COPPEAD/UFRJ

RELATÓRIO COPPEAD No. 245

DESENVOLVIMENTO DE SISTEMAS: um estudo sobre medidas de produtividade

Geraldo Jorge Caputo *
Antonio R. R. Nogueira **
Donaldo de Souza Dias ***

Maio de 1991

* Analista de Sistemas da EMBRATEL. Mestre em Administração pela COPPEAD/UFRJ.

** Professor Pesquisador da COPPEAD/UFRJ.

*** Pesquisador Sênior do Centro Científico da IBM Brasil.

1 INTRODUÇÃO

A comunidade de processamento de dados, principalmente seus gerentes, tem dado maior atenção ao processo de desenvolvimento de sistemas face ao crescimento do custo de software, ou seja, do custo relativo ao desenvolvimento de sistemas. Este custo tem crescido tanto em termos absolutos como percentualmente em relação ao orçamento total de processamento de dados (KEMERER, 1987).

As técnicas da engenharia de software tentam reverter esta tendência. Para tanto, existe, hoje, um grande interesse no estudo e na aplicação de medidas que possibilitem quantificar diversos aspectos relacionados ao desenvolvimento de sistemas.

O objetivo central deste estudo é determinar, dentre algumas das principais métricas existentes, qual a que melhor explica a produtividade identificada no desenvolvimento de sistemas na EMBRATEL S/A.

Para atingir este objetivo foram aplicados alguns modelos, que implementam tais medidas, sobre um conjunto de 27 projetos desenvolvidos na EMBRATEL.

Pretendeu-se, desta forma, contribuir não só com o entendimento sobre a implementação de métricas de software, mas também com o conhecimento adquirido sobre as próprias métricas, dando maiores condições gerenciais de planejamento e controle do

processo de desenvolvimento.

2 MEDIDAS DE PRODUTIVIDADE

Do ponto de vista pragmático, o estudo sobre medidas de produtividade no desenvolvimento de sistemas é importante pelas seguintes razões: primeiro, para estimar o esforço (conseqüentemente o custo) de se desenvolver um sistema e, segundo, para aumentar a produtividade no desenvolvimento dos sistemas, uma vez que vários fatores relacionados à produtividade estarão sendo controlados.

Pressman (1987) aponta cinco razões pelas quais um software é medido: para indicar a qualidade do produto; para determinar a produtividade das pessoas que desenvolvem o produto; para determinar os benefícios (em termos de produtividade e qualidade) derivados dos novos métodos e ferramentas da engenharia de software: para formar uma base de informações para as estimativas; e para ajudar a justificar as necessidades de novas ferramentas ou de treinamento adicional.

Segundo Kemerer (1987, op.cit.), os custos sempre crescentes, associados ao desenvolvimento de software, têm direcionado os pesquisadores no sentido de obterem melhor entendimento sobre o processo de desenvolvimento, bem como de construírem e avaliarem ferramentas de estimativa de custos de software.

O interesse sobre métricas relacionadas ao desenvolvimento de software não é recente. Côté et al. (1988) apontam o artigo de Rubney e Hartwick (1968), publicado em 1968, como o mais antigo a tratar do assunto, sendo que os fundamentos para a métrica de sofware foram estabelecidos na década de 70. A partir destes, muitos outros resultados foram obtidos nos anos 80. Ramamoorthy et al. (1986) explicam como as métricas de software assumirão um papel importante nas futuras metodologias de desenvolvimento.

Basili e Zelkowitz (1978) definiram cinco importantes fatores que influenciam a produtividade no desenvolvimento de um software:

- pessoais - o tamanho e a experiência da organização que desenvolve o software.
- do problema - a complexidade do problema a ser resolvido e o número de mudanças nos requisitos e nas restrições do projeto.
- do processo - as técnicas de análise e de projeto utilizadas, linguagens disponíveis e procedimentos de revisão.
- do produto - confiabilidade e desempenho do sistema desenvolvido.
- dos recursos - disponibilidade de ferramentas para desenvolvimento e de recusos de hardware e de software.

A preocupação de tantos pesquisadores com respeito ao processo de desenvolvimento e aos fatores que decididamente afetam a produtividade associada a este processo enfatiza a crescente importância dada ao estudo das medidas de produtividade.

