

Fundamentos da Análise Orientada a Objetos

Curso de Pós-Graduação *Latu Sensu* em Tecnologias
Aplicadas a Sistemas de Informação com Métodos Ágeis

Prof. M.Sc. Rodrigo Noll, PMP

Porto Alegre, 2012.

Objetivo da Aula

Apresentar conceitos relacionados
com a evolução dos processos de
desenvolvimento de software

Material da disciplina



- <http://bit.ly/JsgsnR>

Discussão



- O que é software?
 - Programa de software + documentação
- O que é Engenharia de Software (ES)?
 - Disciplina de engenharia associada a todos aspectos da produção de software
- Qual a diferença entre ES e Ciência da Computação?
 - CC = teoria e fundamentos; ES = desenv.de software
- O que é um processo de software?
 - Conjunto de atividades e resultados visando desenvolver um produto
- O que é um modelo de um processo de software?
 - Representação simplificada de um processo

Discussão

- Quais as dificuldades em se construir um software? Quais os problemas mais comuns?
- O que faz um analista de sistemas?
- Como podemos entender o que o cliente quer?
- Como validar o que o cliente quer?

Parte 1

- Engenharia e processos de desenvolvimento de software

Software está em todos os lugares...



Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

7

Problemas comuns

- ☐ Dificil manutenção
- ☐ Documentação pobre
- ☐ Problemas na qualidade
- ☐ Custo e prazos sempre ultrapassam
- ☐ Não adequação as necessidades do usuário

Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

8

Quão graves podem ser os problemas?

- Ariane 5, voo 501
 - Primeiro teste de voo em 04/06/96
 - Foguete se autodestruuiu após 37s por mal funcionamento no controle de SW: **conversão de float de 64 bits para inteiro de 16 bits** (começou a sair do curso e foi destruído)
 - O bug mais caro da história
 - Continha quatro satélites: o custo foi de U\$ 500 milhões



Ariane-5 — definição de qualidade

- Do relatório de Lions:
 - “... a documentação demonstrou a alta qualidade do programa Ariane-5 com relação ao trabalho dos engenheiros em geral e também ao grau de perfeição e rastreabilidade dos documentos”.
 - “... o fornecedor do SRI ... estava apenas seguindo as especificações que lhe foram dadas. ... A exceção foi detectada mas tratada de maneira não apropriada, porque, segundo o enfoque adotado, o software deveria ser considerado correto até que fosse evidenciado um defeito”.

Crise de Software

Crise de Software

- Termo utilizado durante as etapas iniciais de desenvolvimento de software:
 - 1968, NATO Software Engineering Conference;
 - Antes de definição de processos e metodologias.
- As origens da crise de software referem a complexidade, expectativa e mudanças;
- Geralmente, projetos de software são grandes, complexos, mal especificados, relacionam aspectos não familiares e são vulneráveis a problemas não previstos.

Crise de Software

- Desenvolvimento de software é um contínuo estado de crise:
 - Dificuldade de escrever, compreender e verificar programas computacionais;
 - Projetos ultrapassam orçamentos e cronograma;
 - Atender requisitos e satisfazer usuário (qualidade).

- "Software é hard" (Donald Knuth)
 - Não existe bala de prata, isto é, uma única abordagem para prevenir casos de falha.

Crise de Software

- "Construímos sistemas como os irmãos Wright construíam aviões - constrói-se de uma vez só, empurra-se para o despenhadeiro, deixa bater e começa tudo outra vez."

(NAUR & RANDELL, 1968) Software Engineering: Report on a Conference Sponsored by the NATO Science Committee.

Crise de Software



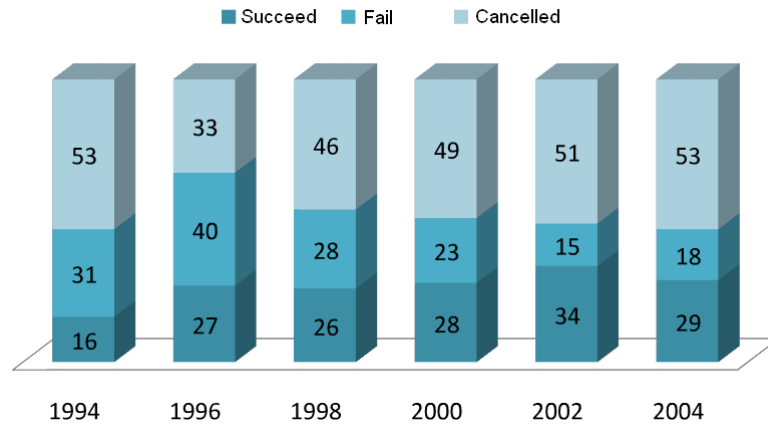
Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

