



PUC

LASO

Laboratório de Software

por

Arndt von Staa

Departamento de Informática

1981

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP-22453
RIO DE JANEIRO - BRASIL

Sumário

1	Introdução	1
2	Motivação	2
3	Descrição do Laboratório de Software	3
	3.1 Objetivos	3
	3.2 Resultados e Benefícios Esperados	5
	3.3 Modo de Operação	6
	3.4 Recursos Típicos Necessários	7
	3.4.1 Equipamento	7
	3.4.2 Recursos Humanos	8
	3.4.3 Custeio e Outros Recursos	8
	3.4.4 Recursos já Disponíveis	9
	3.5 Professores Pesquisadores Permanentes	9
4	Sistema de Apoio ao Desenvolvimento de Software	12
	4.1 Caracterização e Justificativa da Necessidade	12
	4.2 Definição do Sistema	16
	4.3 Requisitos do Sistema de Auxílio ao Desenvolvimento de Software	17
5	Projetos Intermediários	19
6	Formação de Recursos Humanos	22
	Referências Bibliográficas	24

1 Introdução.

O objetivo deste documento é descrever, em linhas gerais, o Laboratório de Software do Departamento de Informática da PUC/RJ. A ênfase principal é dada à diretriz de pesquisa e desenvolvimento sendo realizada neste laboratório. Esta diretriz serve como um mecanismo de coordenação e aglutinação de esforços de uma considerável parcela de atividades de pesquisa do Departamento de Informática.

O Laboratório de Software está centrado em torno de um mini-computador de procedência nacional e provê recursos para o estudo experimental na área de engenharia de software. O programa de pesquisa do laboratório é organizado sob a forma de uma hierarquia de projetos de pesquisa aplicada, descrita no capítulo 3.

Neste documento são descritos os objetivos do laboratório e dos diversos projetos de pesquisa em andamento. Não serão apresentados planos, cronogramas, orçamentos etc., uma vez que temos por objetivo aqui somente relatar o programa de pesquisa, desenvolvimento e ensino suportado pelo Laboratório de Software. Projetos de pesquisa e desenvolvimento específicos serão descritos em outros documentos. Estes conterão, além da definição de objetivos, metas, produtos etc., também os planos e orçamentos necessários para o alcance destes.

O Laboratório de Software atende aos objetivos estabelecidos no Relatório da Comissão de Software e Serviços da Secretaria Especial de Informática (SEI80), em especial às diretrizes presidenciais:

5.1.4 desenvolvimento da Indústria de Software e Serviços;

5.1.5 instituição de normas e padrões, de homologação e certificação da qualidade de produtos e serviços.

e às recomendações:

R3- prioridade de investimentos;

R5- realização de projetos em conjunto entre empresas e universidades;

R6- desenvolvimento de software de interesse nacional;

R14- recursos às universidades;

R15- formação de pessoal;

R16- incremento da qualidade dos cursos já existentes;

R17- software para outros setores de tecnologia;

R18- intercâmbio de software.

A seguir será apresentado o Laboratório de Software do Departamento de Informática da PUC/RJ. A apresentação será feita descrevendo-se o laboratório em si, bem como descrevendo-se um conjunto de metas a serem alcançadas por intermédio de projetos de pesquisa e desenvolvimento a serem levados a cabo neste laboratório.

2 Motivação

Software é um componente complexo e caro de sistemas automatizados. Software é composto por documentação, dados, programas e procedimentos, e é construído com o objetivo de instruir máquinas (computadores, periféricos, instrumentação etc.) e pessoas (operadores, usuários, auditores etc.) no sentido da realização de um conjunto bem definido de tarefas de processamento (coleta, cômputo, transformação, exibição etc.) de dados. Software é, pois, um instrumento para alcançar um fim específico: o de transformar dados de entrada e dados armazenados em arquivos, em resultados úteis e oportunos.

O desenvolvimento de software é um processo caro e demorado. Como a qualidade do software é medida por intermédio do serviço que realiza, conclui-se que, a menos que se utilize um processo de desenvolvimento eficaz, o risco de não se alcançar, ou alcançar-se parcialmente, o objetivo desejado é consideravelmente alto, levando a sérios prejuízos financeiros.

A prática tem mostrado uma frequência excessivamente alta de software possuindo qualidade no mínimo duvidosa. A prática tem mostrado, ainda, que grande parte dos problemas com a baixa qualidade do software advêm da inadequação do processo de seu desenvolvimento. Torna-se então mister o estudo metódico deste processo de desenvolvimento, bem como de problemas relacionados com o software em si, ou seja, torna-se necessário o estudo metódico de Engenharia de Software.

Engenharia de Software é a aplicação de ciência, matemática, técnicas e metodologias visando o desenvolvimento econômico de software que produza resultados úteis na oportunidade certa e que, além disso, seja econômico, utilizável, durável, mensurável e opere eficazmente em máquinas e ambientes reais.

Neste contexto é reconhecido:

- a- ser engenharia de software uma ciência experimental, pois necessita-se da condução de experimentos para abstrair os princípios (regras, disciplinas, teorias etc.) que a fundamentam, necessitando-se também da condução de experimentos para comprovar princípios estabelecidos de forma teórica;
- b- ser necessário um considerável esforço de pesquisa e desenvolvimento de instrumental para identificar e formular os princípios inerentes ao processo de desenvolvimento de software;

- c- tratar-se engenharia de software de uma área de elevada tecnologia, o que reforça a necessidade de receber tratamento racional (científico) e, portanto, enfatiza a necessidade da pesquisa teórica e experimental em engenharia de software;
- d- ser engenharia de software efetivamente um ramo de engenharia no sentido em que é construtiva e utiliza de raciocínios econômicos quando da escolha entre soluções alternativas.

