




Laboratório de Engenharia de Software

# INF1413 Teste de Software Apresentação 2015/1

[www.inf.puc-rio.br/~inf1413](http://www.inf.puc-rio.br/~inf1413)

Arndt von Staa  
LES/DI/PUC-Rio  
Fevereiro 2015



## Sumário

Laboratório de Engenharia de Software

- Quem sou eu?
- Um pouco da história da computação na PUC-Rio
- Objetivo da disciplina
- Qual é o problema abordado na disciplina?
- Qual é o objetivo da disciplina
- Organização

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

2

Quem sou eu?



Laboratório de Engenharia de Software


- Arndt von Staa
  - [arndt \\*at\\* inf.puc-rio.br](mailto:arndt*at*inf.puc-rio.br)
  - Sala RDC 420
  - ramal DI 4333

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

3

Quem sou eu?



Laboratório de Engenharia de Software

- Iniciei atividades em computação em **setembro de 1962**
  - CPD - Centro de Processamento de Dados da PUC
- Professor desde setembro 1967
- “Sócio fundador” do Departamento de Informática
  - criado no final de 1967
  - cursos regulares com uso intensivo de computador começaram em março de 1968
    - ICC (1º. no Brasil), Cálculo Numérico (1º. ??? no Brasil)
    - Mestrado (1º. no Brasil)
- PhD em Ciência da Computação (*Computer Science*) pela University of Waterloo, Canadá, 1974
- Interesse de pesquisa: como **desenvolver e manter** software possuindo qualidade assegurada

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

4

Laboratório de Engenharia de Software

LES

Um pouco de história da computação na PUC-Rio

Fev/2015Arndt von Staa © LES/DI/PUC-Rio5

“[...]E aí se iniciaram as negociações entre as partes envolvidas, principalmente a PUC e a Burroughs. Não é fácil a gente se deslocar para 1958 e 1959, ou seja, falar em computador naquela época era um completo absurdo. [...] todo mundo dizia que o Brasil certamente já não estava pronto para entrar na era da computação eletrônica por várias razões: não havia experiência, não havia outros computadores, não havia analistas, não havia programadores e por não haver técnicos de manutenção.[...] Então se teve a idéia de se formar um consórcio de várias entidades que pudessem utilizar esses computadores.[...]

[...] Depois de grandes batalhas finalmente em 1959 se conseguiu todas as licenças necessárias, [...] e se teve umas medidas para importar esse computador da Califórnia. [...] Aí veio o grande problema de como transportá-lo. [...] Então nós alugamos um DC6 da Panamerican na época para transportar da Califórnia para o Rio de Janeiro[...] e o computador foi transportado do Galeão até a PUC no caminhão aberto do Exército. Eu me lembro disso. Eu fui o primeiro que vi tudo isso.”

Georges Herz - 25/10/2007

Fev/2015Arndt von Staa © LES/DI/PUC-Rio6

## 1o. Computador no Brasil – na PUC-Rio



- Primeiro computador no Brasil: Burroughs Datatron B-205
  - negociação iniciada em 1958
  - instalado em 1960 no CPD da PUC-Rio
  - tinha mais de 1600 válvulas eletrônicas, maioria duplo triodo
  - pesava mais de uma tonelada
  - consumia em torno de 30 KVA,
  - requeria em torno de 101.000 BTU/hora de refrigeração
    - mais ou menos 25 KVA
  - velocidade média 2,5 ms (**mili** segundo ... )
  - acessava memória em +- 1ms ou em 0,1ms
    - dependendo de como se programava
  - memória 4000 palavras de 10 dígitos decimais (BCD)
    - dá cerca de 20 kbytes
    - memória: tambor magnético
  - não tinha sistema operacional, compilador, assembler, ...
    - programação era feita em linguagem de máquina absoluta

