

Quem sou eu?



- Arndt von Staa
 - arndt *at* inf.puc-rio.br
 - Sala RDC 420
 - ramal DI 4333

Ago 201

rndt von Staa © LES/DI/PUC-Ric

Quem sou eu?

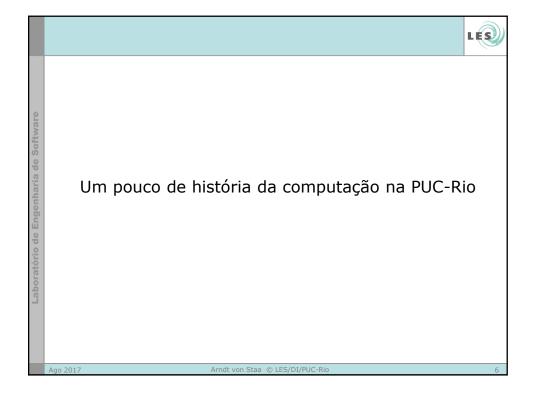


- Iniciei atividades em computação em setembro de 1962
 - CPD Centro de Processamento de Dados da PUC
- Sou professor na PUC-Rio desde setembro 1967
- Sou "sócio fundador" do Departamento de Informática
 - criado no final de 1967
 - cursos regulares obrigatórios com uso intensivo de computador começaram no CTC/PUC-Rio em março de 1968
 - ICC (1º. no Brasil), Cálculo Numérico (1º. ??? no Brasil)
 - Mestrado (1º. no Brasil)
- Engenheiro mecânico, EPUC 1965.
- PhD em Ciência da Computação (Computer Science) pela University of Waterloo, Canadá, 1974

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio

Quem sou eu? • Interesse de pesquisa: como desenvolver e manter software possuindo qualidade assegurada - arquitetura - processos - teste - auto-verificação - em tempo de execução



"[...]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

10. Computador no Brasil - na PUC-Rio



- Primeiro computador no Brasil: Burroughs Datatron B-205
 - negociação iniciada em 1958
 - instalado em 1960 no então CPD da PUC-Rio
 - tinha mais de 1600 válvulas eletrônicas, a maioria duplo triodo
 - pesava mais de uma tonelada
 - consumia em torno de 30 KVA,

equivalente a mais de 70 residências...

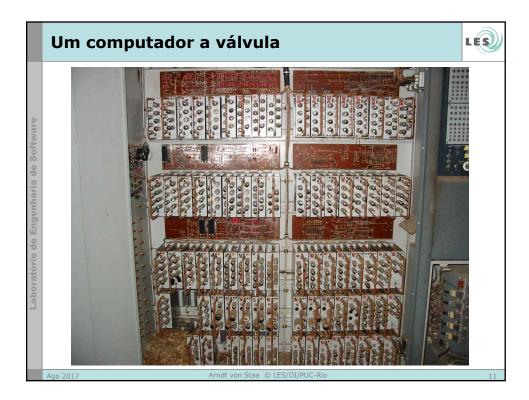
- $-\,$ requeria em torno de 101.000 BTU/hora de refrigeração
 - mais ou menos 25 KVA
- velocidade média 2,5 ms (mili segundo ...)
- acessava memória em +- 1ms (0,001 mips) ou em 0,1ms
 - dependendo de como se programava (atuais ≈ 200.000 mips, Wikipedia)
- memória 4000 palavras de 10 dígitos decimais (BCD)
 - dá cerca de 20 kbytes
 - memória: tambor magnético
- não tinha sistema operacional, compilador, assembler, ...
 - programação era feita em linguagem de máquina absoluta