Esta caracterização de engenharia de software, torna patente a necessidade de um laboratório para o estudo experimental dos problemas a ela relacionados.

3 Descrição do Laboratório de Software.

Neste capítulo será descrito, em linhas gerais, o Laboratório de Software em si. Em capítulos posteriores serão definidas as principais metas a serem alcançadas através de atividades a serem realizadas neste laboratório.

O Laboratório de Software gravita em torno de um mini-computador de procedência nacional. Ao contrário de laboratórios convencionais, este laboratório é geograficamente distribuído, bastando para o seu uso o acesso a um terminal ligado ao processador do laboratório.

A descrição do laboratório será efetuada descrevendo-se os objetivos, os resultados, os benefícios esperados, o modo de operação do laboratório, os recursos típicos necessários para a sua instalação e continuidade e, finalmente, o seu corpo de pesquisadores permanente.

3.1 Objetivos.

Os objetivos do Laboratório de Software são:

- i- desenvolvimento de instrumental de auxílio ao desenvolvimento de software;
- ii- estudo experimental do software em si, do processo de seu desenvolvimento, bem como da gerência de desenvolvimento de software;
- iii- formação de recursos humanos.

O desenvolvimento de instrumental tem por objetivo:

- i- desenvolver, experimentar e validar metodologias de especificação, projeto, construção, teste e gerência de desenvolvimento e manutenção de software. Estas metodologias deverão propiciar o desenvolvimento sistemático e racional de software de qualidade controlada;
- ii- desenvolver, experimentar e validar técnicas de projeto, de programação, de documentação, de arquivamento de dados, de certificação, de teste e de verificação. Estas técnicas deverão estabelecer uma base sólida de "know how" que propicie a construção de software de qualidade controlada;
- iii- desenvolver, experimentar, testar e validar ferramentas, algoritmos e fundamentos que auxiliem e dêem o embasamento necessário ao desenvolvimento de software de qualidade controlada;
- iv- desenvolver protótipos e modelos que permitam a condução de experimentos e a predição do comportamento do instrumental e/ou dos produtos a serem posteriormente desenvolvidos.

O estudo experimental visa:

- i- confirmar ou negar hipóteses quanto à eficácia do instrumental desenvolvido;
- ii- coletar dados que permitam a criação de instrumental eficaz.

A formação de recursos humanos visa:

- i- treinar pessoas no uso e adaptação dos recursos disponíveis no laboratório (hardware, software básico e de suporte);
- ii- treinar pessoas no uso do instrumental desenvolvido no laboratório;
- iii- formar pessoas de modo que estas por sua vez possam contribuir na criação de instrumental, na condução de experimentos e/ou na formação de recursos humanos.

3.2 Resultados e Benefícios Esperados.

Os resultados dos trabalhos no laboratório de software tornar-se-ão visíveis nas seguintes formas:

- i- instrumental desenvolvido, testado, validado e disponível ao uso por terceiros;
- ii- sistemas experimentais, protótipos e modelos de simulação desenvolvidos, testados e disponíveis para o uso e estudo por terceiros;
- iii- documentos tais como teses, dissertações, monografias, artigos etc., descrevendo os resultados alcançados e estabelecendo uma base sólida de conhecimentos e "know how" para uso em projetos posteriores;

IV. RECURSOS HUMANOS FORMADOS:

Os benefícios auferidos com tais resultados são, em linhas gerais:

- i- o uso generalizado do instrumental desenvolvido, levando a práticas de desenvolvimento de software mais econômicas e eficazes e, portanto, contribuindo para o aumento da produtividade e da qualidade, bem como para a redução dos prazos, custos e riscos do desenvolvimento de software;
- ii- dar soluções a problemas ainda não resolvidos, bem como identificar e resolver problemas não antes conhecidos;
- iii- criação e disseminação de tecnologia e "know how" de desenvolvimento econômico de software de elevada qualidade, bem como de instrumental dirigido a mini computadores nacionais. Estes "know how" e tecnologia visam atender às necessidades de diferentes gêneros de software (firmware, básico, suporte, aplicação, sob medida, produção, etc.) e diferentes aplicações do software (comercial, científico, automação industrial, comando e controle, etc.);
- iv- através da maior competência profissional, e maior produtividade do pessoal formado no laboratório.

Os resultados específicos esperados serão definidos junto com os projetos de desenvolvimento a serem conduzidos no laboratório de software, na modalidade descrita a seguir.

