



# PUC

---

Série: Monografias em Ciência da Computação  
Nº 11/82

D S A C

DESENVOLVIMENTO DE SOFTWARE APOIADO EM COMPUTADOR

Arndt von Staa

Departamento de Informática

---

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

Série: Monografias em Ciência da Computação  
Nº 11/82

Editor: Marco A. Casanova

Dezembro, 1982

ESAC  
Desenvolvimento de Software  
Aplicado em Computador

por

Arndt von Staa

Departamento de Informática  
Pontifícia Universidade Católica  
Rio de Janeiro  
1982

Formatado: 09 Dez 82

UC - 30092 - 2

## PREFÁCIO

Por estarmos em fase de mudança de equipamento, o presente texto foi impresso utilizando uma impressora convencional equipada com uma cadeia de caracteres padrão. Conseqüentemente, os sinais gráficos utilizados para acentuação de palavras não correspondem aos sinais costumeiros. Apesar disto cremos ser perfeitamente legível o presente texto e a inconveniência dos sinais serem não convencionais, é bem menor do que a inconveniência de não se ter acentuação alguma.

## Resumo

O objetivo deste documento é descrever, em linhas gerais, o sistema de apoio ao desenvolvimento de software em elaboração no Departamento de Informática da PUC/RJ. Este projeto serve como um mecanismo de coordenação e aglutinação de esforços de uma parcela de atividades de pesquisa e desenvolvimento em engenharia de software. Uma vez desenvolvido, o sistema de apoio ao desenvolvimento de software formará um conjunto harmônico e integrado de instrumentos, destinados a dar apoio à especificação, projeto, construção, teste, verificação, validação, manutenção e gerência do desenvolvimento de sistemas de software.

O desenvolvimento deste sistema de apoio requer um considerável esforço de pesquisa e desenvolvimento estimado em vários homens/ano. Para tornar viável o alcance da meta estabelecida, este esforço será particionado em diversos projetos menores, cada um levando a um instrumento específico individualmente utilizável.

A aplicabilidade dos instrumentos a serem pesquisados e desenvolvidos é geral e independe de:

- a. as classes de software cujo desenvolvimento apoiam (suporte, aplicação);
- b. o ambiente de execução (convencional, banco de dados, "on-line");
- c. o modo de distribuição (sob medida, locais múltiplos, software produto).

Os diversos instrumentos fazem parte de uma metodologia estruturada de desenvolvimento de sistemas automatizados. Esta metodologia cobre as atividades de todo o ciclo de vida de um sistema de software. A obediência a esta metodologia é uma das premissas para tornar viável a posterior integração dos diversos instrumentos desenvolvidos.

## 1 Motivação.

Computadores tornaram-se ferramentas indispensáveis para um sem número de aplicações, quer sejam aplicações comerciais, tecnológicas, controle de processo, comando e controle, pesquisa científica ou outras. Muitas tarefas hoje desempenhadas rotineiramente, seriam virtualmente impossíveis de serem realizadas sem o emprego de computadores digitais devidamente programados. Sem sombra de dúvida, qualquer indivíduo vivendo em uma sociedade moderna utiliza-se direta ou indiretamente dos serviços prestados por intermédio de computadores. A tendência, nos próximos anos, é um aumento significativo da utilização direta de computadores por pessoas sem um treinamento formal em computação. Há, inclusive, quem diga que, em um futuro não muito remoto, computadores terão uma participação no dia a dia das pessoas tão grande como hoje é a participação dos automóveis. Cabe observar que não estamos considerando, aqui, a utilização para fins de entretenimento e sim, a utilização para obter resultados que auxiliem o usuário em suas atividades profissionais.