Foi escolhido um contexto específico, dentro da EMBRATEL, para serem avaliados alguns modelos que implementaram métricas de produtividade, visando determinar qual delas melhor se correlaciona com o esforço de desenvolvimento no referido contexto.

3 A ESCOLHA DOS MODELOS

O estado da arte na pesquisa de modelos que implementam medidas de produtividade ou do esforço aplicado no desenvolvimento de sistemas e no estudo dos fatores que influenciam tal atividade é discutido por Kemerer (1987, op.cit.), Abdel-Hamed e Madnick (1987), Adrangi e Harrison (1987), Gilb (1986), Jeffery (1987), Boehm e Papaccio (1988), e Lind e Vairavam (1989). Muitos destes estudos foram realizados sob a forma de pesquisa de campo.

Pressman (1987, op.cit.) estabelece a seguinte categorização para diversas métricas existentes:

- métricas de produtividade que focalizam a eficiência do processo de engenharia de softwares;

- métricas de qualidade fornecendo uma indicação de quão próximo o projeto chegou em relação aos requisitos do usuário, implicítos e explícitos;

- métricas técnicas que observam mais as características do software (por exemplo: complexidade lógica e grau de modularidade) que o processo de desenvolvimento do mesmo;

- métricas orientadas para o tamanho, utilizadas na coleta de medidas diretas dos produtos ("output") e da qualidade da engenharia de software;

- métricas orientadas para a função, que podem ser consideradas medidas indiretas; e

- métricas orientadas para os aspectos humanos, as quais recolhem as informações sobre o modo de desenvolvimento utilizado pelas pessoas e as percepções humanas da efetividade das ferramentas e dos métodos que lhe são colocados à disposição.

Evans, Piazza e Dolkas destacam a grande quantidade de fatores que podem influenciar na produtividade relativamente ao desenvolvimento de software (Evans et al., 1983). Estes fatores

são agregados em três grupos, enfatizando que o processo de desenvolvimento de um software é altamente sensível ao ambiente total no qual está sendo desenvolvido. Apesar de considerarem que podem existir centenas de itens envolvidos neste processo, eles apresentam alguns fatores mostrando a realidade complexa relacionada à produtividade:

- Ambiente Técnico: metodologia do projeto, técnicas e ferramentas do projeto, requisitos de confiabilidade, requisitos de desempenho, interface da aplicação, ambiente do usuário, ambiente de hardware, requisitos de teste, habilidade de desenvolvimento, definição dos requisitos, definição das interfaces e complexidade da aplicação.

- Ambiente do Projeto: ambiente de apoio (suporte), estrutura do projeto, fluxo de trabalho, demonstração das tarefas, espírito de solidariedade, controle do projeto, objetivo do projeto, desenvolvimento do projeto, interpretação das tarefas, desenvolvimento da gerência e overhead técnico.

- Motivação do pessoal: objetivos, eficiência, satisfação, apoio ao projeto, interesse, comprometimento e motivação.

Os estudos já citados conduzem a um grande número de fatores relacionados com a medida do esforço efetivamente realizado no desenvolvimento de sistemas. A identificação de tantos fatores que influenciam na produtividade das equipes de

desenvolvimento de sistemas dificulta a análise estatística e a conseqüente interpretação dos resultados, uma vez que vários destes fatores freqüentemente estão correlacionados.

Adicionalmente, a maioria dos pesquisadores sugere que os fatores considerados nos modelos que medem a produtividade no desenvolvimento de sistemas sejam ajustados, de modo a refletirem, com maior precisão, o ambiente de desenvolvimento de sistemas na empresa (DeMarco, 1982; Drumond, 1985; Symons, 1988).

A escolha dos modelos, além da relevância apontada por citações de diversos autores de artigos e/ou livros, considerou a diversidade dos fatores que influenciam a produtividade. Deste modo, foram selecionados modelos que tratam de diferentes fatores na composição desta influência e que servem de base para estudos realizados nesta área.