3.3 Modo de Operação.

Com o intuito de adquirir e disseminar conhecimentos que permitam um melhor desempenho nas atividades relacionadas com engenharia de software, bem como com o intuito de desenvolver princípios e ferramentas de trabalho eficazes, o laboratório de software dará suporte a um conjunto hierárquico de projetos de pesquisa e desenvolvimento, assim constituído:

- 1- um projeto de longo prazo visando especificar, desenvolver e implementar um sistema automatizado para o auxílio ao desenvolvimento de software. Este projeto serve como diretriz de pesquisa e desenvolvimento aos demais projetos a serem elaborados no laboratório. O projeto global será descrito no capítulo 4;
- 2- UM CONJUNTO DE PROJETOS INTERMEDIÁRIOS VISANDO DESENVOLVER E IMPLEMENTAR METODOLOGIAS E FERRAMENTAS AUTONOMAS; PROJETADAS DE FORMA A COMPOR O SISTEMA DE AUXÍLIO AO DESENVOLVIMENTO DE SOFTWARE DEFINIDO ACIMA: Cada um destes projetos intermediários possuirá um objetivo claro e bem definido, além disso requererá a colaboração de diversas pessoas e a execução de diversos trabalhos de pesquisa teórica e/ou experimental para possibilitar alcançar-se a meta desejada. Os diversos projetos intermediários em andamento serão descritos no capítulo 5;
- 3- um conjunto de pequenos projetos visando experimentar, desenvolver e implementar metodologias, programas etc., específicos e bem definidos. Os pequenos projetos realizarão os objetivos dos projetos intermediários. Pequenos projetos serão definidos à medida que se tornarem necessários. Cada um dos pequenos projetos possuirá uma meta e um prazo de conclusão bem definidos, e requererá poucos recursos para a sua realização. Exemplos de pequenos projetos são dissertações de mestrado, implementações de programas de complexidades medianas, conjuntos de experimentos de complexidades medianas, etc.

Esta estrutura hierárquica de projetos foi escolhida por propiciar uma melhor orientação geral e, conseqüentemente, facilitar a integração dos resultados das diversas atividades de pesquisa e desenvolvimento. Esta estrutura hierárquica de projetos, permitirá ainda o desenvolvimento do sistema de auxílio ao desenvolvimento de software em um ambiente onde é natural a elevada rotatividade de pessoal. Este é o caso aqui, uma vez que o laboratório tem também o objetivo de formação de recursos humanos.

O projeto geral coordenará os objetivos de cada um dos projetos intermediários. Estes, por sua vez, coordenarão os objetivos dos pequenos projetos. A organização hierárquica apresentada permite ainda acompanhar de forma eficaz o pro-

gresso em direção ao objetivo comum, uma vez que os pequenos projetos produzirão resultados visíveis e, devido à existência dos projetos intermediários que os coordenam, serão definidos de forma a levarem a resultados aproveitáveis nos projetos de âmbito maior.

3.4 Recursos Típicos Necessários.

Nesta seção serão identificados, em linhas gerais, os recursos necessários para a implantação e manutenção do Laboratório de Software.

Nos documentos que descrevem os projetos específicos serão definidos os recursos necessários, planos, cronogramas, orçamentos, resultados esperados etc. Os recursos financeiros necessários aos diversos projetos específicos serão solicitados às diversas instituições de amparo e fomento do desenvolvimento científico e tecnológico nacional.

3.4.1 Equipamento.

O recurso indispensável para o laboratório é um mini-computador de procedência nacional com as seguintes características principais:

- i- suporte ao processamento interativo, tanto para a edição de arquivos quanto para a execução de programas;
- ii- facilidade de acesso e uso;
- iii- confiabilidade e agilidade da manutenção de hardware e de software básico;
- iv- capacidade adequada de armazenamento de programas, dados e textos;
- v- capacidade de impressão de letras maiúsculas e minúsculas (produção de documentos).

3.4.2 Recursos Humanos.

Como este laboratório é dirigido também à formação de recursos humanos, deve-se prever alta rotatividade do pessoal do laboratório, esperando-se continuidade de participação por períodos prolongados somente do corpo de professores pesquisadores do laboratório. A seguir é dada uma ordem de grandeza do corpo de pesquisadores, projetistas e desenvolvedores do laboratório:

- professores pesquisadores, cerca de 10, com uma dedicação média de cerca de 15 horas por semana ao laboratório;
- alunos do programa de doutorado, cerca de 3, desenvolvendo teses relacionadas com trabalhos realizados no laboratório;
- alunos do programa de mestrado, cerca de 15, desenvolvendo dissertações relacionadas com o desenvolvimento e o estudo experimental de instrumental de desenvolvimento de software;
- monitores e estagiários, cerca de 15, auxiliando no desenvolvimento de instrumental e/ou conduzindo experimentos;
- técnicos, programadores etc. necessários ao suporte operacional do laboratório, em particular de seu equipamento, cerca de 2.

Outros recursos humanos necessários são operadores e pessoal de suporte administrativo.

3.4.3 Custeio e Outros Recursos.

No que tange a outros recursos, os itens mais ponderáveis são:

- investimento em infra-estrutura para a instalação do laboratório (estabilizadores, cabos, etc.)
- custeio de material de consumo, em particular de formulários contínuos e material de expediente.
- custeio dos serviços de manutenção de hardware e software;
- custeio da distribuição aos interessados do software desenvolvido no laboratório.

3.4.4 Recursos já Disponíveis.

Uma parcela ponderável dos recursos necessários para a instalação do laboratório já estão disponíveis na PUC/RJ a saber:

- i- espaço físico;
- ii- parte da infra-estrutura (ar condicionado, fitoteca etc.);
- iii- infra-estrutura administrativa;
- iv- custeio para os pesquisadores permanentes do laboratório. Como estes são professores do Departamento de Informática da PUC/RJ, já existem recursos para o seu custeio. Serão solicitados recursos extra (complementação salarial) para pesquisadores que participem de projetos específicos. Esta solicitação dependerá de cada projeto específico.