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

7

## 1o. Computador no Brasil



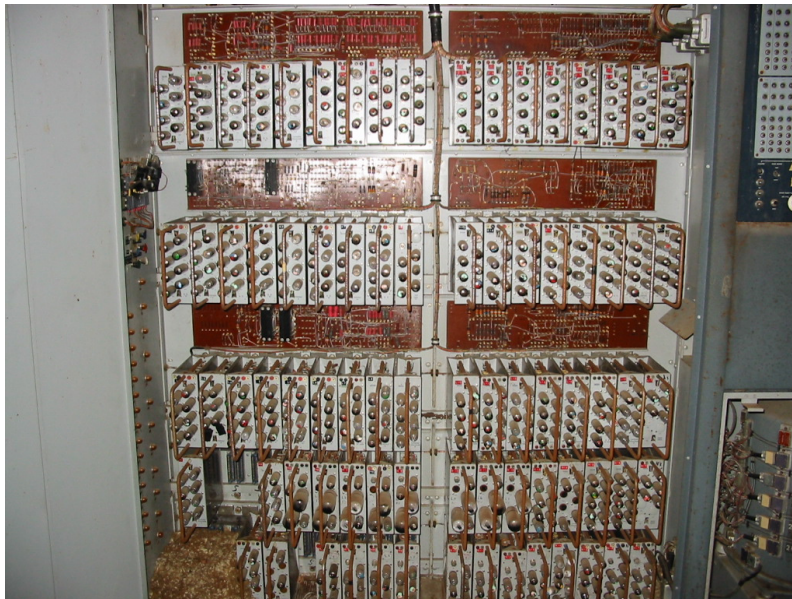
Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

8



## Um computador a válvula



Laboratório de Engenharia de Software

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

10



## 1o. Computador no Brasil



- Alguns trabalhos realizados no B205
  - Simulação da operação de reservatórios de usinas hidroelétricas
    - cerca de 3200 comandos em linguagem de máquina absoluta
  - Simulação do despacho de carga na rede de transmissão entre usinas e centros de consumo
    - adaptação de programa americano
  - Desenvolvimento de utilitários
  - Manutenção do hardware
    - adição de três novas instruções ao hardware
  - Programação de painel de tabuladora IBM407 *a seguir*
  - Cálculo estrutural da Ponte da Amizade em Foz do Iguaçu
  - Cálculo da estabilidade de barragens de terra
  - . . .

Fev/2015

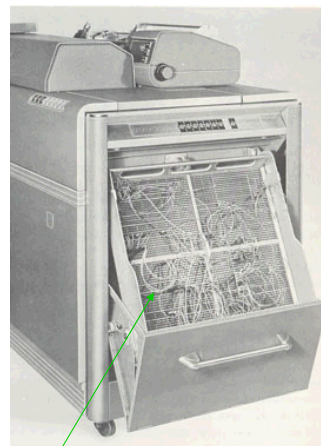
Arndt von Staa © LES/DI/PUC-Rio

11

## 1o. Computador, saída: tabuladora IBM 407



- Isso foi uma impressora de 150 linhas por minuto
  - programável por meio de um painel
  - eletromecânica – circuitos: relés



Isso é o programa!