Assim sendo, os modelos Análise de Pontos de Função (Albrecht, 1979) e COCOMO (COnstructive COst MOdel) (Boehm, 1981), considerados como modelos clássicos por Côté et al.(1988), foram dois dos escolhidos. A métrica apresentada no Modelo de Análise de Pontos de Função, além da contagem dos pontos de função, considera 14 características para estabelecer o Fator de Complexidade Técnica (FCT) imposto sobre a equipe de desenvolvimento. Já no Modelo COCOMO são analisados 15 Atributos Direcionadores de Custo, os quais são diferentes dos referenciados pelos Pontos de Função. O terceiro Modelo selecionado foi o proposto por DeMarco (1982), denominado,

genericamente, de BANG. Na utilização de tal modelo considerou-se como premissa o fato de todos os projetos constituintes do estudo serem orientados para dados, uma vez que pertencem ao grupamento de sistemas de apoio à gestão das atividades da empresas. Esta premissa balisou a utilização específica da métrica BANGDADOS para sistemas orientados para dados.

Incluíram-se, também, nesta pesquisa os resultados oriundos da avaliação dos sistemas a partir do número de Linhas de Código Fonte (LCF). Tal medida, apesar de não se constituir em um modelo isoladamente, tem servido de base para diversos modelos e é considerada uma das medidas de software mais familiares dentre as existentes (CONTE et al., 1986).

Adicionalmente, sobre a medidas baseadas no número total de linhas de código fonte foi aplicado um fator de correção, conforme proposto por Jones (1988), de modo a equalizar as diferenças existentes entre as linguagens COBOL e NATURAL, utilizadas nos projetos da amostra. Tal correção simplesmente duplicava o número de linhas de código fonte escritas em NATURAL.

4 SÍNTESE DOS MODELOS

Os modelos básicos (Pontos de Função, COCOMO e BANG) comportam-se de modo semelhante quanto à sua aplicação. Todos estabelecem um valor inicial para o porte de um sistema,

ajustando este valor por diferentes critérios a fim de obterem uma medida final.

O modelo de Análise de Pontos de Função, conforme definido por Albrecht (1979, op.cit.), parte da contagem dos pontos de função existentes nas entradas externas, nas saídas externas, nos arquivos lógicos internos, nos arquivos de interface e nas consultas externas. Após a contagem destes componentes de um sistema, ponderados pela complexidade da função que atua sobre eles, o resultado (PFNA - Pontos de Função Não-Ajustados) sofre uma correção baseada no total do grau de influência registrado a partir de 14 características gerais do sistema: comunicação de dados, funções distribuídas, desempenho, configuração usada pesadamente, taxa de transações, entrada de dados "on-line", eficiência do usuário final, atualização "on-line", processamento complexo, reaproveitamento, facilidade de instalação, facilidade de operação, "sites" múltiplos e mudanças facilitadas. Cada característica recebe um valor conforme seu nível de influência (zero, caso a característica não esteja presente, até cinco, quando está presente e exerce grande influência todo o tempo). O resultado fornece, então, o número de PONTOS DE FUNÇÃO do sistema.

O modelo COCOMO Básico utiliza o número de linhas de código fonte do sistema como valor inicial. Em seguida, aplica uma de suas equações, geradas empiricamente por Boehm (1981, op. cit.) dependendo do modo de desenvolvimento adequado ao sistema (orgânico, difuso ou restrito), que fornece o número de HOMEM-MES

NOMINAL necessários ao seu desenvolvimento. Este valor se constitui na medida do tamanho do sistema para o modelo COCOMO Básico.

Já o modelo COCOMO Intermediário representa uma extensão do COCOMO Básico, uma vez que, sobre a quantidade de HOMEM-MES NOMINAL, calculada neste, é aplicada uma correção considerando 15 atributos direcionadores de custo: confiabilidade requerida do software, tamanho da base de dados, complexidade do produto, tempo de execução, memória principal, volatilidade da máquina virtual, tempo de resposta para a equipe de desenvolvimento, capacidade dos analistas, experiência da equipe na aplicação, capacidade dos programadores, experiência da equipe na máquina virtual, experiência na linguagem de programação, uso de técnicas modernas de programação, uso de ferramentas de software e prazo requerido para o desenvolvimento. Esta operação gera o valor final do COCOMO Intermediário para o tamanho do sistema, que é o HOMEM-MES AJUSTADO.