3.5 Professores Pesquisadores Permanentes.

ALBRECHT KARL VON PLEHWE

"Diplom-Mathematiker" (1967) e Doutor em Matemática Aplicada (1970) pela Universidade de Freiburg, Alemanha. Suas áreas de interesse compreendem sistemas operacionais e arquitetura de computadores. Publicou trabalhos e orientou teses nas áreas de métodos numéricos e sistemas de computação.

ANTONIO LUZ FURTADO

Bacharel em Economia (1964) pela Universidade do Estado da Guanabara, Mestre em Informática (1969) pela PUC/RJ e Doutor em Ciência da Computação (1974) pela Universidade de Toronto, Canadá. Suas áreas de interesse compreendem estruturas de informação, bancos de dados e teoria dos grafos. Publicou trabalhos e orientou teses nas áreas de extensão de linguagens de programação, estruturas de informação, teoria dos grafos e bancos de dados.

ARNDT VON STAA

Engenheiro Mecânico (1965) pela PUC/RJ, Mestre em Informática (1969) pela PUC/RJ em programa combinado com a Universidade de Waterloo, e Doutor em Ciência da Computação (1974) pela Universidade de Waterloo, Canadá. Suas áreas de interesse compreendem linguagens de programação, sistemas operacionais, desenvolvimento de software e arquitetura de computadores. Publicou trabalhos e orientou teses nas áreas de estruturas de dados, linguagens de programação, desenvolvimento de software e sistemas operacionais.

CARLOS JOSÉ PEREIRA DE LUCENA

Bacharel em Economia e Matemática (1965) pela PUC/RJ, Mestre em Ciência da Computação (1969) pela Universidade de Waterloo, Canadá, e Doutor em Ciência da Computação (1975) pela Universidade da Califórnia em Los Angeles, EEUU. Suas áreas de interesse compreendem semântica de linguagens de programação e engenharia de software. Publicou trabalhos e orientou teses nas áreas de estruturas de dados, linguagens de programação e engenharia de software.

DANIEL ALBERTO MENASCÉ

Engenheiro Eletricista, Sistemas, (1973) pela PUC/RJ, Mestre em Informática (1975) também pela PUC/RJ, e Doutor em Ciência da Computação (1978) pela Universidade da Califórnia em Los Angeles. Suas áreas de interesse compreendem bancos de dados, teleprocessamento, sistemas de dados distribuídos e teoria de filas. Publicou trabalhos e orientou teses em sistemas de dados distribuídos e transmissão de mensagens.

DANIEL SCHWABE

Bacharel em Matemática (1974) pela PUC/RJ, Mestre em Informática (1976) pela PUC/RJ e Doutor em Ciência da Computação (1981) pela Universidade da Califórnia em Los Angeles, EEUU. Suas áreas de interesse compreendem sistemas distribuídos e engenharia de software.

HANS PETER O. ALBRECHT

Mestre em Matemática Aplicada (1963) pela Universidade de Hannover, Alemanha, Doutor em Matemática Aplicada (1965) pela Universidade de Munique, Alemanha, e "Habilitation" (1975) na Universidade de Dortmund, Alemanha. Suas áreas de interesse compreendem matemática computacional, análise numérica, otimização e desenvolvimento de software para aplicações científicas em geral. Publicou trabalhos e orientou teses em análise numérica.

MARCO ANTONIO CASANOVA

Engenheiro Eletrônico (1975) pelo Instituto Militar de Engenharia, Mestre em Informática (1976) pela PUC/RJ e Doutor em Ciência da Computação (1980) pela Universidade de Harvard, EEUU. Suas áreas de interesse compreendem semântica de linguagens de programação e sistemas de dados distribuídos.

MICHAEL ANTHONY STANTON

Bacharel em Matemática (1967) pela Universidade de Cambridge e Doutor em Matemática (1971) também pela Universidade de Cambridge, Inglaterra. Suas áreas de interesse são métodos numéricos e sistemas operacionais. Publicou trabalhos e orientou teses em matemática computacional e sistemas de computação.

RAUL CÉSAR BAPTISTA MARTINS

Engenheiro de Telecomunicações (1972) pelo Instituto Militar de Engenharia, RJ, e Mestre em Engenharia Elétrica (1974) também pelo IME. Suas áreas de interesse compreendem teoria de computação e engenharia de software.

RUBENS NASCIMENTO MELO

Engenheiro Eletrônico (1968) pelo ITA, Mestre em Ciência da Computação (1971) pelo ITA, e Doutor em Ciência da Computação (1976) também pelo Instituto Tecnológico da Aeronáutica. Suas áreas de interesse compreendem bancos de dados, sistemas de informação e engenharia de software. Publicou trabalhos e orientou teses nas áreas de sistemas de informação, técnicas de programação, processamento gráfico e bancos de dados.

THEREZINHA DA COSTA FERREIRA CHAVES

Licenciada em Matemática (1964) pela Universidade Federal do Rio de Janeiro, Mestre em Informática (1971) pela PUC/RJ, e Doutora em Ciências, Informática (1981), também pela PUC/RJ. Suas áreas de interesse compreendem matemática computacional, matemática aplicada e desenvolvimento de linguagens para uso científico. Orientou teses e publicou trabalhos em análise numérica.