Computadores, no entanto, não resolvem problemas por si só. Eles precisam ser programados e o serviço que poderão prestar será, no máximo, tão bom quanto forem os programas, ou seja, o software que empregarem. A maneira pela qual software deve ser construído vem sofrendo uma série de modificações. Atualmente, ao desenvolvermos um software para fins industriais(1), já não podemos mais tolerar o uso de métodos artesanais de desenvolvimento de software por uma série de razões, entre elas:

- é necessário assegurar elevada confiabilidade e manutibilidade ao software. Ou seja, precisamos assegurar que o software possua um elevado nível de qualidade, onde esta qualidade é definida a priori. Durante o processo de desenvolvimento, o controle da satisfação dos requisitos de qualidade estipulados deve ser continuamente efetuado. Somente assim poderemos assegurar que o software venha a ter a qualidade necessária para poder ser utilizado para fins industriais.
- é necessário assegurar elevada economicidade ao software. Ou seja, precisamos de instrumentos e de técnicas de desenvolvimento que contribuam para a redução dos custos de desenvolvimento e de manutenção do software.
- é necessário aumentar o desempenho (produtividade) das equipes de desenvolvimento de software. Em função do crescimento explosivo da presença de computadores na sociedade, existe hoje uma demanda por software novo

---

(1) Entendemos por software industrial, software desenvolvido para auferir algum benefício diferente de experimentar novas soluções (pesquisa) ou satisfazer uma necessidade individual.

difícil de ser atendida a menos que aumentemos significativamente a produtividade das pessoas que desenvolverão este software.

- é necessário agilizar-se o desenvolvimento de aplicações. São muito comuns queixas quanto à demora do desenvolvimento de software, frequentemente levando à perda de oportunidade dos resultados, apesar dos dados necessários já se encontrarem disponíveis em arquivos mecanizados. Para aproveitar melhor os dados, é necessário que seja tornado mais rápido o desenvolvimento de aplicações, em particular de aplicações para as quais é difícil determinar especificações precisas e completas a priori. Ou seja, é necessário agilizar o desenvolvimento de aplicações onde as especificações são geradas por intermédio de experimentação.
- é necessário capacitar o usuário a criar e a ajustar soluções às suas necessidades, sem, no entanto, exigir dele um conhecimento dos artificialismos e dos detalhes exotéricos tão comuns à computação. Tendo em vista a dificuldade e a demora da comunicação entre usuários e desenvolvedores, torna-se desejável permitir ao usuário ajustar relatórios e/ou processos de cálculo às suas necessidades individuais. Isto é particularmente desejável, por exemplo, em aplicações de apoio à tomada de decisão. Isto pode ser conseguido desenvolvendo-se sistemas "quase programáveis" capazes de atenderem a uma determinada classe de aplicações. O usuário de um tal sistema terá à sua disposição uma linguagem de definição próxima à sua forma de raciocínio. Isto o capacitará a ajustar e/ou a desenvolver aplicações eficazes e suficientemente eficientes para uma boa parte de suas necessidades específicas, sem requerer o auxílio de analistas e/ou programadores e sempre dentro de uma gama pré-definida de possíveis aplicações.

Em decorrência destas novas exigências, surge uma disciplina chamada de Engenharia de Software e que pode ser definida como sendo:

"O desenvolvimento e a aplicação de ciência, matemática, técnicas, métodos, metodologias e ferramentas para o desenvolvimento e a manutenção econômica de software de qualidade preditiva e controlável, operando de modo econômico em máquinas e ambientes reais."

É reconhecido neste contexto:

- a- ser engenharia de software uma ciência experimental, pois necessita da condução de experimentos para abstrair os seus fundamentos (regras, princípios, teorias,

etc.), necessitando também da condução de experimentos para validar os fundamentos estabelecidos por via teórica;

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

Esta caracterização de engenharia de software, torna patente a necessidade de um esforço de pesquisa e desenvolvimento para identificar e formular os princípios inerentes ao processo de desenvolvimento e de manutenção de software, bem como para apresentar soluções técnica e economicamente viáveis aos problemas identificados.

Costumeiramente a pesquisa em engenharia de software tem sido realizada focalizando e detalhando um aspecto específico. Apesar de sua evidente utilidade, o resultado é um conjunto não integrado de soluções. Assim sendo, o objetivo do sistema de apoio ao desenvolvimento de software é servir como mecanismo de aglutinação e integração da pesquisa e do desenvolvimento em engenharia de software.