Finalmente, o modelo BANG, no cálculo do BANGDADOS, utiliza o modelo de objetos de um sistema, sobre o qual são contados os objetos, ponderando esta contagem segundo a complexidade da implementação dos objetos em função de seus inter-relacionamentos.

O Anexo sumariza as informações básicas dos modelos.

5 METOLOGIA DA PESQUISA

5.1 PERGUNTA DA PESQUISA

Esta pesquisa pretendeu identificar, no ambiente específico de desenvolvimento de sistemas na EMBRATEL S/A, que medida de produtividade apresentou a maior correlação com o esforço dispendido no desenvolvimento de sistemas. Neste sentido, determinou-se o porte (tamanho) dos projetos estudados, segundo cada modelo, contrapondo-os com o esforço efetivamente realizado para desenvolvê-los.

5.2 HIPÓTESES

Para responder à pergunta da pesquisa, estabeleceram-se hipóteses cujo objetivo é avaliar o grau de relacionamento existente entre o esforço realizado pela equipe de desenvolvimento de um sistema e o tamanho do sistema gerado, segundo cada medida apresentada.

HIPÓTESE 1

O esforço realizado no desenvolvimento de um sistema não está associado ao número total de linhas de código fonte do sistema.

HIPÓTESE 2

O esforço realizado no desenvolvimento de um sistema não está associado ao número total de linhas de código fonte do sistema corrigido pela tabela de Carpers Jones.

HIPÓTESE 3

O esforço realizado no desenvolvimento de um sistema não está associado ao tamanho do sistema conforme medido pelo modelo COCOMO Básico.

HIPÓTESE 4

O esforço realizado no desenvolvimento de um sistema não está associado ao tamanho do sistema conforme medido pelo modelo COCOMO Básico, tendo como base o número total de linhas de código fonte corrigido pela tabela de Carpers Jones.

HIPÓTESE 5

O esforço realizado no desenvolvimento de um sistema não está associado ao tamanho do sistema conforme medido pelo modelo COCOMO Intermediário.

HIPÓTESE 6

O esforço realizado no desenvolvimento de um sistema não está associado ao tamanho do sistema conforme medido pelo modelo COCOMO Intermediário, tendo como base o número total de linhas de código fonte corrigido pela tabela de Carpers Jones.

HIPÓTESE 7

O esforço realizado no desenvolvimento de um sistema não está associado ao tamanho do sistema conforme medido pelo modelo de Análise de Pontos de Função.

HIPÓTESE 8

O esforço realizado no desenvlvimento de um sistema não está associado ao tamanho do sistema conforme medido pelo modelo BANG.

5.3 UNIDADE DE ANÁLISE

A população do estudo constituiu-se de sistemas desenvolvidos na EMBRATEL S/A.

A EMBRATEL - Empresa Brasileira de Telecomunicações S/A se enquadra no setor público de prestação de serviços, atuando em todo o Brasil, com sede no Rio de Janeiro. Os serviços de telecomunicações prestados pela EMBRATEL têm abrangência tanto doméstica quanto internacional.

A seleção dos sistemas integrantes deste trabalho atendeu aos seguintes critérios básicos:

- o sistema deveria possuir documentação completa, especialmente do "projeto operacional", ou "projeto do produto" e "projeto detalhado", conforme Boehm (1988, op.cit.), fundamental na determinação dos PONTOS DE FUNÇÃO;
- o sistema deveria possuir o diagrama de objetos, fundamental na determinação do BANG;
- o responsável pelo desenvolvimento do sistema deveria ser facilmente contatado, permitindo o levantamento de informações complementares relativas aos modelos COCOMO e

PONTOS DE FUNÇÃO.

O segundo requisito foi o grande limitador da amostra. Pode-se dizer que isto deveu-se ao fato de que a função de Administração de Dados existe efetivamente na empresa desde 1985.

Assim sendo, dos 218 projetos catalogados na biblioteca de sistemas existentes na empresa, somente 31 possuíam diagramas de objeto. Destes, 4 não atendiam a pelo menos uma das outras condições básicas, resultando 27 projetos em plenas condições de integrarem a unidade de análise.

6 ANÁLISE DOS RESULTADOS

6.1 DIAGRAMAS DE DISPERSÃO

A observação dos diagramas de dispersão objetivou, caso possível fosse, identificar, através da análise gráfica, a medida de tamanho de sistemas de melhor correlação linear com o esforço realizado no desenvolvimento dos projetos.

