

Abstração e Modelagem

Modelagem e Programação
Orientada a Objetos
BSI – DEINFO – UFRPE

1

Software

- Motivação: A cada dia a sociedade está mais dependente de sistemas de software
- Anos atrás: bancos, empresas, sistemas telefônicos
- Hoje: celulares, aplicações distribuídas, Internet, aplicações críticas
- Pressão sobre desenvolvedores: rapidez, confiabilidade, preço, complexidade, tamanho

Estamos preparados?

2

Crise do Software

- Cenário atual do desenvolvimento
 - Dificuldade de entendimento dos requisitos
 - Dificuldade de manutenção
 - Duplicação de esforços
 - Proliferação de tecnologias
 - Baixa qualidade

3

Crise do Software

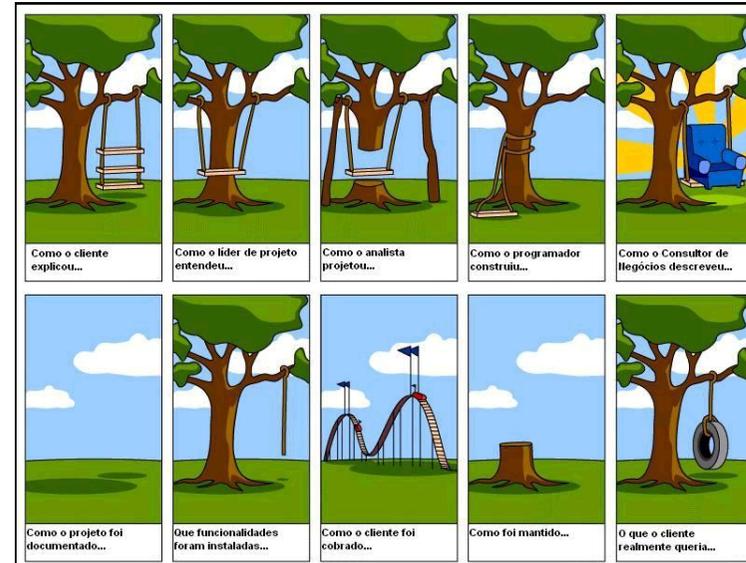
- Sintomas
 - Pouco sucesso real
 - Estouro de orçamento e cronograma são comuns
 - Sistemas em operação deixam de atender muitos requisitos e são difíceis de manter
 - Cancelamento de projetos
- Diferenças para as “outras” engenharias
 - “Hardware is basically impossible to change. But luckily, we’ve got these programs that we can constantly tweak to suit our needs! That’s the opposite of hardware.”

4

Fred Brooks em 1975 já dizia...

- Vamos resolver a crise se
 - planejarmos nossos projetos
 - estruturarmos nossas equipes
 - documentarmos o que fazemos
 - realizarmos comunicação efetiva

5



Engenharia de Software Hoje

- Desenvolver software de qualidade próximo aos custos e prazo estimados
- Utilizar modelos mais baratos para visualizar soluções
- Utilizar soluções já utilizadas com sucesso
 - Reuso de componentes
 - Reuso de idéias
- Melhorar a comunicação na equipe

7

Engenharia de Software Hoje

- Utilização de conceitos ligados ao mundo real
 - Representação uniforme (objetos)
- Organizar o processo com o intuito de aumentar produtividade e qualidade
 - De acordo com o nível de exigência estabelecido

8

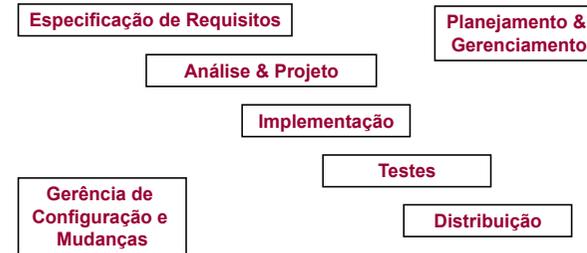
Exemplo: engenharia de software para web

- Dificuldade de coleta de requisitos
- Ênfase na interface com o usuário
 - usabilidade
- Fatores de qualidade: robustez, tolerância a falhas
 - Tratamento de concorrência
 - Segurança
- Compatibilidade com sistemas legados e navegadores mais usados