Software é um componente de sistemas automatizados sendo composto por documentação, dados, programas e procedimentos. É construído com o objetivo de instruir máquinas e pessoas no sentido da realização de um conjunto bem definido de tarefas de processamento de dados. Software é, portanto, um instrumento para alcançar um fim específico: transformar dados em resultados confiáveis, ágeis e oportunos, requerendo pouco esforço por parte de quem faça uso deste software.

De modo geral pode-se dizer que esta visão é uma visão tradicional do emprego de sistemas automatizados. Ela tende a levar a sistemas relativamente rígidos capazes de resolverem somente uma determinada aplicação específica e, conseqüentemente, dificultando o ajuste e a alteração (manutenção) do sistema. Note que modificações são necessárias devido à evolução natural das necessidades e expectativas dos usuários. Numa visão mais moderna, deseja-se que o software, além de ser confiável e fácil de usar, também facilite e agilize a modificação dos processos de cálculo, tabulação e exibição dos resultados. Como já foi mencionado, o ideal é que estas alterações possam ser efetuadas pelo próprio usuário.

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

alcançar (ou alcançar apenas parcialmente) os objetivos desejados é consideravelmente alto. Evidentemente, deixar de alcançar os objetivos traçados pode levar a elevados prejuízos.

Para ser útil, o software necessita possuir um elevado nível de qualidade. No entanto, é ainda grande a porcentagem de software de qualidade no mínimo duvidosa, mesmo no caso de software e/ou de aplicações simples. Tem-se verificado que as principais causas para esta baixa qualidade do software são a inadequação de seu processo de desenvolvimento e a inadequação da gerência deste desenvolvimento. É fato conhecido que, à medida que se exige mais qualidade e/ou à medida que o software se torna mais complexo e sofisticado, o esforço despendido em seu desenvolvimento cresce exponencialmente. Além disso, para desenvolver software é necessário utilizar-se um instrumental(1) eficaz que possibilite assegurar a satisfação dos requisitos de qualidade estabelecidos, e dominar a complexidade.

Existe atualmente um considerável esforço mundial no desenvolvimento de instrumentos de apoio ao desenvolvimento, manutenção e gerência de desenvolvimento de software. No entanto, este esforço, apesar de grande em termos absolutos, é muito pequeno proporcionalmente aos problemas reais encontrados no dia a dia do desenvolvimento e da manutenção de software.

Como resultado de pesquisas já realizadas foram identificados diversos princípios básicos que governam o desenvolvimento de software. Entretanto, grande parte destes resultados não são de aplicabilidade imediata em ambiente industrial, requerendo de um considerável esforço de adaptação para serem utilizáveis em larga escala. Para isto é necessário dominar-se perfeitamente os conceitos e princípios inerentes a estes instrumentos. Torna-se necessário, portanto, pesquisa e formação de recursos humanos de elevado nível.

Além de terem que ser adaptados, muitos dos resultados foram obtidos de modo individual necessitando, portanto, de ajustes que propiciem a integração dos diversos resultados. O objetivo desta integração é desenvolver um conjunto de instrumentos harmônico e consistente e que apoie eficaz e eficientemente todo o ciclo de vida do software desde a sua concepção até a sua descontinuação, incluindo aí a operação e a manutenção deste software.

---

(1) entendemos por instrumentos: normas, padrões, técnicas, métodos, metodologias, ferramentas automatizadas e sistemas de suporte destinados a apoiar o desenvolvimento, a manutenção e a gerência de desenvolvimento de software.



Os prazos de desenvolvimento de software têm sido excessivamente longos. É imprescindível, pois, capacitar os desenvolvedores para reduzir estes prazos, assegurando além disso um aumento da qualidade do software desenvolvido.

Alguns estudos efetuados recentemente evidenciaram que o simples emprego de melhores metodologias pouco contribui para aumentar significativamente o desempenho das equipes de desenvolvimento. A grande vantagem do emprego destas metodologias tem sido a melhoria da qualidade e, principalmente, a maior manutenibilidade dos sistemas desenvolvidos por intermédio destas metodologias. Cabe salientar que esta melhor qualidade é conseguida em geral em conjunto com uma pequena melhoria de desempenho das equipes. Ou seja, é evidente ser vantajoso o emprego destas metodologias quando se considera o tempo de vida útil total do sistema.