4 Sistema de Apoio ao Desenvolvimento de Software

4.1 Caracterização e Justificativa da Necessidade

É fato conhecido que, à medida que sistemas automatizados vão se tornando mais complexos e sofisticados, o esforço despendido em seu desenvolvimento cresce exponencialmente. Ademais, para desenvolver sistemas complexos são necessário instrumental (ferramentas, metodologias), possibilitando ao desenvolvedor dominar a complexidade, sendo virtualmente impossível o desenvolvimento de sistemas complexos sem a disponibilidade de tal instrumental. A sofisticação e a complexidade inerente a este instrumental de auxílio ao desenvolvimento de software, cresce à medida que aumenta a complexidade do sistema a cujo desenvolvimento dão suporte. Finalmente, é sabido que sistemas complexos e/ou sofisticados necessitam possuir elevado nível de qualidade para que sejam úteis.

Tornam-se necessários, então, métodos e ferramentas de trabalho que possibilitem o desenvolvimento, ou pelo menos reduzam o esforço necessário para o desenvolvimento de sistemas complexos e/ou sofisticados. Estes métodos e ferramentas devem simultaneamente:

- permitir a redução do custo total do sistema a ser desenvolvido (custo de desenvolvimento, mais custo de operação, mais custo de manutenção, mais custos induzidos por falhas do sistema e/ou de seu ambiente);
- assegurar que o sistema após desenvolvido efetivamente possua os níveis de qualidade especificados quando da definição (especificação) deste sistema. São exemplos de requisitos de qualidade: eficácia, eficiência, robustez, confiabilidade, manutenibilidade, flexibilidade, etc.;
- possibilitar ao desenvolvedor vencer barreiras de complexidade.

Nos últimos anos tem sido efetuado um considerável esforço em pesquisa experimental e teórica, visando:

- identificar os princípios, problemas e dificuldades mais comumente encontrados durante o desenvolvimento de software de elevada qualidade;
- desenvolver metodologias e ferramentas que solucionem tais problemas, reduzindo o impacto causado pelas dificuldades identificadas.

No decorrer deste esforço foi comprovado ser o computador o instrumento de apoio mais eficaz, desde que adequadamente programado. Assim já existem e continuamente surgem novas ferramentas baseadas no uso de computadores, onde estas ferramentas têm comprovado sua eficácia no auxílio à solução de problemas específicos encontrados durante o desenvolvimento de software.

O uso do computador como instrumento de auxílio ao desenvolvimento de sistemas de programação não é novidade. Tal uso já vem sendo feito há muito, sendo os exemplos mais notórios: sistemas de processamento de linguagens (compiladores, interpretadores), sistemas operacionais, sistemas de gerência de arquivos e de dados; sistemas de edição de arquivos; sistemas de execução em tempo compartilhado ("time-sharing") etc.

A aplicabilidade econômica de ferramentas automatizadas não é restrita a grandes sistemas. Tais ferramentas prestam inegável auxílio também ao desenvolvimento de médios e pequenos sistemas automatizados. Contudo, a rentabilidade destas ferramentas automatizadas é de difícil avaliação, sendo certamente positiva caso os custos de desenvolvimento sejam amortizados por diversos projetos de desenvolvimento utilizando tais ferramentas. Assim, a rentabilidade de ferramentas automatizadas para o auxílio ao desenvolvimento de pequenos e médios sistemas é maior à medida que estas ferramentas sejam de aplicação mais generalizada, e à medida que estejam disponíveis a mais grupos de desenvolvimento de software.

Ferramentas automatizadas têm sido sediadas costumeiramente em grandes computadores. Isto por ser mais conveniente desenvolver tais ferramentas em equipamento pouco restritivo. Além disso, ferramentas automatizadas são imprescindíveis ao desenvolvimento de grandes sistemas complexos, constituindo pois um investimento economicamente justificado neste caso.

Contudo, não é um requisito indispensável dispor-se de um sistema de grande porte para implementar e economicamente utilizar ferramentas automatizadas. Hoje é reconhecido ser perfeitamente viável criar-se poderosos sistemas de auxílio ao desenvolvimento de sistemas automatizados sediados em minicomputadores. Exemplos de tais sistemas já existem, utilizando micro ou mini computadores e, se bem que simples

e, muitas vezes, incompletos, dando suporte automatizado eficaz a uma parcela não desprezível de atividades inerentes ao processo de desenvolvimento e manutenção de software Luce-na80 .

É também de conhecimento generalizado estarem em desenvolvimento sistemas automatizados de auxílio ao desenvolvimento de software, residentes em minicomputadores dedicados a esta atividade de desenvolvimento. Estando disponível um tal sistema, a equipe responsável pela implementação e manutenção de um sistema automatizado X, utilizaria o processador de suporte para criar, testar, validar e manter este sistema automatizado X. À medida que as versões de produção do sistema X são aprovadas, estas seriam transferidas para processadores de produção por intermédio de linhas de comunicação interligando os processadores de suporte e de produção. Sofisticados sistemas de auxílio ao desenvolvimento poderão até executar, para fins de teste, versões instrumentadas do sistema X através do processador de suporte, utilizando porém um processador de produção para a execução efetiva (teleprocessamento). Desta forma torna-se possível utilizar processadores de desenvolvimento e de produção heterogêneos. Cabe observar que os processadores de desenvolvimento e de produção não são obrigatoriamente diferentes, podendo ser na realidade um único processador.