A análise destes diagramas forneceu subsídios para concluir que várias medidas possuem "bom" grau de correlação com o esforço realizado. No entanto, não foi possível determinar a melhor apenas pela análise gráfica.

Por outro lado, observou-se que a medida BANGDADOS mostrou o menor nível de correlação.

6.2 REGRESSÃO LINEAR

As hipóteses foram efetivamente testadas a partir da análise de regressão linear simples sobre cada uma das medidas de tamanho de sistema (variável independente) em relação ao esforço realizado no seu desenvolvimento (variável dependente).

Neste teste utilizou-se o coeficiente de correlação R. As distribuições t de Student foram usadas como estatísticas de teste visando à rejeição das hipóteses nulas a um nível de significância de 1%. O valor crítico de t a este nível de significância, com 25 graus de liberdade, é de 2,485.

Os valores da regressão foram obtidos através do procedimento "REGRELIN" do programa estatístico SAEG, da Universidade Federal de Viçosa, MG.

A TABELA 1 sumariza os valores obtidos a partir da regressão linear efetivada para cada uma das hipóteses.

TABELA 1

Resultados das regressões lineares

HIPÓTESE	VARIÁVEL INDEPENDENTE	R	SIGNI-FICÂNCIA	t	R2
1	LCF	0,69622	0,0000	4,84952	0,48473
2	LCFC	0,60261	0,0004	3,77558	0,36314
3	CB	0,64935	0,0001	4,26926	0,42165
4	CBC	0,53824	0,0019	3,19320	0,28970
5	CI	0,72551	0,0000	5,27097	0,52636
6	CIC	0,65943	0,0000	4,38581	0,43484
7	PF	0,48840	0,0049	2,79845	0,23853
8	BD	0,13713	0,2476	0,69219	0,01880

Pelos valores da TABELA 1, embasado nos procedimentos estabelecidos para teste das hipósteses, concluiu-se:

HIPÓTESE 1 (H_o: $R_{LCF} = 0$)

Rejeitada. O valor de t calculado (4,84952) excedeu o valor crítico de t (2,485). Deste modo, R é estatisticamente diferente de zero, ou seja, existe um relacionamento linear, estatisticamente significativo, entre o número total de linhas de código fonte de um sistema (LCF) e o esforço efetivamente realizado para desenvolvê-lo (HMREAL).

HIPÓTESE 2 (H_o: $R_{LCFC} = 0$)

Rejeitada. O valor de t calculado (3,77558) excedeu o valor crítico de t (2,485). Deste modo, R é estatisticamente diferente de zero, ou seja, existe um relacionamento linear, estatisticamente significativo, entre o número total de linhas de código fonte de um sistema corrigido pela tabela de Carpers

Jones (LCFC) e o esforço efetivamente realizado para desenvolvê-lo (HMREAL).

HIPÓTESE 3 (H_o: R_{CB} = 0)

Rejeitada. O valor de t calculado (4,26926) excedeu o valor crítico de t (2,485). Deste modo, R é estatisticamente diferente de zero, ou seja, existe um relacionamento linear, estatisticamente significativo, entre o tamanho do sistema (HOMEM-MES NOMINAL) conforme medido pelo modelo COCOMO Básico (CB) e o esforço efetivamente realizado para desenvolvê-lo (HMREAL).

HIPÓTESE 4 (H_o: R_{CBC} = 0)

Rejeitada. O valor de t calculado (3,19320) excedeu o valor crítico de t (2,485). Deste modo, R é estatisticamente diferente de zero, ou seja, existe um relacionamento linear, estatisticamente significativo, entre o tamanho do sistema (HOMEM-MES NOMINAL) conforme medido pelo modelo COCOMO Básico, tendo como base o número total de linhas de código fonte, corrigido segundo Jones (CBC) e o esforço efetivamente realizado para desenvolvê-lo (HMREAL).