Um programa físico

Fev/2015

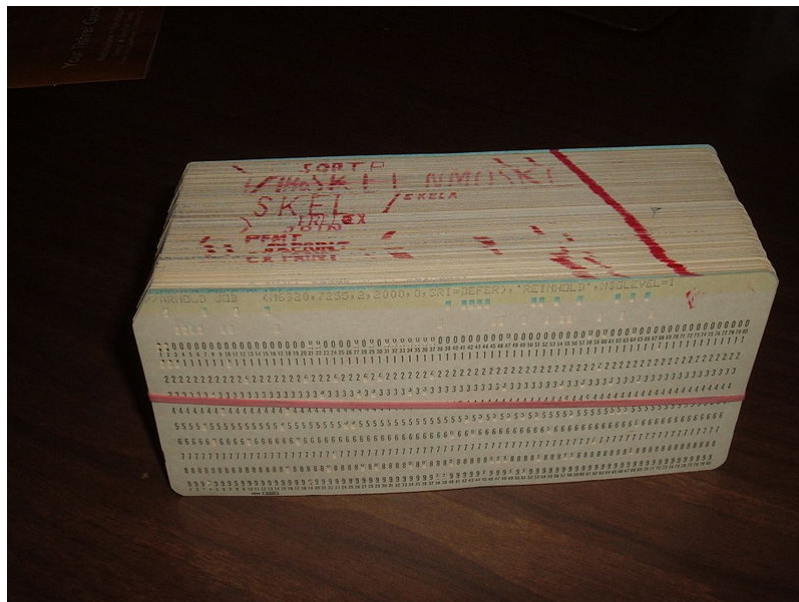
Arndt von Staa © LES/DI/PUC-Rio

12

## Programas medidos em cm



Laboratório de Engenharia de Software



Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

13

## Programas medidos em cm



Laboratório de Engenharia de Software



Bundesarchiv, B 145 B64 F021434-0006  
Foto: Gutfreund, Jens 12. März 1970

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

14

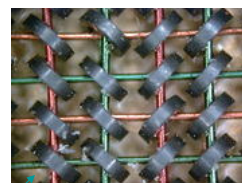
## Programas medidos em cm, ai ai ai



## Outros computadores instalados na PUC



- Burroughs B200 (1964)
  - circuitos de componentes discretos → transistores, ...
  - tinha assembler, mas não tinha sistema operacional
  - 4096 bytes de memória, mais tarde 8192 bytes
    - núcleo de ferrite
  - 4 unidades de fita magnética
  - alguns trabalhos realizados
    - folha de pagamentos de um grande estaleiro



cada toro corresponde a um bit!  
as memórias eram "tecidas" a mão



## Outros computadores instalados na PUC



- IBM 1130 (1966)
  - uma espécie de computador pessoal?
  - circuitos de baixa integração  $2,2 \mu s$ , **16k** memória, disco de **1Mbyte**
  - tinha um sistema operacional monoprogramado rudimentar: DUP
  - Fortran, Cobol, Algol, Assembler
  - utilizado para dar uma disciplina eletiva de cálculo numérico
  - alguns trabalhos realizados por mim
    - jogo da velha *com aprendizado*



Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

17

## 1º. computador usado em disciplinas obrigatórias



- IBM 7044
- Parentes
  - IBM 704
    - usado para desenvolver o 1º. FORTRAN
  - IBM 7040
  - IBM 7090
  - IBM 7094




Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

18

**1º. computador usado em disciplinas obrigatórias**




Laboratório de Engenharia de Software

- IBM 7044 (instalado no final de 1967)
  - circuitos de componentes discretos
  - na época o computador mais potente no país
  - velocidade em torno de 5 micro segundos
  - memória 32 k palavras de 36bits, dá cerca de 128 kbytes
    - núcleo de ferrite
  - entrada e saída (cartões, impressão) era via um IBM 1401
  - sistema operacional IBSYS monoprogramado
  - Fortran, COBOL, Macro-Assembler
  - alguns trabalhos realizados
    - alterei o sistema operacional para viabilizar o uso do computador por alunos de graduação
    - compilador de compiladores
    - processador de expressões algébricas
    - simulador do computador PUC-007 (versão simplificada do B-205)
    - sistema de gerência acadêmica para a DAR

Fev/2015
Arndt von Staa © LES/DI/PUC-Rio
19

**Outros sistemas instalados na PUC**



Laboratório de Engenharia de Software

- IBM 370/165 (a partir de 1972)
- CDC
- IBM
- Micros isolados no DI
- Micros interligados via internet
  - hoje o DI conta com mais de 700 máquinas, back bone 1 giga
- Alguns sistemas desenvolvidos
  - Formataadores de textos (IBM 360 ; IBM 370)
    - ainda não existiam Word e similares
  - Editor de projetos estruturados Mosaico
  - Meta-ambiente de engenharia de software Talisman
  - Arcabouços de apoio ao teste automatizado
  - . . .

Fev/2015
Arndt von Staa © LES/DI/PUC-Rio
20


## Por que mencionar tudo isso?




- Houve uma época em que **sem raciocínio cuidadoso não se desenvolvia software** capaz de funcionar corretamente
  - não existiam *debuggers*
  - compiladores eram muito lentos
    - o tempo de retorno de uma submissão (compilação, execução de testes, ...) era medido em horas quando não em dias
    - desenvolvimento baseado em tentativa e erro era garantia de projetos fracassados
- Mesmo hoje em dia **desenvolvedores de sucesso** fazem uso de reflexão baseada em lógica
  - ideal: **técnicas formais leves**


## Descrição da disciplina



Laboratório de Engenharia de Software	<h2>Objetivos da disciplina</h2>	
	<ul style="list-style-type: none"> <li>• Habilitar o aluno a aplicar com eficácia conceitos de controle da qualidade de software             <ul style="list-style-type: none"> <li>– visando as atividades: verificação, validação e aceitação;</li> <li>– envolvendo as técnicas: inspeção, teste convencional, teste automatizado, técnicas formais leves.</li> </ul> </li> <li>• Habilitar o aluno a organizar e gerenciar o processo de controle e garantia da qualidade de software.</li> </ul>	
	Fev/2015	Arndt von Staa © LES/DI/PUC-Rio 23