Apesar de ser reconhecida a utilidade de sistemas de auxílio ao desenvolvimento, empresas em geral recusam projetos de desenvolvimento de tais sistemas, uma vez que:

- são projetos de duração muito longa;
- são projetos custosos, produzindo benefícios de difícil quantificação principalmente sob a forma de um aumento da produtividade no desenvolvimento de software, bem como de um aumento da qualidade deste software, demorando, pois, a tornarem-se evidentes e requerendo medições complexas para evidenciar estes benefícios;
- requerem considerável esforço de pesquisa e de experimentação, uma vez que o conhecimento que fundamenta a engenharia de software é primordialmente empírico.

Assim, o desenvolvimento de sistemas automatizados de auxílio ao desenvolvimento de software caracteriza projetos de pesquisa e desenvolvimento a serem conduzidos primordialmente em instituições de pesquisa.

Além do objetivo final - o desenvolvimento de um sistema de auxílio ao desenvolvimento de software - o projeto de pesquisa e desenvolvimento aqui apresentado produzirá vários resultados intermediários úteis mesmo isoladamente, tais como por exemplo:

- metodologias para especificação, projeto, implementação, teste, manutenção etc. de software;
- ferramentas automatizadas utilizáveis independentemente dando suporte a atividades específicas do desenvolvimento. Estas ferramentas estarão fortemente relacionadas com as metodologias desenvolvidas;
- relatórios (artigos, monografias, dissertações, teses, livros etc.) divulgando os conhecimentos adquiridos;
- instrumental automatizado para o planejamento, acompanhamento, controle e gerência de projetos de desenvolvimento de software;
- formação de recursos humanos altamente qualificados.

Baseado na argumentação exposta até agora, pode-se assegurar que:

- i- o projeto tem por objetivo uma meta alcançável e de inegável valor;
- ii- durante a evolução do projeto surgirão diversos resultados transferíveis para o setor industrial, constituindo assim benefícios intermediários reais;
- iii- os resultados do projeto, intermediários e final, inegavelmente, têm aplicação no país, uma vez que já existe demanda por um instrumental eficaz, bem como já existe uma considerável atividade de desenvolvimento de sistemas automatizados complexos, carecendo-se, no entanto, do instrumental de auxílio ao seu desenvolvimento.

Portanto, projetos de desenvolvimento de sistemas de auxílio ao desenvolvimento de software constituem investimentos de ensino, pesquisa e desenvolvimento com elevada expectativa de retorno a médio prazo. Pode-se afirmar ainda que este retorno será maior do que o montante investido, bem como que contribuirá para a consolidação da Indústria Nacional de Informática (hardware, software, instrumentação). Cabe aqui salientar que, apesar do sistema ser desenvolvido para um equipamento específico, o conhecimento adquirido e a arquitetura do sistema são passíveis de serem portadas para outros equipamentos. O esforço para efetuar esta adaptação é estimado ser suficientemente pequeno para tornar-se compensador. Esta possibilidade de portar a baixo custo aumenta ainda mais o retorno esperado para o investimento.

4.2 Definição do Sistema.

O sistema de apoio ao desenvolvimento de software é um sistema integrado (ou integrável) visando apoiar as seguintes atividades do desenvolvimento de software:

- a- planejamento, controle e gerência de desenvolvimento de software;
- b- especificação, projeto e validação de componentes de software;
- c- desenvolvimento, transformação e/ou composição de componentes de software;
- d- testes, certificação, verificação, validação e controle de qualidade de componentes, programas, dados e documentação do software;
- e- documentação interna e externa dos componentes, programas, subsistemas e sistemas;
- f- controle da evolução (manutenção e auditoria de sistemas através do computador) do software após ter iniciado a fase de operação produtiva;
- g- gerência de configurações de software (controle de alterações), durante o desenvolvimento bem como durante a posterior vida útil do programa ou sistema;
- h- proteção e segurança do software contra o seu mau uso, falhas internas e/ou ambientais, quer estes mau uso e/ou falhas sejam intencionais ou acidentais.

Para que o sistema de apoio ao desenvolvimento de software seja eficaz, os seus componentes devem ser complementares entre si, devendo existir interfaces adequadas entre estes componentes. No entanto, não é necessário que todos os componentes sejam automatizados. Em muitos casos poderá ser mais econômico dispor-se de um suporte semi-automatizado ou, até mesmo manual, desde que adequadamente disciplinado.

Os objetivos a serem alcançados pelo sistema automatizado de auxílio ao desenvolvimento de software são:

- 1- assegurar o alcance de níveis de qualidade elevados nos sistemas automatizados desenvolvidos com seu auxílio;
- 2- aumentar a produtividade no desenvolvimento de software;
- 3- reduzir os custos totais (desenvolvimento, manutenção, operação produtiva, decorrentes de falhas, etc.) de sistemas automatizados;

- 4- reduzir o tempo necessário para tornar operacionais sistemas ou alterações a sistemas;
- 5- assegurar uma maior durabilidade dos sistemas através de uma maior flexibilidade, adaptabilidade e portatili-
dade destes sistemas.

Um sistema de auxílio tal como proposto acima consta na realidade de um conjunto de subsistemas úteis per se e que podem encontrar aplicabilidade também em áreas diferentes da de desenvolvimento de software. É possível então o desenvolvimento parcelado, bem como a utilização autônoma dos diversos subsistemas componentes.