HIPÓTESE 5 (H_o: R_{CI} = 0)

Rejeitada. O valor de t calculado (5,27097) excedeu o valor crítico de t (2,485). Deste modo, R é estatisticamente diferente de zero, ou seja, existe um relacionamento linear, estatisticamente significativo, entre o tamanho do sistema (HOMEM-MES AJUSTADO) conforme medido pelo modelo COCOMO

Intermediário (CI) e o esforço efetivamente realizado para desenvolvê-lo (HMREAL).

HIPÓTESE 6 (H_o: R_{LCF} = 0)

Rejeitada. O valor de t calculado (4,38581) excedeu o valor crítico de t (2,485). Deste modo, R é estatisticamente diferente de zero, ou seja, existe um relacionamento linear, estatisticamente significativo, entre o tamanho do sistema (HOMEM-MES AJUSTADO) conforme medido pelo modelo COCOMO Intermediário tendo como base o número total de linhas de código fonte, corrigido segundo Jones (CIC) e o esforço efetivamente realizado para desenvolvê-lo (HMREAL).

HIPÓTESE 7 (H_o: R_{PF} = 0)

Rejeitada. O valor de t calculado (2,79845) excedeu o valor crítico de t (2,485). Deste modo, R é estatisticamente diferente de zero, ou seja, existe um relacionamento linear, estatisticamente significativo, entre o tamanho do sistema (PONTOS DE FUNÇÃO) conforme medido pelo modelo de Análise de Pontos de Função (PF) e o esforço efetivamente realizado para desenvolvê-lo (HMREAL).

HIPÓTESE 8 (H_o: R_{BD} = 0)

Não foi rejeitada. O valor de t calculado (0,69219) localizou-se no interior do intervalo do valor crítico de t (entre -2,485 e 2,485). Deste modo, não se pode afirmar que R seja estatisticamente diferente de zero, ou seja, não existe um relacionamento linear, estatisticamente significativo ao nível

de 1%, entre o tamanho do sistema (BANGDADOS) conforme medido pelo modelo BANG (BD) e o esforço efetivamente realizado para desenvolvê-lo (HMREAL).

7. CONCLUSÕES

Numa primeira análise dos resultados obtidos pela aplicação dos diversos métodos, foi avaliada a existência ou não de uma associação entre o tamanho (porte) do sistema, conforme cada método, e o esforço realizado para desenvolvê-lo.

Observou-se que somente no modelo BANGDADOS (Hipótese 8) não houve condições de se rejeitar a hipótese nula. Uma vez que as hipóteses 1 a 7 demonstraram existir correlação entre tamanho e esforço, a um alto nível de significância, pode-se concluir que: ou os modelos de objetos que serviram de base para o estabelecimento do BANGDADOS não foram contruídos de forma homogênea pelas equipes de desenvolvimento, ou a medida proposta por DeMarco (1982, op.cit.) não consegue sintetizar o conjunto de fatores que determinam o porte de um sistema. São necessários estudos adicionais que permitam firmar uma opinião conclusiva.

O segundo passo da análise dos resultados concentrou-se nas medidas que efetivamente indicaram a existência de uma associação entre tamanho e esforço, visando apontar a medida mais aderente à realidade da EMBRATEL. Neste sentido, foram comparados os coeficientes de determinação decorrentes das regressões

lineares realizadas nos testes das hipóteses 1 a 7 (TABELA 1). Concluiu-se, então, que a medida de tamanho de sistema especificada pelo modelo COCOMO Intermediário (R^2 = 0,52636) é a mais ajustada ao ambiente de desenvolvimento de sistemas da EMBRATEL, onde 52% da variação ocorrida no esforço efetivamente realizado para o desenvolvimento dos sistemas da empresa conseguem ser explicados pelo comportamento da variável CI, que representa o tamanho do sistema pela qualidade de HOMEM-MES AJUSTADO necessários ao seu desenvolvimento.

Este fato, porém, não garante que esta medida possa ser utilizada como previsora do tempo e dos recursos humanos necessários ao desenvolvimento de um sistema. Isto porque o item básico para a determinação da quatidade de HOMEM-MES AJUSTADO é o número total de linha de código fonte do sistema (LCF). E este número somente será obtido, com precisão, ao final do projeto, havendo, portanto, total dependência do feeling de quem está estimando este valor antecipadamente, com o objetivo de usar o modelo COCOMO Intermediário para realizar as previsões necessárias.

