

Introdução à modelagem de software (parte 1)

Prof. Murillo G. Carneiro
FACOM/UFU

Material baseado nos slides disponibilizados pelo Prof. Ricardo Pereira e Silva (UFSC)

Objetivos

- Compreender conceitos básicos relacionados à modelagem de software
- Compreender a relevância da modelagem no processo de desenvolvimento de software
- Apresentar um breve histórico sobre a modelagem de software

O que é Software?

“Conjunto de componentes lógicos de um computador ou sistema de processamento de dados; programa, rotina ou conjunto de instruções que controlam o funcionamento de um **computador**; suporte lógico.”

fonte: Oxford dictionary

O que é Software?

Não apenas computadores, qualquer dispositivo eletrônico

Software vs e Hardware (sistemas embarcados)

Aplicativos para tablet e smartphones

Sistemas operacionais

...

O que é Engenharia de Software?

“A disciplina de Ciência da Computação voltada ao desenvolvimento de **aplicações de grande porte**. Engenharia de Software abrange **não apenas os aspectos tecnológicos** da construção de sistemas de software, mas **também aspectos gerenciais**, tais como coordenação de times de programadores, cronograma e orçamento.”

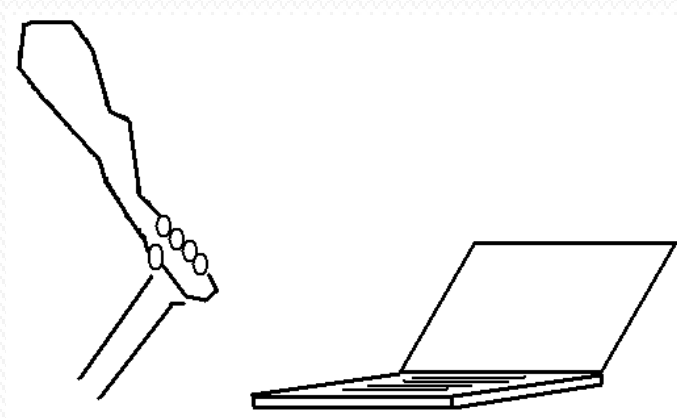
fonte: www.webopedia.com

Engenharia de Software

- Disciplina de Ciência da Computação
- Particularmente importante quando se trata alta complexidade
- Busca meios sistemáticos de desenvolvimento de software
 - Aspectos tecnológicos (ex.: análise e projeto)
 - Aspectos gerenciais (ex.: orçamento e cronograma)

Engenharia de Software

- Visa a maximização da qualidade e da produtividade na atividade de desenvolvimento de software
- Uma melhor relação custo/benefício no processo de desenvolvimento
- Em contraposição à prática primitiva de desenvolvimento