4.3 Requisitos do Sistema de Auxílio ao Desenvolvimento de Software.

Nesta seção serão definidos os diversos requisitos a serem atendidos pelo sistema de apoio ao desenvolvimento de software. Estes requisitos determinam os critérios de avaliação do sucesso dos resultados produzidos no laboratório.

O sistema de auxílio deverá dar apoio interativo e/ou "batch" às seguintes classes de usuários:

- i- engenheiros, analistas, etc., ou seja pessoas responsáveis pela especificação, validação e posterior aceitação dos componentes, programas, dados e documentação a serem desenvolvidos;
- ii- projetistas e programadores responsáveis pela arquitetura, projeto, desenvolvimento, teste, certificação e verificação dos componentes, programas, dados e documentação;
- iii- equipes de teste e auditoria responsáveis pelo controle de qualidade, teste, certificação, verificação e validação dos componentes, programas, dados e documentação;
- iv- equipes de manutenção responsáveis pela adaptação, correção, perfeição e extensão do software após posto em operação;
- v- equipes de auditoria de sistemas responsáveis pela contínua avaliação dos sistemas;
- vi- gerentes de projetos de desenvolvimento de software e administradores do setor de processamento de dados.

O sistema de auxílio ao desenvolvimento de software deverá dar suporte ao desenvolvimento de:

- i- sistemas aplicativos (produto ou sob medida);
- ii- sistemas de automação industrial, e/ou de comando e controle;
- iii- sistemas de suporte, ou apoio, a outros sistemas;
- iv- software básico e firmware;

O sistema de auxílio ao desenvolvimento deve ainda:

- i- ser configurável de forma a atender às necessidades de uma grande gama de equipes de desenvolvimento;
- ii- ser configurável de modo a atender às necessidades de uma grande gama de projetos de desenvolvimento de sistemas, diferenciados pelo seu porte e/ou complexidade;
- iii- apoiar o desenvolvimento concorrente de uma grande gama de sistemas de naturezas diferentes;
- iv- assegurar, para cada configuração e classe de sistemas a serem produzidos, a compatibilidade do custo de uso e aquisição do sistema de auxílio ao desenvolvimento, com o efetivo aumento de qualidade dos sistemas produzidos e a redução de esforço requerido para o seu desenvolvimento e manutenção (redução do custo total);
- v- dar suporte econômico ao desenvolvimento de sistemas de pequeno e médio porte. Fica evidente assim que o sistema de auxílio deverá poder operar em pequenos processadores;
- vi- permitir o desenvolvimento de sistemas destinados a operarem no processador contendo o sistema de auxílio ao desenvolvimento, bem como deverá permitir o desenvolvimento de sistemas destinados a outros processadores, ligados ao de desenvolvimento através de linhas de teleprocessamento.

5 Projetos Intermediários.

Neste capítulo serão apresentados, em linhas gerais, os projetos de pesquisa em andamento no Departamento de Informática, onde estes projetos se coadunam com o sistema de auxílio ao desenvolvimento de software. Estes projetos de pesquisa e desenvolvimento são descritos em mais detalhe em outros documentos.

Os seguintes projetos de pesquisa e desenvolvimento já se encontram em andamento no Departamento de Informática:

1- Ambiente de Teste de Programas

Objetivo: Especificar, desenvolver, e validar experimentalmente um sistema de auxílio à condução de testes sistemáticos de programas. O sistema deverá apoiar a especificação, criação, aplicação e manutenção de casos teste; além disso, a instrumentação dos programas a serem testados; e a avaliação dos resultados dos testes.

Responsáveis:

Arndt von Staa
Carlos José Pereira de Lucena

2- Ambiente para a criação e validação de especificações de software e posterior verificação da conformidade do produto com estas especificações

Objetivo: Estudar teoricamente, especificar, desenvolver e validar experimentalmente um sistema de auxílio à criação de especificações de software, onde este sistema deverá ter a capacidade de verificar a qualidade e a validade das especificações produzidas, auxiliar na verificação da conformidade dos produtos com estas especificações, e auxiliar o projetista nestas tarefas.

Responsáveis:

Arndt von Staa
Carlos José Pereira de Lucena
Daniel Schwabe
Rubens Nascimento Melo

3- Sistema de apoio à documentação

Objetivo: Especificar, desenvolver e validar experimentalmente um sistema para a produção e manutenção da documentação externa e interna de forma que seja consistente com o sistema realmente implementado.

Responsável:

Arndt von Staa

4- Sistema de suporte ao controle de interfaces

Objetivo: Especificar, desenvolver e validar dicionários de dados úteis à especificação, projeto, construção e manutenção de módulos, programas, subsistemas e sistemas. Estes sistemas deverão auxiliar ainda nas tarefas de produção e manutenção de documentação, bem como na gerência de dados.

Responsáveis:

Arndt von Staa
Rubens Nascimento Melo

5- Metodologia de Especificação e Controle da Qualidade de Software

Objetivo: Especificar, desenvolver e validar um modelo de especificação e avaliação da qualidade de software, bem como das métricas relacionadas com este modelo.

Responsável:

Arndt von Staa

6- Metodologia para o desenvolvimento e certificação de pacotes de rotinas numéricas

Objetivo: Especificar, desenvolver e validar experimentalmente uma metodologia para a criação de pacotes de rotinas numéricas de elevada qualidade. Neste projeto serão adaptados também alguns algoritmos de uso generalizado.