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio







10. Computador no Brasil



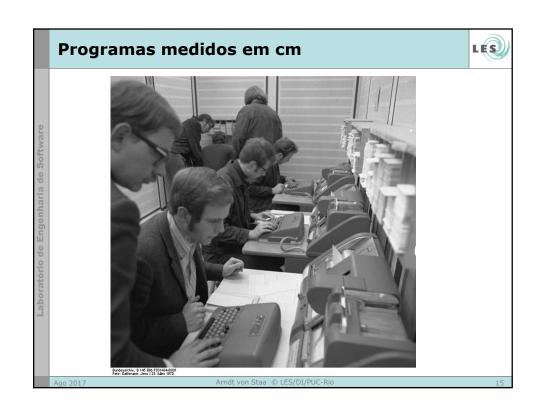
- Alguns trabalhos realizados no B205
 - Simulação da operação de reservatórios de usinas hidroelétricas
 - cerca de 3200 comandos em linguagem de máquina, +- 6 meses
 - um programa C requereria cerca de 800 instruções...
 - Simulação do despacho de carga na rede de transmissão entre usinas e centros de consumo
 - adaptação de programa americano
 - Cálculo estrutural da Ponte da Amizade em Foz do Iguaçu
 - Cálculo da estabilidade de barragens de terra
 - . . .
 - Manutenção do hardware
 - adição de três novas instruções ao hardware
 - Programação de painel de tabuladora IBM407 a segui

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio









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



1º. computador usado em disciplinas obrigatórias



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



Ago 2017

Arndt von Staa © LES/DI/PUC-Rio

1º. computador usado em disciplinas obrigatórias



- 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 (≈ 0,2 mips)
 - 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

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio

Outros sistemas instalados na PUC



- 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
 - Formatadores 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
 - . . .

Ago 2017

Arndt von Staa © LES/DI/PUC-Ric

21

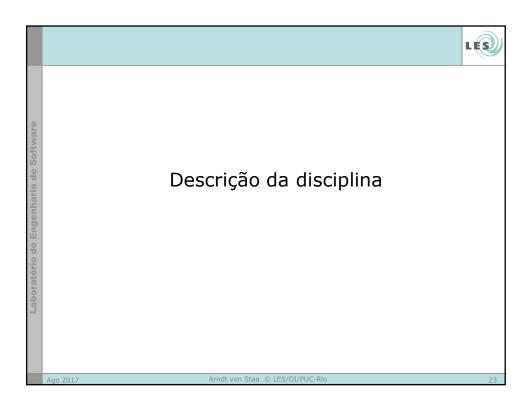
Por que mencionar tudo isso?



- 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

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio



Objetivos da disciplina



- Habilitar o aluno a aplicar com eficácia conceitos e métodos de controle da qualidade de software
 - visando as atividades: verificação, validação e aceitação;
 - envolvendo as técnicas:
 - leitura, inspeção, análise estática,
 - especificação formalizada, técnicas formais leves,
 - · teste convencional, teste automatizado,
 - medição estática e medição dinâmica.
- 2. Habilitar o aluno a organizar e gerenciar o processo de controle e garantia da qualidade de software.

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio

Problema abordado na disciplina INF1413



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
 - corretude dos procedimentos (algoritmos)

Ago 2017

Arndt von Staa © LES/DI/PUC-Ric

2.5

Problema abordado na disciplina INF1413



Como controlar a qualidade do software?

- saber raciocinar sobre artefatos
- ler e criticar construtivamente os diversos artefatos
 - revisões e inspeções
- testar de forma sistemática
 - formular casos de teste que testem algo relevante
- automatizar os testes sempre que possível
- 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

Ago 2017

Arndt von Staa © LES/DI/PUC-Rid

Problema abordado na disciplina INF1413



- Um sistema estará correto se satisfizer plenamente a sua especificação para todas as possíveis formas de uso
 - mas o controle da qualidade jamais poderá será completo
- Precisa-se dispor de especificações completas e corretas
 - requisitos funcionais, não funcionais, e inversos
- Mas especificações podem estar, ou tornar-se, erradas
 - inadequadas: não correspondem a o que o usuário efetivamente necessita
 - incompletas, ou erradas, ou baseadas em teorias incorretas
 - manutenção
- Logo podemos ter como resultado:
 - sistemas corretamente incorretos (epa!)

Ago 2017

Arndt von Staa © LES/DI/PUC-Ric

27

Por que ...?



- Por que testar?
- Por que testes n\u00e3o conseguem identificar todos os defeitos?
- Por que dá tanto trabalho?
- Por que estudar isso?
 - qualquer criança sabe testar...
 - sabe mesmo?
 - você acreditaria num sistema de matrícula testado de forma pouco cuidadosa?
 - você voaria num avião cujo software foi testado por uma criança?
 - você acreditaria em um sistema de controle da distribuição instantânea de energia elétrica que não fosse capaz de fazê-lo 24 horas por dia para todos os dias da sua instalação em diante?
- Testar é desafio intelectual, por que?