Laboratório de Engenharia de Software	<h2>Por que ...?</h2>	
	<ul style="list-style-type: none"> <li>• Por que testar?</li> <li>• Por que testes não conseguem identificar todos os defeitos?</li> <li>• Por que dá tanto trabalho?</li> <li>• Por que estudar isso?             <ul style="list-style-type: none"> <li>– qualquer criança sabe testar...</li> <li>– sabe mesmo?                 <ul style="list-style-type: none"> <li>• você voaria num avião cujo software foi testado por uma criança?</li> <li>• você acreditaria num sistema de matrícula testado de forma pouco cuidadosa?</li> </ul> </li> </ul> </li> <li>• Por que testar é desafio intelectual?</li> </ul>	
	Fev/2015	Arndt von Staa © LES/DI/PUC-Rio 24

**Problema abordado na disciplina INF1413**



Laboratório de Engenharia de Software

**Como assegurar qualidade satisfatória do software?**


- em primeiro lugar desenvolver e mantê-lo de modo que contenha muito poucos defeitos já desde o início
  - aproximar-se o mais possível de correto por construção
    - ideal → zero defeitos, por enquanto uma utopia
- definir precisamente o que é esperado
  - especificar
- verificar continuamente – desde a primeira especificação até a descontinuação –
  - a satisfação de
    - desejos e expectativas do usuário
    - requisitos da interface humano computador
    - requisitos funcionais
    - requisitos não funcionais
    - arquitetura e projeto
    - correteza dos procedimentos (algoritmos)

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

25

**Problema abordado na disciplina INF1413**



Laboratório de Engenharia de Software

**Como controlar a qualidade do software?**


- saber raciocinar sobre artefatos
- saber ler e criticar construtivamente os diversos artefatos
  - inspeções
- testar de forma sistemática
  - formular casos de teste que efetivamente testem algo relevante
- automatizar os testes
- utilizar testes como técnicas de apoio ao desenvolvimento
  - estabelecer uma estratégia de teste
  - ao especificar sempre perguntar: como posso testar isso?
- utilizar instrumentos como mecanismos de apoio ao desenvolvimento e de controle continuado da execução
  - técnicas formais leves
- desenvolver visando manutenibilidade
  - subsistema de apoio à manutenção


Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

26




Laboratório de Engenharia de Software	<b>Objetivo do curso, revisitado</b>		
	<ul style="list-style-type: none"> <li>• Espera-se que, ao concluir a disciplina INF1413, o aluno           <ul style="list-style-type: none"> <li>– esteja capacitado a usar e adaptar, de <b>forma racional</b>, técnicas eficazes e eficientes de <b>controle da qualidade de software</b></li> <li>– entenda que assegurar a qualidade do software é uma <b>propriedade sistêmica</b> envolvendo, além de técnicas de controle da qualidade, o entendimento das necessidades do usuário (serviço a prestar), especificação, arquitetura, design e codificação todos realizados visando qualidade</li> <li>– esteja habilitado a aplicar os principais conceitos de controle da qualidade:               <ul style="list-style-type: none"> <li>• verificação, validação e aceitação de software</li> </ul> </li> <li>– tenha conhecimento de como organizar e gerenciar o processo de controle e garantia da qualidade de software</li> </ul> </li> </ul>		
	<p><b>Sistema:</b> conjunto de elementos interdependentes que visam o alcance de um objetivo comum</p> <p><b>Propriedade sistêmica:</b> um objetivo de um sistema</p>		
	Fev/2015	Arndt von Staa © LES/DI/PUC-Rio	27

Laboratório de Engenharia de Software	<b>Pré-requisitos esperados</b>		
	<ul style="list-style-type: none"> <li>• INF1301 Programação Modular ou INF1628 Programação em Ponto Grande           <ul style="list-style-type: none"> <li>– existe propositalmente uma pequena superposição com assuntos estudados em INF1301 / INF1628</li> </ul> </li> <li>• Alguma prática em programação orientada a objetos</li> <li>• Saber utilizar ferramentas de desenvolvimento           <ul style="list-style-type: none"> <li>– Eclipse (Java) e/ou Visual Studio (C / C++), ou outro qualquer               <ul style="list-style-type: none"> <li>• os trabalhos que envolvam programação utilizarão C++</li> </ul> </li> <li>– janela de linha de comando</li> <li>– “batch files”, muito melhor: Lua</li> </ul> </li> </ul>		
	Fev/2015	Arndt von Staa © LES/DI/PUC-Rio	28