Atualmente se reconhece que um dos caminhos a ser trilhado para aumentar simultaneamente a produtividade do desenvolvimento e a qualidade do software desenvolvido é o de utilizar o computador como um instrumento de apoio ao desenvolvimento. Desta forma transfere-se para a máquina uma série das tarefas efetuadas durante o desenvolvimento de sistemas de software. Ou seja, ao invés de constituir um obstáculo ao desenvolvimento de software, o computador passará a ser um aliado dos gerentes, analistas, programadores e controladores de qualidade responsáveis pelo desenvolvimento e pela manutenção deste software.

Outro caminho a ser trilhado é o da reutilização dos diversos componentes criados durante o desenvolvimento de um determinado software. Exemplos de tais componentes são módulos de programas, algoritmos, estruturas de dados, tipos abstratos de dados, projetos físicos, projetos lógicos, especificações, roteiros e arquivos de dados teste, etc. Grande parcela do esforço de desenvolvimento de software é despendida em reescrever, com ligeiras alterações, componentes anteriormente redigidos. Torna-se necessário, pois, o desenvolvimento de técnicas e de ferramentas que permitam reutilizar partes anteriormente redigidas, independentemente destas serem especificações, projetos, programas, módulos, bancos de dados, instrumentos para teste, verificação, validação, etc. Através desta reutilização, processada em escala muito mais ampla do que a que ocorre na simples reutilização de módulos (biblioteca de módulos), espera-se ser possível aumentar em ordens de grandeza a produtividade das equipes de desenvolvimento.

Finalmente, um terceiro caminho a ser trilhado é o do desenvolvimento de sistemas configuráveis e/ou "quase programáveis". Tais sistemas são capazes de dar apoio a uma determinada classe de aplicações bem definida. Caso a linguagem definida para selecionar (configurar) e/ou "quase-programar" seja próxima à cultura do usuário, este ficará capacitado a utilizar o sistema para a definição de aplicações especifi-

cas, sem, para tal, requerer treinamento intensivo. É evidente que isto reduz consideravelmente o tempo para a obtenção dos resultados esperados pelo usuário.

## 2 Sistema de Apoio ao Desenvolvimento de Software.

### 2.1 Descrição Geral.

Baseados na discussão acima concluímos que o sistema de apoio ao desenvolvimento de software deve reduzir o esforço necessário para o desenvolvimento de sistemas de software de qualidade preditível e controlada. Os instrumentos que compõem o sistema de apoio 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, uma vez desenvolvido, efetivamente possua os níveis de qualidade especificados na definição (especificação) deste sistema;
- possibilitar ao desenvolvedor vencer barreiras de prazo (ou de produtividade);
- possibilitar ao desenvolvedor vencer barreiras de qualidade;
- 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, os problemas e as 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 sendo, já existem e continuam surgindo novas ferramentas baseadas no uso de computadores. 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 apoio ao desenvolvimento de sistemas de programação não é novidade. Tal uso já vem sendo feito há muito tempo, sendo os exemplos mais notórios: sistemas de processamento de linguagens (compiladores, interpretadores), sistemas operacionais, sistemas de gerência de arquivos e de bancos de dados; sistemas de edição de arquivos; sistemas de execução em tempo compartilhado ("time-sharing") etc.

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, neste caso, um investimento economicamente justificado. Contudo, não é um requisito indispensável dispor-se de um sistema de grande porte para implementar e utilizar de forma econômica ferramentas automatizadas. Reconhece-se, atualmente, ser viável criar-se poderosos sistemas de apoio ao desenvolvimento de sistemas automatizados sediados em micro e minicomputadores.

A aplicabilidade econômica de ferramentas automatizadas não é restrita ao desenvolvimento de sistemas grandes. Estas ferramentas prestam ímportável apoio também ao desenvolvimento de médios e pequenos sistemas automatizados. Entretanto, a rentabilidade destas ferramentas é de difícil avaliação, sendo certamente positiva caso os seus custos sejam amortizados por diversos projetos de desenvolvimento que utilizem estas ferramentas. Assim sendo, a rentabilidade de ferramentas automatizadas para apoio 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. Isto aumenta a dificuldade de desenvolvimento destas ferramentas, pois passarão a ter que ser ajustáveis a diversos ambientes de desenvolvimento de software.