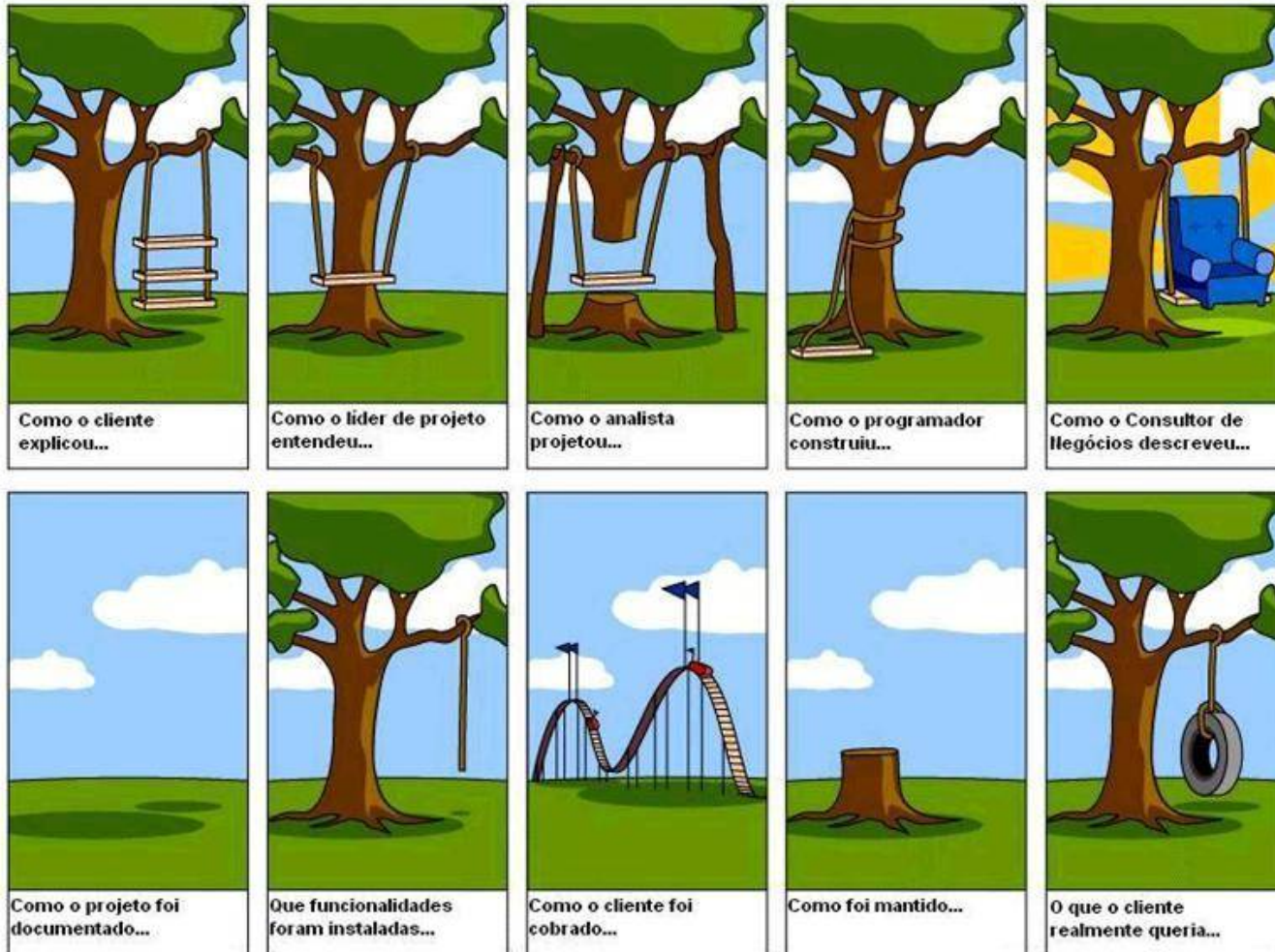
O que é Modelagem de Software?

Construção de modelos gráficos abstratos para analisar e compreender diferentes perspectivas de um sistema a ser desenvolvido

Modelagem de Software (MS)

- Modelagem prescritiva: antes do código
- Modelagem descritiva: após o código
- O mesmo conteúdo do código, mas em outro formato
- *Não seria esforço dobrado modelar e depois codificar? Não seria fazer a mesma coisa duas vezes?*

Modelagem de Software



Para que modelagem?

- Para auxiliar a concepção de uma solução
- Exemplo das engenharias:
 - Todo empreendimento complexo demanda um esforço de planejamento antes da construção
 - Exemplos: edificações, estradas, veículos, equipamentos eletrônicos etc.
 - Ninguém constrói um edifício partindo diretamente para o assentamento de tijolos (sem planejamento prévio)

Para que modelagem?

- Exemplos das engenharias mostram que situações complexas exigem tratamento diferente de situações não complexas
 - Construção de um edifício de 20 andares: necessidade de projeto prévio
 - Construção de uma casa de cachorro: martelo, pregos e mãos à obra

Para que modelagem?