Laboratório de Engenharia de Software

## Livro texto

- Pezzè, M.; Young, M.; *Teste e Análise de Software*; Porto Alegre, RS: Bookman; 2008
- Cockburn, A.; *Escrevendo Casos de Uso Eficazes - Um Guia para Desenvolvedores de Software*; São Paulo, SP: Bookman; 2005
- Notas de aula
- Enorme lista de textos complementares
  - vou utilizar material desses livros e artigos em adição ao livro texto
  - não espero que os alunos leiam todos ☺
- O documento descritivo da disciplina – *Apresentação da disciplina* – encontra-se no site da disciplina
  - contém o plano de aulas tentativo



Fev/2015
Arndt von Staa © LES/DI/PUC-Rio
29

Laboratório de Engenharia de Software


## Critério de aprovação

- 2 provas, com consulta
  - P1: quarta      29/abril
  - P2: segunda    22/junho
- 4 trabalhos, prazos
  - T1: segunda    13/abril
  - T2: quarta     29/abril
  - T3: quarta     27/maio
  - T4: segunda    22/junho
- Cálculo da nota final
  - $G1 = (P1 * 2. + T1 + T2) / 4.$
  - $G2 = (P2 * 2. + T3 + T4) / 4.$
  - **GrauFinal** = **if** (  $G2 \geq 3.$  ) **then** (  $G1 + G2$  ) / 2.  
    **else** (  $G1 + 3. * G2$  ) / 4. **fi**

Datas, horários e local das provas e dos trabalhos podem mudar. Consultem periodicamente a página de avisos.

Fev/2015
Arndt von Staa © LES/DI/PUC-Rio
30

## Plano de aulas



Laboratório de Engenharia de Software


- O plano de aulas encontra-se no site da disciplina, no documento *Apresentação da disciplina*
- Os dias em que não haverá aula estão assinalados

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

31

## CrITÉrios de correção dos trabalhos



Laboratório de Engenharia de Software


- Levem em consideração os critérios de avaliação dos trabalhos
  - estão anexados à descrição da disciplina
- Devem ser realizados em grupo de 2 ou 3 alunos
  - procurem organizar os grupos **ainda hoje!**
- Devem gerar documentos, por exemplo
  - descrição de como foi feito o controle da qualidade
  - o laudo do controle
  - se for o caso, o artefato utilizado para realizar o controle
  - devem ser enviados em um único arquivo **.zip**
  - se o provedor de acesso não permitir enviar arquivos **.exe** ou similares, rebatizem esses arquivos adicionando um **'x'** à extensão, exemplos: **abc.exex** , **cde.batx**

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

32

**Organização**



Laboratório de Engenharia de Software


- Toda a comunicação deverá ser feita preferencialmente de forma eletrônica
  - e-mail: [arndt \\*at\\* inf.puc-rio.br](mailto:arndt*at*inf.puc-rio.br)
- Se necessário, podem me procurar na sala RDC420

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

33

**Organização**



Laboratório de Engenharia de Software

- Site da disciplina: [www.inf.puc-rio.br/~inf1413](http://www.inf.puc-rio.br/~inf1413)
  - descrição da disciplina, plano de aulas
  - notas de aula
  - avisos diversos
    - datas das provas
  - enunciados dos trabalhos
  - software para download
  - documentos para download

Fev/2015

Arndt von Staa © LES/DI/PUC-Rio

34

## Convivência



- Pontualidade
  - inicio as aulas às 17:10
  - se não tiver pelo menos 3 alunos em sala, não dou aula e considero a matéria dada
- Eu tenho dificuldade auditiva
  - requer um pouco de paciência por parte dos interlocutores
  - meu entendimento do que está sendo falado depende de saber qual é o contexto do assunto falado
- Apesar do problema de audição, espero que sejam feitas perguntas
  - aprendizado requer participação, mais precisamente: diálogo

Perguntas?