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio

Objetivo do curso, revisitado



- Espera-se que, ao concluir a disciplina INF1413, o aluno
 - esteja capacitado a usar e adaptar, de forma racional, técnicas eficazes e eficientes de controle da qualidade de software
 - entenda que assegurar a qualidade do software é uma atividade sistêmica 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
 - esteja habilitado a aplicar os principais conceitos de controle da qualidade:
 - verificação, validação e aceitação de software
 - tenha conhecimento de como organizar e gerenciar o processo de controle e garantia da qualidade de software

Sistema: conjunto de elementos interdependentes que visam o alcance de um objetivo comum Propriedade sistêmica: um objetivo ou característica de um sistema

Ago 2017

Arndt von Staa © LES/DI/PUC-Ric

29

Pré-requisitos esperados



- INF1301 Programação Modular
 - existe propositalmente uma pequena superposição com assuntos estudados em INF1301 / INF1628
- Alguma prática em programação orientada a objetos
 - uso C++
- Saber utilizar ferramentas de desenvolvimento
 - Eclipse (Java) e/ou Visual Studio (C / C++), ou outro qualquer
 - os trabalhos que envolvam programação utilizarão C++
 - janela de linha de comando
 - "batch files", muito melhor: Lua

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio

Material de estudo



- Livro texto
 - Pezzè, M.; Young, M.; Teste e Análise de Software; Porto Alegre, RS: Bookman; 2008
- Livros complementares
 - Cockburn, A.; Escrevendo Casos de Uso Eficazes Um Guia para Desenvolvedores de Software; São Paulo, SP: Bookman; 2005
 - Adzic, G.; Bridging the Communication Gap: Specification by Example and Agile Acceptance Testing; London, UK: Neuri; Kindle edition; 2009
- Enorme lista de textos complementares
 - vou utilizar material desses artigos e livros em adição ao livro texto
 - listados na seção de bibliografia de cada aula
- As notas de aula estão no site
 - recomenda-se que sejam lidas antes da aula

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio

31

Critério de aprovação



Datas, horários e local das provas

e dos trabalhos podem mudar.

Consultem periodicamente a página de avisos.

- 2 provas, com consulta somente a papel
 - P1: quarta 04/outubro
 - P2: quarta 29/novembro
- 4 trabalhos, prazos
 - T1: segunda 11/setembro
 - T2: quarta 04/outubro
 - T3: quarta 01/novembro
 - T4: quarta 29/novembro
- · Cálculo da nota final
 - G1 = (P1 * 2. + T1 + T2) / 4.
 - G2 = (P2 * 2. + T3 + T4) / 4.
 - GrauFinal = if (G2 >= 3.) then (G1 + G2) / 2. else (G1 + 3. * G2) / 4. fi

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio

Plano de aulas



- O documento descritivo da disciplina Apresentação da disciplina – encontra-se no site da disciplina
 - www.inf.puc-rio.br/~inf1413
 - contém o plano de aulas encontra-se no site da disciplina,
- Estão assinalados os dias
 - em que ocorrerão provas
 - do prazo de entrega de trabalhos
 - em que não haverá aula

Ago 201

Arndt von Staa © LES/DI/PUC-Ric

33

Critérios de correção dos trabalhos



- 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!
- Usualmente irão 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 (ou .rar)
 - se o provedor de acesso não permitir enviar arquivos "zipados" contendo arquivos .exe , .bat ou similares, rebatizem o arquivo .zip ou .rar adicionando um 'x' à extensão, exemplos: abc.zipx , cde.rarx

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio

34

.....

Organização



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

Ago 201

Arndt von Staa © LES/DI/PUC-Rio

25

Organização



- Site da disciplina: 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

Ago 2017

Arndt von Staa © LES/DI/PUC-Rio

Convivência



- Pontualidade
 - inicio as aulas às 17:10
 - se não houver pelo menos 3 alunos quando entrar em sala, não dou aula e considero a matéria dada
- · Eu tenho dificuldade auditiva
 - requer um pouco de paciência por parte dos interlocutores
 - falar bem articulado ajuda muito
 - 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

Ago 201

Arndt von Staa © LES/DI/PUC-Ric