- Software complexo exige planejamento antes da construção
 - Planejamento: modelagem
 - Construção: codificação
- Baixa complexidade: viável codificar direto

História da MS (1945-1960's)

- Mainframes
- Software relativamente simples
- Preocupação maior com hardware
- Fluxogramas
- Diagramas de módulos
- Decomposição funcional
- COBOL, FORTRAN, ALGOL

História da MS (1970's-1980's)

- Expansão do mercado computacional (mainframes, micros, estações de Trabalho)
- Software mais complexo
- Programação estruturada (Dijkstra, Wirth)
- Análise e projeto estruturados (Tom DeMarco, Gane & Sarson, Yourdon, Constantine)
- Diagramas de Fluxo de Dados, Pseudo-linguagem, Dicionário de Dados, Tabelas e árvores de decisão
- Pascal, C, Ada

História da MS (1980's-1995)

- Interfaces homem-máquina, redes locais, PCs, Internet
- Programação orientada a objetos
- Análise e projeto orientados a objetos (Mellor, Fusion, Coad, Yourdon, OOSE, OMT)
- Diagramas de Classes
- C++, Eiffel

História da MS (1995-)

- Internet, celular, tablets, ...
- Programação orientada a objetos atinge maturidade UML*, RUP, Metodologias ágeis
- Padrões de projeto, frameworks, componentes, linhas de produtos
- Java, C#, Python, Ruby

UML*

- Do inglês, *Unified Modeling Language*
- Linguagem desenvolvida para visualizar, especificar, construir e documentar a criação de sistemas computacionais orientados a objetos

O que aprender para modelar?

- Pontos a serem abordados:
 - Conhecer os conceitos referentes a modelagem
 - Conhecer uma linguagem de modelagem
 - Saber que passos seguir
 - Avaliar o que for produzido
 - Ênfase na descrição diagramática
 - Da modelagem para o código (e vice-versa)

Conhecer os conceitos referentes a modelagem

- Ter claro por que e para que modelar
- Noções fundamentais de Engenharia de Software
- Paradigma de orientação a objetos**
- Requisitos para uma modelagem completa

Conhecer uma linguagem de modelagem

- Linguagens de modelagem estruturada
- Linguagens de modelagem orientada a objetos**
- Os treze diagramas de UML
- Para que serve cada diagrama

Saber que passos seguir

- Desenvolvimento: sucessão de esforços que, gradualmente constroem uma solução
- Conhecer um caminho lógico para a construção da especificação
- Complemento do conhecimento de uma linguagem de especificação: saber usá-la

Avaliar o que for produzido

- Seguir etapas não garante resultado satisfatório
 - Possibilidade de escolhas inadequadas por falta de clareza ao longo do processo
 - Pequenas decisões ruins podem resultar em fracasso
- Avaliação de consistência
 - Critérios de consistência?
- Avaliação de qualidade
 - Critérios de qualidade?

Ênfase na descrição diagramática

- Explorar a expressividade dos diagramas de UML
- Em oposição a basear-se em descrição textual
 - Opção adotada em algumas metodologias
 - Vantagem das imagens sobre as palavras

Da modelagem para o código (e vice-versa)

- Modelar é um meio e não um fim
- Objetivo final é um programa que compile e execute sem erros e que cumpra os requisitos estabelecidos
- A geração de código é uma das etapas do processo
- Código gerado subsidia o aperfeiçoamento da modelagem

Considerações sobre esta aula

- Vimos conceitos básicos relacionados à MS
- Discutimos motivações importantes para MS
- Apresentamos uma breve história da MS, bem como pontos importantes a serem considerados no aprendizado de MS

Referências

Booch, G.; Jacobson, I. e Rumbauch, J. **UML: Guia do Usuário**. Campus, 2006.

Silva, R. P. **UML 2 em modelagem orientada a objetos**. Visual Books, 2007.