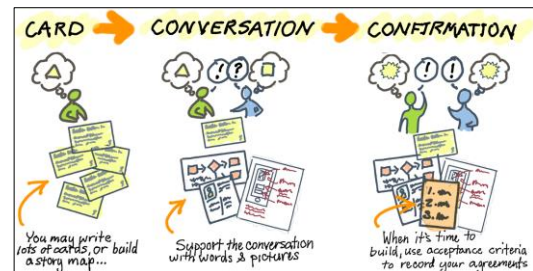
PDS 2 - Fundamentos de projeto e especificação de software

15

Histórias de usuário (User Stories)

Etapas

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



User Story Mapping. Jeff Patton.

<https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/>

DCC **111**

PDS 2 - Fundamentos de projeto e especificação de software

16

Histórias de usuário (User Stories)

Cartão



- Diferentes papéis dos usuários do software
- Ex: visitante, cliente, admin, ...



- Qual o objetivo? Qual a funcionalidade?
- Sempre pelo ponto de vista do usuário



- Por que tem valor para o usuário? Qual o contexto?
- Podem haver diferentes formas de alcançar um objetivo

Histórias de usuário (User Stories)

Cartão

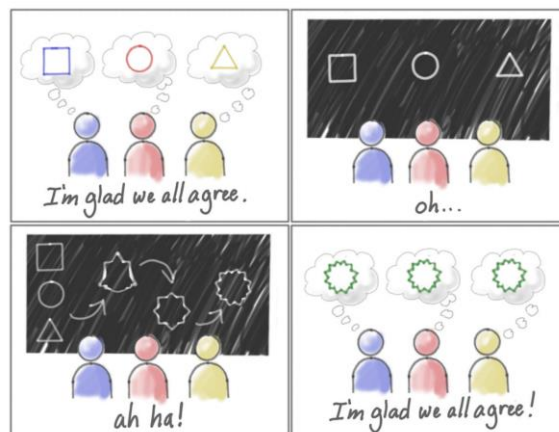


Como **[Quem?]** eu quero **[O quê?]** para **[Por quê?]**.

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)

Conversação



Histórias de usuário (User Stories)

Critérios de aceitação

- Critérios que precisam ser alcançados para que a User Story atenda os requisitos do usuário e seja aceita (concluída)
 - Devem apresentar a intenção, não a solução
 - Mínimo necessário para ser útil ao usuário
 - Evitar a adição de características desnecessárias
- 3-5 testes/critérios de aceitação por história

Histórias de usuário (User Stories)

Critérios de aceitação

- Deixar bem claro o que o time deve fazer antes de iniciar
- Permitir ao time saber exatamente quando ela está completa
- Auxiliar no desenvolvimento de testes automatizados

- Características funcionais e não funcionais
- Questões relativas à desempenho e usabilidade
- Tratamento de erros (cenários negativos)
- Impacto para outras funcionalidades

Histórias de usuário (User Stories)

Critérios 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**.

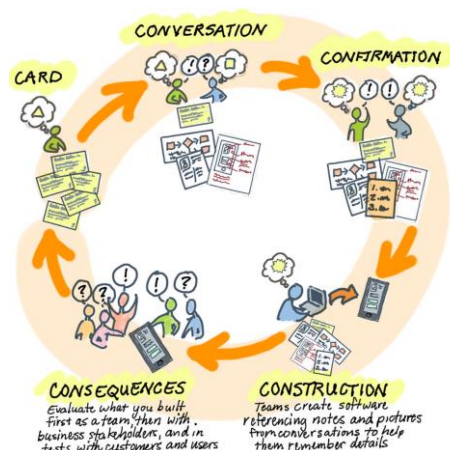
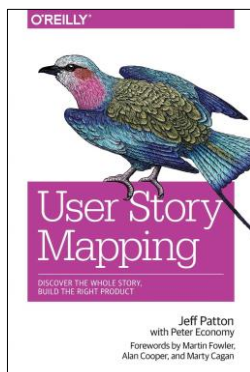
- Critérios de aceitação
 - Exibir saldo total
 - Exibir o pagamento mínimo devido
 - Exibir data de vencimento do pagamento
 - Exibir mensagem de erro se o serviço não estiver respondendo

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

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

← Uma nova história?

Histórias de usuário (User Stories)



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 sozinho garante bons resultados
 - 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 interativo
 - Usuários, analistas, designers, ...



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

Modelagem CRC

Classe

- Conjunto de elementos (membros) que compartilham uma mesma estrutura e um comportamento (objetivo) comum

Atributos

“De 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

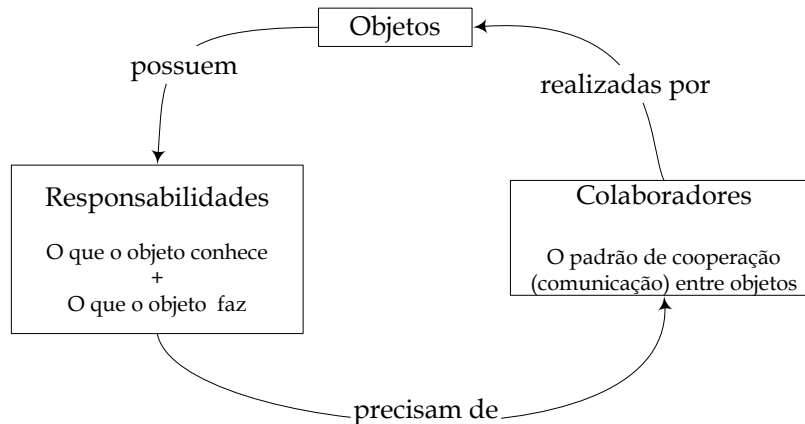
- **Conhecimento**
 - Objeto conhece e gerencia seus dados internos
 - Objeto conhece outros objetos relacionados
- **Realização**
 - Objeto faz algo sobre ele próprio
 - Conhece e coordena ações em outros objetos

Modelagem CRC

Colaboração

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

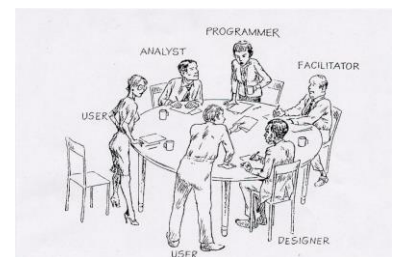
Modelagem CRC



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 algo bom!)



The CRC Card Book. David Bellin, Susan Suchman Simone.

Modelagem CRC

Sessão

1. Brainstorming
2. Preenchimento dos cartões
3. Determinar possíveis cenários
4. Executar cenários para refinar os cartões

Modelagem CRC

Sessão – Brainstorming

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

Modelagem CRC

Sessão – Preenchimento dos cartões

- **Classes**
 - Procurar pelos substantivos nas descrições
 - Dado o contexto, decidir os mais importantes
- **Responsabilidades**
 - O que essa classe conhece e faz?
 - Verbos → Comportamento
- **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**

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

- 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
 - Dificuldades em juntar toda a equipe
 - Aplicação limitada em relação ao design
 - Pouco formal, ainda distante da implementação, ...
 - Outras alternativas → UML

Considerações finais

Exercícios

- Sistema locadora, bancário
- Uber, Netflix, Waze, Amazon
- Sistema para um smartphone
 - Aplicativo (Whatsapp)
 - Jogo (Angry Birds)
- Sistema operacional