E de conhecimento generalizado estarem em desenvolvimento sistemas de apoio ao desenvolvimento de software, residentes em micro e minicomputadores dedicados a esta atividade de desenvolvimento. Tais sistemas podem ser imaginados como sendo "terminais inteligentes" capacitados a resolverem uma parcela considerável dos problemas específicos encontrados quando do desenvolvimento e da manutenção de software. Estando disponível um tal sistema, a equipe responsável pela implementação e manutenção de um sistema automatizado X, utilizaria os processadores 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.

Projetos de desenvolvimento de sistemas de apoio ao desenvolvimento de software:

- são projetos de duração muito longa.
- são projetos muito custosos. Além disto, por serem de duração muito longa, o retorno de capital é demorado. Isto obriga que os instrumentos desenvolvidos sejam utilizados em mais de uma instalação. Ou seja, para que estes instrumentos sejam economicamente viáveis, é necessário que sejam desenvolvidos como software produto a ser utilizado em diversas instalações. Torná-los software produto, além de criar dificuldades técnicas, onera ainda mais o seu desenvolvimento e a sua manutenção.
- requerem considerável esforço de pesquisa e de experimentação, uma vez que o conhecimento que fundamenta a engenharia de software ainda é primordialmente empírico. A prática tem mostrado que instrumentos desenvolvidos sem a devida atenção aos seus princípios, tendem a dificultar, ao invés de simplificar, o desenvolvimento de software efetuado por intermédio destes instrumentos.

Assim, o desenvolvimento de sistemas automatizados de apoio 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 apoio ao desenvolvimento de software - o projeto de pesquisa e desenvolvimento aqui apresentado, produzirá vários resultados intermediários úteis mesmo isoladamente. Estes resultados são, em geral, facilmente transferíveis para o setor industrial. São exemplos de resultados:

- metodologias para especificação, projeto, implementação, teste, manutenção etc. de software.
- ferramentas automatizadas dando suporte a atividades específicas do desenvolvimento. Estas ferramentas estarão fortemente relacionadas com as metodologias desenvolvidas. São exemplos de tais ferramentas: dicionários de dados, processadores de texto, sistemas de apoio à geração e controle de qualidade de estruturas modulares, pré-processadores, sistemas de apoio à preparação de programas, etc.
- ferramentas dando suporte ao planejamento, acompanhamento, controle e gerência de projetos de desenvolvimento de software.

- formação de recursos humanos altamente qualificados.
- relatórios (artigos, monografias, dissertações, teses, livros etc.) divulgando os conhecimentos adquiridos.

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 são de interesse da indústria nacional, uma vez que já existe demanda por instrumentos eficazes, bem como já existe uma considerável atividade de desenvolvimento de sistemas automatizados complexos;
- iv- a indústria sente falta de pessoal com boa formação e bom treinamento em engenharia de software. A participação na execução do projeto aqui exposto certamente contribuirá para formar e treinar o pessoal com o perfil procurado.

Portanto, projetos de desenvolvimento de sistemas de apoio ao desenvolvimento de software constituem investimentos de ensino, pesquisa e desenvolvimento com elevada expectativa de retorno a médio prazo. Cabe aqui salientar que, mesmo que o sistema seja desenvolvido para um equipamento específico, o conhecimento adquirido e a arquitetura do sistema são passíveis de serem transportados para outros equipamentos. O esforço para efetuar esta adaptação é estimado ser pequeno. Esta possibilidade de transportar a baixo custo aumenta ainda mais o retorno esperado para o investimento.

## 2.2 Definição do Sistema.

O sistema de apoio ao desenvolvimento de software é um conjunto integrado de instrumentos, mecanizados ou não, visando apoiar as seguintes atividades do desenvolvimento de software:

- a- planejamento, controle e gerência de desenvolvimento de software;
- b- especificação, projeto, verificação 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 configuração de software (controle de alterações) durante o desenvolvimento bem como durante a posterior vida útil do sistema;
- h- estabelecimento de proteção e segurança do software contra o seu mau uso, falhas internas e/ou ambientais, quer 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 económico dispor-se de um suporte semi-automatizado ou, até mesmo manual, desde que adequadamente disciplinado.