Associada ao estabelecimento prévio do número de linhas de código fonte está a definição da linguagem de programação a ser utilizada na construção dos sistemas. É bastante "razoável", conforme Jones (1988,op.cit.), que o número de linhas varie segundo a linguagem de programação em uso. Deste modo, os especialistas que farão as previsões devem levar em consideração tal fato. A dificuldade de termos este modelo como previsor

aumenta ainda mais se considerarmos que uma boa parte dos sistemas são construídos com mais de uma linguagem de programação (na amostra estudada, cerca de 60%).

Um resultado importante ocorreu com as hipóteses 2, 4 e 6, que tiveram o total de linhas de código fonte corrigido pela TABELA 14, levando em consideração as observações de Jones. Esperava-se, inicialmente, uma melhora nos valores de R2 obtidos nas hipóteses 1, 3 e 5, respectivamente. O que se registrou foi exatamente o contrário. Nos três casos reduziu-se o poder de explicação da variação do esforço efetivamente realizado no desenvolvimento dos sistemas.

Uma análise um pouco mais cuidadosa constatou que a correção efetuada, equalizando a linguagem NATURAL ao nível da linguagem COBOL, aumentou o porte do sistema desenvolvido, desconsiderando que o esforço para desenvolvê-lo contou efetivamente com uma ferramenta mais poderosa, que era a própria linguagem NATURAL.

Pesquisas futuras poderão verificar com maior profundidade a proposição de Jones neste contexto.

A observação das hipóteses 1, 3 e 5 revelou um interessante resultado. Na amostra considerada, a aplicação das equações fundamentais do COCOMO Básico sobre o número total de linhas de código fonte dos sistemas reduz a capacidade de explicação da variação do esforço realizado (48% na hipótese 1

e 42% na hipótese 3). Corrigindo os valores calculados no COCOMO Básico através dos atributos direcionadores de custo do COCOMO Intermediário, consegue-se aumentar o poder de explicação em 10% (de 42% na hipótese 3 para 52% na hipótese 5). Este fato é uma forte indicação de que os atributos direcionadores de custo do COCOMO Intermediário representam aspectos do desenvolvimento de sistema que realmente estão relacionados à produtividade das equipes.

O modelo de Análise de Pontos de Função proposto por Albrecht (1979,op.cit.), com grande aceitação em várias empresas, principalmente na IBM, apesar de bem-definido conceitualmente, foi o que menos se aproximou do ambiente de desenvolvimento da EMBRATEL (somente 23% da variação do esforço é por ele explicada). Como está calcada em informações que se tornam disponíveis mais cedo, no ciclo de vida de um sistema, esta medida poderá servir, se aperfeiçoada, como excelente previsor. Alguns trabalhos já têm sido elaborados criticando e sugerindo alterações no modelo proposto, objetivando dar maior consistência à contagem dos pontos de função (Symons, 1988, op.cit.). Neste sentido, sugere-se que estudos posteriores aprofundem a questão.

Evidentemente, as conclusões obtidas neste estudo se restringem ao conjunto de projetos avaliados, não podendo ser estendidas. Uma vez que cada empresa proporciona à sua área de desenvolvimento de sistemas um contexto organizacional específico, tais conclusões serão de grande valor para a EMBRATEL. Dentro deste contexto, o trabalho ora apresentado

poderá ser reaplicado e, somente então, extrair-se-ão conclusões válidas para a empresa.

Espera-se que este estudo contribua decisivamente na melhoria da gerência de desenvolvimento de sistemas nas empresas brasileiras, não somente pela revisão atualizada da literatura sobre o assunto em questão, como também pela metodologia apresentada no teste das diversas medidas. Entretanto, é sabido que ainda existe um longo caminho a percorrer no aprofundamento do tema, de modo a que benefícios reais possam ser observados pelas empresas.