Responsáveis:

Peter Albrecht
Therezinha C. F. Chaves

7- Sistema de gerência de banco de dados

Objetivo: Especificar, desenvolver, testar e validar um sistema de gerência de banco de dados baseado no modelo relacional.

Responsáveis:

Antonio Luz Furtado
Marco Antonio Casanova
Rubens Nascimento Melo

8- Suporte a sistemas distribuídos

Objetivo: Especificar, desenvolver, certificar e validar protocolos de comunicação visando ao estabelecimento de redes de teleprocessamento permitindo o uso distribuído de recursos computacionais desta rede.

Responsável:

Daniel A. Menascé
Daniel Schwabe

- 9- Suporte ao desenvolvimento de software para micro-processadores

Objetivo: Especificar, desenvolver e validar "cross montadores", "cross compiladores" e simuladores que permitam a criação de software para sistemas construídos na base de micro-processadores.

Responsáveis:

Albrecht K. von Plehwe
Daniel Schwabe
Michael A. Stanton

- 10- Suporte ao desenvolvimento de sistemas operacionais

Objetivo: Especificar, desenvolver e validar metodologias para a criação de programas concorrentes, objetivando o desenvolvimento de sistemas operacionais.

Responsáveis:

Albrecht K. von Plehwe
Michael A. Stanton

- 11- Arquiteturas experimentais através de firmware

Objetivo: Especificar, desenvolver e validar metodologias e simuladores para o estudo e desenvolvimento de firmware, visando o aproveitamento pleno do potencial da micro-arquitetura de sistemas de computação.

Responsáveis:

Albrecht K. von Plehwe
Arndt von Staa
Michael A. Stanton

- 12- Suporte ao desenvolvimento de processadores de linguagens

Objetivo: Especificar, desenvolver, certificar e validar pré-processadores e geradores de analisadores (lêxica, sintática e semântica), de geradores de otimizadores (independentes e dependentes da máquina objeto) e de geradores de geradores de código portáteis.

Responsáveis:

Arndt von Staa
Michael A. Stanton
Rubens N. Melo

6 Formação de Recursos Humanos.

Um dos objetivos do Laboratório de Software é a formação de recursos humanos. Esta dar-se-á nas seguintes modalidades:

- disciplinas de graduação e pós-graduação utilizando o laboratório para a execução de exercícios;
- disciplinas de graduação e pós-graduação utilizando os recursos do laboratório (hardware e software) como objetos de estudo;
- participação nas atividades de pesquisa do laboratório. Esta modalidade já foi amplamente discutida nas seções anteriores deste documento.

As seguintes disciplinas farão uso regular do laboratório:

- i organização de computadores, sistemas de programação, laboratório de programação (graduação em tecnologia de processamento de dados) cerca de 100 alunos por ano em cada uma das disciplinas;
- ii- sistemas de computação (graduação em engenharia elétrica) cerca de 35 alunos por ano;
- iii- sistemas de computação I e II (pós-graduação em Informática) cerca de 15 alunos por ano;
- iv- programação, e projeto de programas (pós-graduação em Informática) cerca de 40 alunos por ano;
- v- laboratório de programação (graduação em engenharia elétrica) cerca de 15 alunos por ano (eletiva).

Além destes cursos diversos outros exigem trabalhos práticos, onde o recurso computacional específico utilizado é, em princípio, indiferente. Diversos destes trabalhos poderiam ser dirigidos ao uso do equipamento do laboratório, como, por exemplo, o desenvolvimento de aplicações interativas. Cabe salientar que recentemente foi descontinuado o serviço de execução interativa de programas no computador do Rio Datacentro.

A título de informação geral, damos a seguir as ordens de grandeza dos corpos discentes, por programa de ensino:

- extensão universitária (Coordenação Central de Atividades de Extensão) cerca de 500 alunos, nos cursos de programação e análise de sistemas;

- graduação em tecnologia de processamento de dados (projeto 15), cerca de 550 alunos;
- ciclo básico do Centro Técnico Científico (Introdução à Ciência dos Computadores, Cálculo Numérico), cerca de 400 alunos;
- graduação em engenharia elétrica (opção sistemas), cerca de 35 alunos;
- mestrado em informática, cerca de 100 alunos;
- doutorado em informática, cerca de 15 alunos.

Referências Bibliográficas

- Boehm, B.W.
"On the High Cost of Software"; in Horowitz ed.; 1975;
pp 3-14
- Boehm, B.W.
"Software Engineering - As It is"; in Freeman; Lewis
eds.; 1980; pp 37-73
- Brooks, F.P.
The Mythical Man Month; Addison Wesley; 1975
- Freeman, P.
"The Central Role of Design in Software Engineering"; in
Freeman; Lewis eds.; 1980; pp 121-132
- Freeman, H.; Lewis, P.M. eds.
Software Engineering; Academic Press; 1980
- Freeman, P.; Wasserman, A.I. eds.
Software Engineering Education; Springer Verlag, New
York; 1976
- Horowitz, E. ed.
Practical Strategies for Developing Large Software
Systems; Addison Wesley Pub. Co.; 1975
- Lucena, C.J.P.; Martins, R.; Krahe, M.L.
"Projeto de Programas Assistido por Micro-Computador" ;
in Anais XIII Congresso Nacional de Processamento de
Dados, SUCESU; 1980; pp 309-320
- Secretaria Especial de Informática
Relatório da Comissão de Software e Serviços; SEL, Bra-
sília; 1980