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

- 1- assegurar o alcance de níveis de qualidade elevados nos sistemas automatizados desenvolvidos com seu apoio;
- 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 portatibilidade destes sistemas;
- 6- poder operar eficazmente em computadores de baixo custo (micro computadores);

- 7- apresentar rentabilidade também para projetos de pequeno e médio porte.

Um sistema de aplic 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.

### 2.3 Requisitos do Sistema de Apoio 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.

O sistema de apoio 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 colocado em operação;
- v- equipes de auditoria de sistemas responsáveis pela contínua avaliação dos sistemas já em operação;
- vi- gerentes de projetos de desenvolvimento de software e administradores do setor de processamento de dados.

O sistema de apoio 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;

O sistema de apoio 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 apoio 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 apoio deverá poder operar em pequenos processadores;
- vi- permitir o desenvolvimento de sistemas destinados a operarem no processador contendo o sistema de apoio ao desenvolvimento, bem como permitir o desenvolvimento de sistemas destinados a outros processadores, ligados ao desenvolvimento através de linhas de teleprocessamento.



### 2.3 Resultados e Benefícios Esperados.

Os resultados de pesquisa e desenvolvimento relacionados com o sistema de apoio ao desenvolvimento 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 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 emprego de melhores práticas de desenvolvimento e manutenção de software, contribuindo assim para o aumento da produtividade e da qualidade, bem como para a redução dos prazos, custos e riscos do desenvolvimento de software;
- ii- a solução de problemas ainda não resolvidos, bem como a identificação e resolução de problemas anteriormente desconhecidos;
- iii- criação e disseminação de tecnologia e "know how" de desenvolvimento econômico de software de elevada qualidade, bem como de instrumental dirigido a micro e mini computadores nacionais. Estes "know how" e tecnologia visam atender às necessidades de diferentes gêneros e aplicações de software.
- iv- a elevação da competência profissional e da produtividade do pessoal formado ao participar do projeto de pesquisa e desenvolvimento sendo descrito.

Estes benefícios serão auferidos através do emprego dos diferentes instrumentos desenvolvidos. Ao definir-se cada um dos instrumentos, serão identificados os benefícios específicos que estes poderão trazer.

#### 2.4 Alguns caminhos para a solução.

É evidente que o desenvolvimento de um sistema de apoio ao desenvolvimento de software tal como aqui exposto não poderá partir do nada. Ou seja, para viabilizar o desenvolvimento, é necessário utilizar o maior número possível de instrumentos já desenvolvidos e/ou descritos na literatura corrente.

Um dos principais problemas do desenvolvimento é a especificação dos sistemas. Para dominar o problema torna-se necessário identificar os princípios básicos para a criação de especificações evolutivas, verificáveis, validáveis e modificáveis. Ou seja, torna-se necessário aprender a medir especificações, metodologias e linguagens de especificação. Ao mesmo tempo é necessário efetuar experimentos de automação de metodologias de especificação. De posse destes dois conhecimentos, espera-se ser possível definir e implementar uma metodologia eficaz, eficiente e mecanizada.

Sabe-se que esta mecanização não pode querer ser total, por isto ser demasiadamente complexo. Consequentemente, o sistema deverá ser interativo auxiliando o especificador a chegar ao resultado desejado. Este auxílio deve ser altamente flexível e dinamicamente ajustável às necessidades instantâneas encontradas quando da especificação de um determinado sistema.

O caminho a ser seguido é:

- 1- mecanizar o sistema de registro e rastreamento de requisitos.
- 2- criar um sistema de manutenção de dicionários de dados hierárquicos apoiando o desenvolvimento, independente do emprego ou não de bancos de dados para dar suporte à aplicação sendo desenvolvida.
- 3- criar um sistema semi-gráfico de apoio ao traçado de fluxos de dados.
- 4- criar um sistema de apoio à transformação de fluxos de dados em estruturas modulares.
- 5- criar um sistema semi-gráfico de apoio à definição de estruturas modulares.
- 6- criar um sistema de apoio à geração e à formatação de pseudo-código e de tabelas de decisão.
- 7- criar um sistema de preparação de programas em determinada linguagem específica tendo por base as estruturas modulares e as definições de dados contidas no dicionário de dados.