9

Etapas do Processo de Software

Divisão do processo em um conjunto de atividades



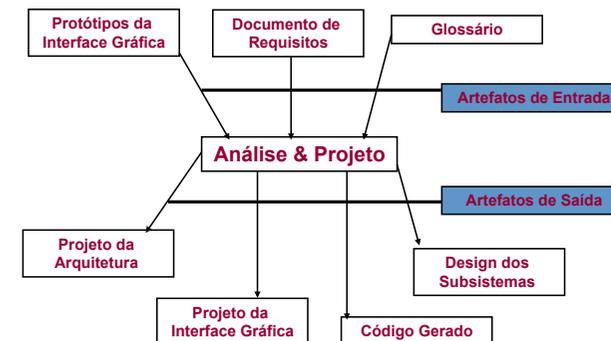
10

Etapas do Processo de Software

- Caracterização das Etapas
 - Cada etapa possui:
 - Objetivo bem definido
 - Atividades a serem realizadas
- Artefatos de Entrada
- Artefatos de Saída
- Ferramentas e Técnicas que são utilizadas

11

Nosso foco!



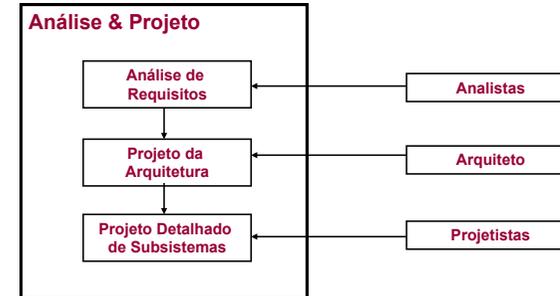
12

Análise e Projeto (Design)

- Análise (*faça a coisa certa*)
 - Investigação do problema
 - Entendimento dos requisitos
 - Ignora detalhes de implementação
- Design ou Projeto (*faça certo a coisa*)
 - Solução, ainda que não implementada
 - Leva em consideração requisitos não-funcionais
 - Normalmente agrega maior conhecimento da equipe

13

Responsáveis



14

Modelagem

- Em software, produto primário é o programa executável
 - Resto é secundário
- Mas secundário não é irrelevante!!
 - Empresas querendo construir prédios complexos do mesmo jeito de uma casinha de cachorro!
- Como desenvolver software do jeito que queremos sem modelar?

15

Modelagem

- Modelar
 - Comunicar estrutura desejada
 - Visualizar e controlar o sistema
 - Entender o sistema que estamos construindo
 - Gerenciar melhor os riscos
- Análise e projeto são etapas que necessitam de modelagem
- Linguagens e ferramentas auxiliam
 - Uso de símbolos auxilia, isso se os símbolos forem conhecidos do público-alvo

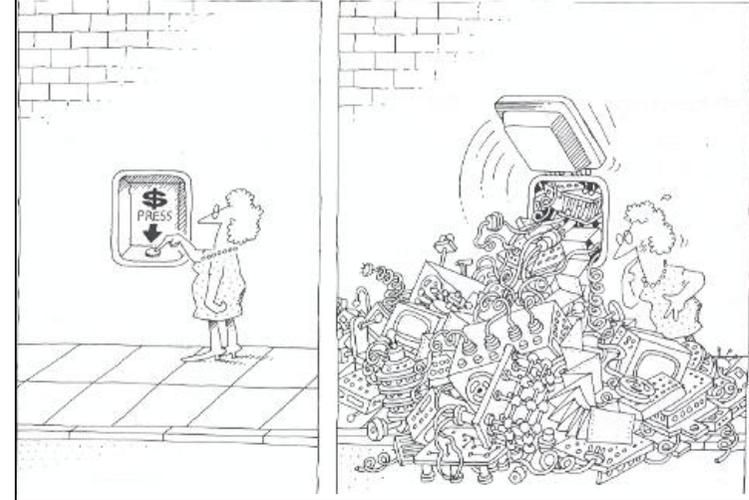
16

Complexidade

- Complexidade de software não pode ser evitada!
- Elementos responsáveis pela complexidade do software
 - Complexidade do domínio do problema
 - Dificuldade no gerenciamento do processo de desenvolvimento
 - A flexibilidade possível em sistemas de software
- Mas podemos tratar
 - Decomposição
 - Abstração