15



Evidências da Crise de Software

Chaos Report

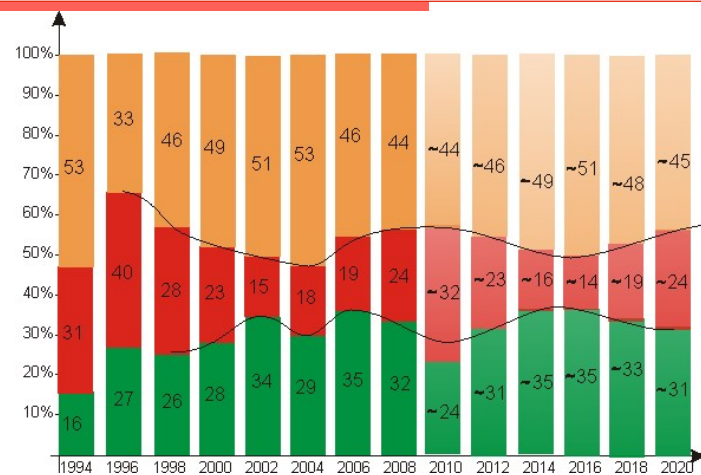


THE STANDISH GROUP REPORT CHAOS

Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

17

Chaos Report

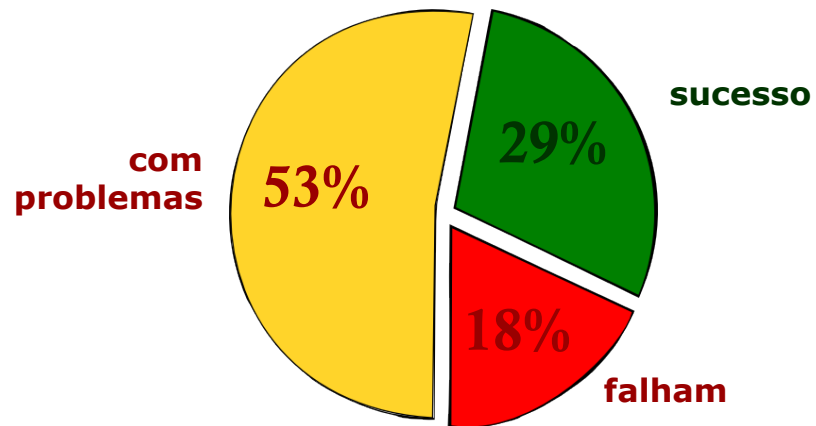


Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

18

Chaos Report

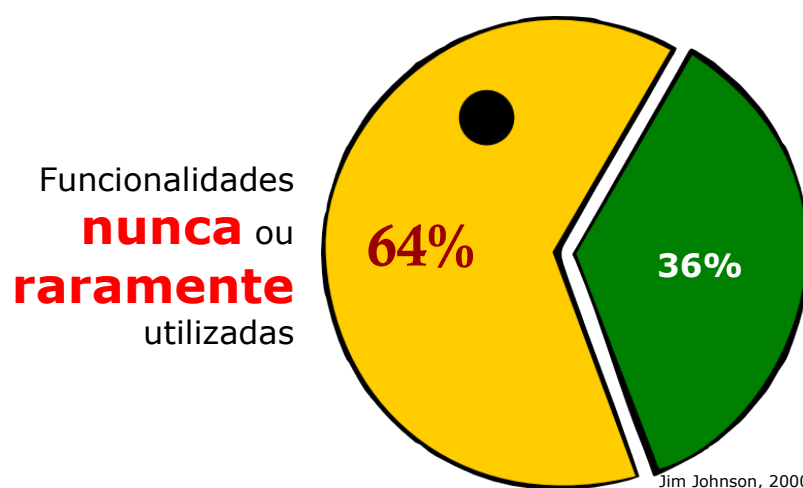
- Resultado dos projetos (2004):



Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

19

Desperdício das Funcionalidades



Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

Jim Johnson, 2000

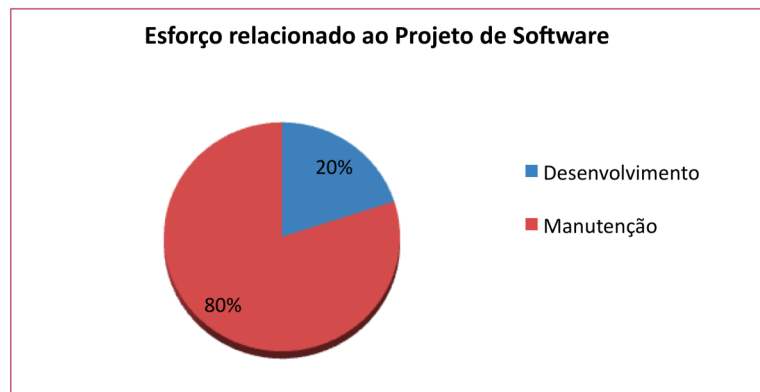
20

Esforços para o desenvolvimento de software

Desenvolvimento e Manutenção de Software

- Desenvolvimento (20%):
 - Início: quando necessidade do produto é identificada;
 - Fim: quando teste do produto implantado é concluído e o produto é entregue para a operação/produção.
- Manutenção(80%):
 - Todas as atividades após a entrega:
 - Aumento da capacidade do produto (60%)
 - Adaptação do produto a novos ambientes (20%)
 - Correção de erros (20%)

Desenvolvimento e Manutenção de Software



Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

23

Desenvolvimento e Manutenção de Software

- Por exemplo, uma pesquisa de Fjeldstad e Hamlen (1979) verificou que 25 empresas gastavam:
 - 31% do seu esforço em desenvolvimento
 - 61% em manutenção (incluindo suporte a usuário).
- Outra pesquisa (Parikh e Zvegintzov, 1983) verificou que a manutenção é cerca de 3 vezes maior que o desenvolvimento.
- Recentemente, as pesquisas estimam que 20% do esforço das empresas é gasto em desenvolvimento e 80% em manutenção.

Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

24

Desenvolvimento e Manutenção de Software



- Manutenção é cara e inevitável pois:
 - Software legado (mais de 5 anos de uso);
 - Implantação é incompleta, necessitando ajustes;
 - Mudanças nos sistemas, refletindo a mudança de negócio da empresa.
- Como diminuir?
 - Melhoria da qualidade do software
 - **Melhoria do processo de produção de SW**
- Ambas associadas à uma mudança de cultura do desenvolvimento do software

Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

25

Mudança de Cultura de Desenvolvimento de Software



- Barateamento do HW;
- Fazer Software ainda é atividade intelectual:
 - Mão de obra intensiva (\$)
 - Baseada em experiência
 - Baixos níveis de reuso e automação
- Usuários mais exigentes: qualidade, escopo e prazo (triângulo)
- Software mais complexo: metodologias tradicionais já não servem.

Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

26

Problemas com o desenvolvimento de software: Natureza do SW



□ Dificuldade:

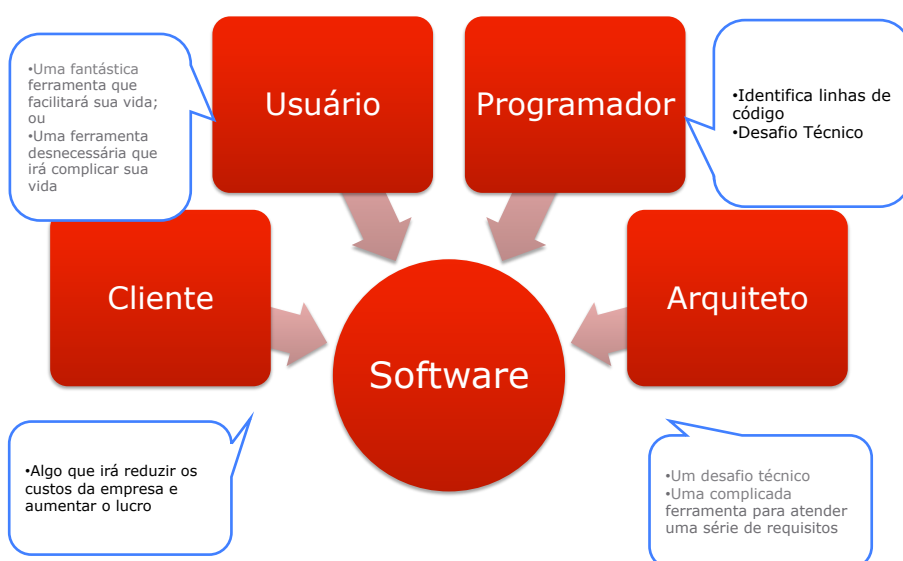
- **NÃO há teoria** subjacente (motor elétrico: equações de tensão, potência, corrente, etc.)
- **NÃO existem equações**, fórmulas para se desenvolver software, como nas demais engenharias
- É um componente de um sistema maior que **combina hardware, pessoas e dispositivos mecânicos**
- Possui uma **estrutura complexa**
- Extremamente **maleável**
- Construído por humanos, **trabalho intelectual**

Desenvolver software é um processo criativo!

Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.

27

Problemas com o desenvolvimento de software: Fator Interdisciplinar



Problemas com o desenvolvimento de software



- Mas existem características análogas aos processos de engenharia física:
 - Identificação de pontos de similaridade (métricas)
 - Sugestões de simulações de abordagens bem sucedidas (benchmarks)
 - Evitar erros conhecidos

- Na prática, enfoque de Engenharia para sistematizar as atividades de:
 - Entender claramente o problema que se quer resolver
 - Desenvolver ferramentas e técnicas para resolvê-lo
 - Gerenciar equipe para resolvê-lo
 - Aspectos Tecnológicos e Gerenciais

*Fundamentos da Análise Orientada a Objetos
Prof. Rodrigo Noll, 2012.*

29



ENGENHARIA DE SOFTWARE

- Engenharia:
 - É a arte, a ciência e a técnica de bem conjugar os conhecimentos especializados (científicos) de uma dada área do saber com a sua viabilidade técnico-econômica, para produzir novas utilidades e/ou transformar a natureza, em conformidade com idéias bem planejadas.

- “Aplicação prática do conhecimento científico para o projeto e a construção de programas computacionais e a documentação necessária à sua operação e manutenção”. (Boehm 1976)
- “Engenharia de Software é a área interdisciplinar que engloba vertentes tecnológica e gerencial, visando abordar, de modo sistemático, os processo de construção, implantação e manutenção de produtos de software com qualidade assegurada por construção, segundo cronogramas e custos previamente definidos.”(Bruno Maffeo)

Engenharia de Software



- Engenharia de Software
 - “Abrange um conjunto de três elementos fundamentais - métodos, ferramentas e procedimentos – que possibilita ao gerente controlar o processo de desenvolvimento de software e oferece ao profissional uma base para a construção produtiva de software de alta qualidade”. (Pressman, 1995)

Engenharia de Software Definição simplificada



- Engenharia de Software
 - O estudo e a aplicação de metodologias para desenvolver/ manter software de qualidade e que atendam as expectativas do cliente.
- O objetivo é produzir software:
 - Confiável: não quebra
 - Completo: documentação e conforme expectativa do cliente
 - Desenvolvido no Prazo (entregue na data acordada)

Analogias com desenvolvimento/engenharia de software

O que é desenvolvimento de software?

- David Gries says [writing software is a science](#) (1981)
- Donald Knuth says [it's an art](#) (1998)
- Watts Humphrey says [it's a process](#) (1989)
- Kent Beck say it's like [driving a car](#) (2000)
- Alistair Cockburn says [it's a game](#) (2002) or like [Rock Climbing](#)
- Eric Raymond says it's [like a bazaar](#) (2000)
- Andy Hunt and Dave Thomas say [it's like gardening](#)
- Paul Heckel says [it's like filming Snow White and the Seven Dwarfs](#) (1994)
- Fred Brooks says that it's like farming, hunting werewolves or drowning with dinosaurs in a tar pit (1995)

O que é desenvolvimento de software?



- Steve McConnell says [its like Construction](#): The whole notation of software architecture, blue prints and modeling comes from here.
- Naresh Jain believes [Software Development is like Trekking/Hiking](#)
- Some people say its like Craft/Martial Arts (more on learning aspects)
- Others say its like Book Writing/Composition/Poetry. Software development has an improvisational aspect
- [Software as oyster farming](#). I'm really interested from the system accretion aspect. Accretion is closely related to "incremental", "adaptive", and "evolutionary"
- And then of course we have the manufacturing analogy. [Lean Manufacturing](#) is more meaningful than assembly line manufacturing.
- I also like the Painting analogy. Demonstrates iterative and incremental thinking very well.
- Fonte: Naresh Jain from [Managed Chaos](#)

Interdisciplinaridade da Engenharia de SW



- Ciência da Computação
- Administração/Gerência de Projetos
- Comunicação
- Técnica de Solução de Problemas

Perguntas?

Exercício

Responder as seguintes questões sobre o OpenUP:

<http://epf.eclipse.org/wikis/openup/>

1. Quais são as 4 fases definidas pelo OpenUP?
2. Quais são as disciplinas sugeridas?
3. Qual o propósito da disciplina de requisitos?
4. O que são os passos para se identificar e evidenciar os requisitos?
5. Apresente a orientação técnica (guideline) para detalhar casos de uso e cenários.
6. Explique o que significa um Modelo de Caso de Uso.
7. Como se projeta uma solução a partir dos requisitos?
8. O que é um padrão Entity-Control-Boundary? (identifique a guideline)