- 8- para cada um destes instrumentos, criar um sistema de apoio ao controle da qualidade do produto gerado por intermédio deste instrumento.

### 3 Resultados já Obtidos.

Nesta seção serão descritos os resultados já alcançados em projetos de pesquisa e desenvolvimento relacionados com o sistema de apoio ao desenvolvimento de software.

#### 1- Sistema de suporte à documentação [Staa78]

Este sistema é um processador de palavra altamente flexível e poderoso. Permite a geração e manipulação de textos de tamanho virtualmente ilimitado. É capaz de formatar, inserir acentuação, hifenizar, inserir notas de rodapé e figuras, etc. O sistema está implementado e é amplamente utilizado.

#### 2- Metodologia de controle de qualidade [Mazzoni81, Barros82a, Staa83]

Define um modelo de especificação e controle da qualidade de software. Esboça ainda os princípios de gerência a serem observados de modo a tornar possível o alcance da qualidade estipulada. Estão identificados os princípios básicos, bem como um conjunto mínimo de critérios de medição e avaliação.

#### 3- Metodologia de especificação de requisitos [Staa81, Staa82a]

Define o conteúdo e o processo de criação de propostas de desenvolvimento de software. Estas propostas identificam os usuários, os objetivos e os requisitos a serem satisfeitos pelo sistema. A metodologia foi experimentada em alguns projetos e demonstrou ser eficaz.

#### 4- Esboço de uma metodologia integrada [Staa82b]

Descreve uma sequência de etapas de desenvolvimento amparadas em metodologias conhecidas. Estas metodologias foram adaptadas de modo a formarem um todo harmônico cobrindo todo o espectro de atividades técnicas e gerenciais que ocorrem durante o ciclo de desenvolvimento de sistemas. O objetivo é assegurar a produção de software de qualidade preeditável e controlável. Este esboço servirá de guia para o desenvolvimento dos diversos instrumentos.

5- Modelo de qualidade de especificações [Barros82b, Rocha82, Staa83]

Descreve os princípios básicos que regem a avaliação e o controle de qualidade de especificações, de metodologias e de linguagens de especificação. Foi experimentalmente e comprovada a aplicabilidade prática do modelo. Atualmente está sendo detalhado com o intuito de definir um conjunto quase completo de critérios de avaliação e medição.

4 Projetos em andamento.

Nesta seção serão apresentados, em linhas gerais, os projetos de pesquisa e desenvolvimento relacionados com o sistema de apoio ao desenvolvimento de software atualmente sendo desenvolvidos.

1- Controle de qualidade de especificações

Objetivo: Estudar, especificar e validar um conjunto de critérios de avaliação da qualidade de especificações e de metodologias e linguagens de especificação.

Responsável:

Ana Regina Cavalcanti da Rocha (doutoranda)

2- Indexação semi-automática de documentos

Objetivo: Especificar, desenvolver e validar experimentalmente um sistema de apoio à produção e manutenção de índices remissivos de documentos

Responsável:

Paulo Roberto Pinheiro Torres (mestrando)

3- Critérios de controle de qualidade de software

Objetivo: Tornar operacional o modelo de especificação e avaliação da qualidade de software particularmente com vistas a software técnico-científico. Para tal serão definidos os diversos fatores e critérios de qualidade, sendo examinada a sua aplicabilidade prática através de exemplos.

Responsável:

Crésio Miranda Ribeiro (mestrando)

#### 4- Arquiteturas experimentais através de firmware

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

Responsável:

Credine da Silva Renezes (mestrando)

#### 6 Projetos de pesquisa e desenvolvimento propostos.

Nesta seção serão apresentados, em linhas gerais, os projetos de pesquisa e desenvolvimento relacionados com o sistema de apoio ao desenvolvimento de software a serem iniciados em futuro próximo:

##### 1- Ambiente de Teste de Programas

Objetivo: Especificar, desenvolver e validar experimentalmente um sistema de apoio à condução de testes sistemáticos de programas. O sistema deverá apoiar a especificação, criação, aplicação e manutenção de casos teste. Deverá apoiar também a instrumentação dos programas a serem testados, bem como deverá auxiliar na avaliação dos resultados dos testes.