ANEXO

Caracterísiticas dos modelos escolhidos

CARACTERÍSTICAS	PONTOS DE FUNÇÃO	COCOMO	BANGDADOS
1- Autor/Ano	Allan Albrecht/1981	Barry Boeha/1981	Tom DeMarco/ 1982
2- Base para coleta de dados	-Entradas externas -Saídas Externas -Arquivos Lógicos -Arquivos de Interface -Consultas ("inquiries")	-Linhas de código fonte entregues ao usuário -Modo de desenvolvimento	-Diagramas de Objeto
3- Índice calculado inicialmente	Pontos de Função Não-Ajustados - PFNA's	HOMEM-MES Nominal - (HM)nos	Número de Objetos-OB
4- Fatores considerados no ajuste	. Comunicação de dados . Funções distribuídas . Desempenho . Configuração usada pesadamente . Taxa de transações . Entrada de dados "on-line" . Eficiência do usuário . Atualização "on-line" . Processamento complexo . Reaproveitamento . Facilidade de instalação . Facilidade de operação . "Sites" múltiplos . Mudanças facilitadas	. Confiabilidade requerida do . Software . Tamanho da base de dados . Complexidade do produto . Tempo de Execução . Memória Principal . Volatilidade da máquina virtual . Tempo de Resposta . Capacidade dos analistas . Experiência na aplicação . Capacidade dos prog. . Experiência na máquina virtual . Experiência na linguagem de prog. . Técnicas modernas de programação . Uso de ferramentas de software . Prazo requerido para o desenvolvimento	. Complexidade da implementação dos objetos em função de seus interrelacionados
5- Medida final ajustada	Pontos de Função - PF	HOMEM-MES Ajustado - (HM) ajust	BANGDADOS

8 BIBLIOGRAFIA

ABDEL-HAMID, T.K.; MADNICK, S.E. On the portability of quantitative software estimations models. Information and Management, v.13, n.1, 1-10, Ago. 1987.

ADRANGI, B.; HARRISON, W. Effort estimation in system development project. Journal of Systems Management, v.38, n.8, p.21-23, Ago. 1987.

ALBRECHT, A.J. Measuring application development productivity. In: IBM APPLICATIONS DEVELOPMENT SYMPOSIUM, Monterey, CA, out. 1979. Proceedings..., p.83-92.

BASILI, V.; ZELKOWITZ, M. Analyzing medium scale software development. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 3, 1978. Proceedings..., p. 116-123.

BOEHM, Barry W. Software engineering economics. Englewfood Cliffs, NJ: Prentice-Hall, 1981.

_____; PAPACCIO, P.N. Understanding and controlling software costs. IEEE Transactions on Software Engineering, v.14, n.10, p.1462-1477, Oct. 1988.

CONTE, S. D. et al. Software engineering metrics and models. Menlo Park, CA: The Benjamin/Cummings Publ., 1986.

COTE, V. et al. Software metrics: an overview of recents results. The Journal of Systems and Software, v.5, p.121-131, 1988.

DEMARCO, Tom. Controlling software projects. Englewood Cliffs, NJ: Prentice-Hall, 1982.

DRUMMOND, S. Measuring applications development performance. Datamation, v.15, n.2, p.102-108, 1985.

EVANS, M.W. et al. Principles of productive software management. New York, NY: J. Wiley, 1983.

GILB, Tom. Estimating software attributes: some unconventional points of view. ACM SIGSOFT Software Engineering Notes, v.11, n.1, p.49-59, Jan. 1986.

JEFFERY, D.R. A software development productivity model for MIS environments. The Journal of Systems and Software, v.17, p.115-125, 1987.

JONES, Carpers. A new look at languages. Computerworld, Nov.7, 1988, p.97-103.

KEMERER, Chris F. An empirical validation of software cost estimation models. Communications of the ACM, v.30, n.5, p416-429, May 1987.

LIND, R.K.; VAIRAVAN, K. An experimental investigation of software metrics and their relationship to software development effort. IEEE Transactions on Software Engineering, v.15, n.5, p.649-653, May 1989.

PRESSMAN, Roger S. Software engineering: a practitioner's aproach. New York: McGraw-Hill, 1987.

RAMAMOORTHY, C.V. et al. Programming in the large. IEEE Transactions on Software Engineering, v.SE-12,p.769-783, 1986.

RUBEY, R.J.; HARTWICK, R.D. Quantitative measurement of program quality. In: ACM NATIONAL CONFERENCE, 1968. Proceedings..., p.671-677.

SYMONS, C.R. Function point analysis: difficulties and improvements. IEEE Transactions on Software Engineering, v.14, n.1, p.2-11, Jan. 1988.