17

Decomposição: Ilusão de Simplicidade

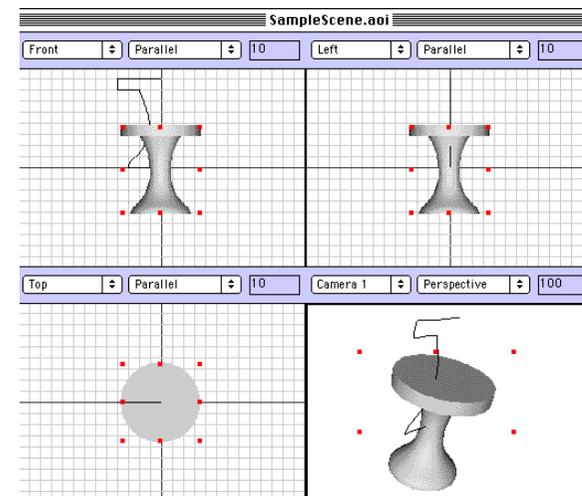


Decomposição

- Decompor em partes menores e menores, refinando cada parte independentemente
- Satisfazendo, assim, a restrição real do ser humano
 - Para entender qualquer nível de um dado sistema precisamos compreender um aspecto por vez
- Modelagem deve seguir este princípio
 - Vários modelos necessários
 - Impossível entender de forma completa

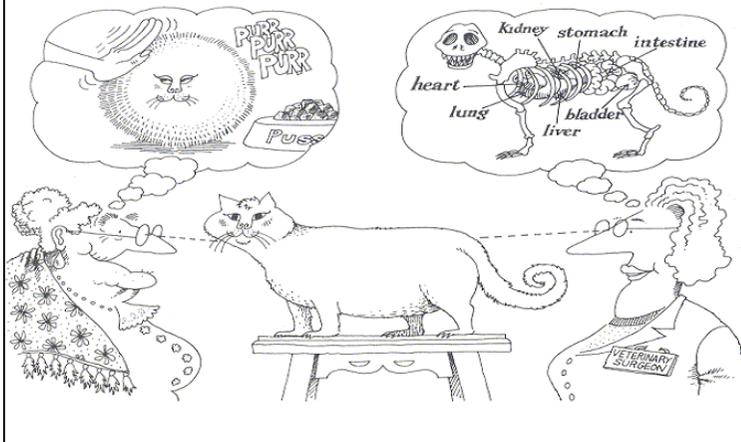
19

Decomposição



20

Abstração: Dependente do Observador



Abstração

- É uma das formas que nós humanos tratamos a complexidade do mundo
- Aspectos irrelevantes ao problema estudado devem ficar de fora



22

Abstração em Modelos

- Abstração cujo propósito é permitir que se conheça algo antes de construí-lo
- O modelo omite os detalhes não essenciais tornando a manipulação mais fácil
- Necessários durante a construção de softwares complexos

23

Importância dos Modelos

- Visualizar e controlar o sistema a ser construído
 - Conhecer bem o público-alvo da modelagem
 - Decomposição
- Melhora o entendimento do sistema que estamos construindo observando oportunidades de reuso e simplificação
 - Uso de notações conhecidas
- Permitir evoluir em situações esperadas e não usuais e alterá-los quando falhar para o comportamento desejado
- Gerenciar riscos
 - Manter nível de abstração correto

24

Ferramentas CASE

- Sistemas para auxiliar o desenvolvimento de sistemas
- Objetivo: melhorar a produtividade dos desenvolvedores
- Suporte para a maioria das etapas de desenvolvimento
- Desafio
 - Facilitar o trabalho e melhorar produtividade sem ter o efeito contrário (ferramentas podem em muitos casos diminuir a produtividade)

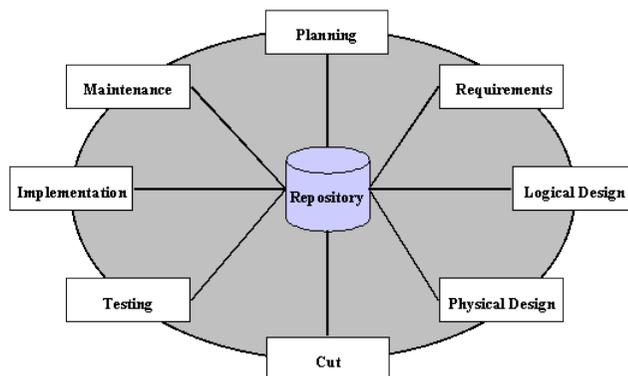
25

Ferramentas CASE

- Ferramentas para análise e projeto:
 - Facilitar a criação de modelos do sistema
 - Avaliação de modelos
 - Transformação de modelos
 - Geração de código
- Integração entre ferramentas para várias etapas é importante

26

Ferramentas CASE



27