##### 2- 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.

##### 3- Suporte ao desenvolvimento de processadores de linguagens

Objetivo: Especificar, desenvolver, certificar e validar pré-processadores e geradores de analisadores léxicos, sintáticos e semânticos e de geradores de código portáteis.

##### 4- Suporte semi-gráfico ao traçado de organizações modulares

Objetivo: Estudar, especificar, desenvolver e validar um sistema de apoio à geração de gráficos representando organizações modulares. O meio de impressão a ser utilizado é impressora convencional. O sistema será interativo e dará suporte ao controle de qualidade e ao caminhamento em níveis hierárquicos ("zooming").

## 7 Benefícios colaterais.

O sistema de apoio ao desenvolvimento de software atende aos objetivos estabelecidos no Relatório da Comissão de Software e Serviços da Secretaria Especial de Informática (SEI80), em especial às diretrizes:

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:

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

R6- desenvolvimento de software de interesse nacional;

R15- formação de pessoal;

R16- incremento da qualidade dos cursos já existentes.

R18- intercâmbio de software.

## Referências Bibliográficas.

- Barros, S.C.  
Controle de Qualidade Aplicado ao Desenvolvimento de Software; Dissertação de Mestrado, Departamento de Informática, PUC/RJ, Rio de Janeiro; 1982
- Barros, J.C.  
Validação de um Modelo para o Controle da Qualidade de Linguagens de Especificação; Dissertação de Mestrado, Universidade Federal de Pernambuco, Recife, PE; 1982
- 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; 1975
- Lucena, C.J.P.; Martins, R.; Krahe, M.L.  
 "Projeto de Programas Assitido por Micro-Computador"; in Anais XIII Congresso Nacional de Processamento de Dados, SUCESU; 1980; pp 309-320
- Mazzoni, C.J.  
Um Modelo para o Controle de Qualidade de Software; Dissertação de Mestrado, Departamento de Informática, PUC/RJ, Rio de Janeiro; 1981
- Rocha, A.R.C.; Staa, A.v.  
 "Software quality in the requirements analysis and specification phase"; em Fourth Israel Conference on Quality Assurance; Herzlia on Sea, Israel; outubro 1982;

Secretaria Especial de Informática

Relatório da Comissão de Software e Serviços; SEI, Brasília; 1980

Staa, A.v.

"Gerador de Filtros Léxicos"; em 5o. Seminário Integrado de Hardware e Software Nacionais; Universidade Federal do Rio de Janeiro, Rio de Janeiro; julho 1978

Staa, A.v.

Engenharia de Programas; Livros Técnicos e Científicos, Rio de Janeiro; 1983

Staa, A.v.

"Qualidade de Software"; em XV Congresso Nacional de Informática; SUCESU, Rio de Janeiro; outubro 1982

Staa, A.v.; Cowan, D.L.

"The Development Proposal: The First Step in Software System Construction"; em: The Journal of Systems and Software vcl. 1; Elsevier North-Holland, New York; 1979; pags. 107-120



## Apêndice A.

### Laboratório de Software.

Apresentamos a seguir, em linhas gerais, o Laboratório de Software do Departamento de Informática da PUC/RJ. O sistema de apoio ao desenvolvimento de software é um dos projetos em andamento neste laboratório.

Ao contrário de laboratórios convencionais, este laboratório é geograficamente distribuído, bastando para o seu uso o acesso a um micro computador ou, então, o acesso a um terminal ligado a um dos processadores do laboratório.

Os objetivos do Laboratório de Software são:

- i- desenvolvimento de instrumental de apoio ao desenvolvimento de software;
- ii- estudo experimental de 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, programação, documentação, arquivamento de dados, certificação, teste e 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 deem o embasamento necessário ao desenvolvimento de software de qualidade controlada;
- iv- desenvolver protótipos e modelos que permitam a condução de experimentos e a predição de 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.