

PAFUNCIO : UMA FERRAMENTA DE AVALIAÇÃO DE DESEMPENHO DE APLICAÇÕES

Claudia Baizer Medeiros

DCC - IMECC - UNICAMP - CP 6065

13081 Campinas - SP

Luiz Antonio Iaderoza

Departamento de Sistemas de Informação - IBM Brasil

Fábrica Sumaré - Rod. Campinas Monte-Mor Km 9

SUMARIO

Uma das preocupações constantes de equipes de desenvolvimento de software é a avaliação de desempenho. A área de métrica de software oferece várias alternativas para esta avaliação, que raramente são utilizadas devido à sua complexidade teórica. Este artigo descreve uma medida prática — a de índices de pontos de função — e apresenta uma ferramenta desenvolvida para automatizar sua utilização.

1. INTRODUÇÃO

A literatura sobre desempenho de software está repleta de métricas para avaliar a complexidade de um software. O livro de Boehm [B81] dá uma boa visão teórica da área. Uma das métricas mais empregadas é a de contagem de linhas de código, devido à sua simplicidade. Ela é usada para quantificar não apenas tamanho de um sistema, mas também a produtividade do pessoal envolvido em seu desenvolvimento. É pouco razoável, no entanto, medir produtividade ou complexidade em função simplesmente do tamanho de programas (vide, por exemplo, [G86]).

Como alternativa para métricas de "tamanho", surgiu em 1979 a medida das "funções" do usuário que um software deve executar, que reflete a relação entre os dados produzidos e consumidos por um sistema — o "function point index", aqui chamado de índice de pontos de função, ou simplesmente pontos de função. Este índice independe da tecnologia utilizada, já que é baseado na contagem de funções atendidas por um software, quaisquer que sejam suas características de implementação. É possível, inclusive, definir critérios através dos quais se pode determinar o número de linhas de código por ponto de função, como discutido por Albrecht e Gaffney [AG83]. Os autores mostram a equivalência entre esta métrica e a de Halstead [H77], que estabelece uma correspondência entre o número de símbolos de um programa e sua complexidade.

Na métrica de pontos de função, as funções do usuário são multiplicadas por pesos estabelecidos empiricamente, resultando no índice final. A partir do momento em que vários sistemas são avaliados por esta métrica, os pesos estimados passam a representar fatores adequados da complexidade de cada classe de aplicações.

O mesmo tipo de enfoque (definição de complexidade utilizando pesos derivados empiricamente) é encontrado em várias outras métricas. Davis e Addleman [DA86], por exemplo, apresentam o Escore de Metodologia (Methodology Score) para atribuir índices de produtividade a metodologias de especificação e com isto selecionar a mais adequada para cada tipo de problema.

Este artigo apresenta a métrica de pontos de função e descreve uma ferramenta – Produtividade Associada a FUNÇÕES (PAFUNCio) – desenvolvida para seu suporte. Esta ferramenta pode ser usada por vários grupos de desenvolvimento simultaneamente, para avaliação do desempenho do software produzido, bem como na previsão de custos de projetos futuros. Seguem-se três seções: a seção 2 descreve a metodologia de pontos de função; a seção 3 apresenta a ferramenta desenvolvida para suporte da metodologia e exemplos de seu uso; e a seção 4 contém conclusões.

2. O INDICE DE PONTOS DE FUNÇÃO

A métrica de pontos de função, desenvolvida por Alan Albrecht [AG83], é um método para caracterizar o tamanho e a complexidade de aplicações. A produtividade de um grupo de desenvolvimento é expressa em termos do número de horas de trabalho necessárias para gerar um ponto de função. Quanto maior o valor do índice, maior o número de funções implementadas e portanto maior a complexidade do software. Dentro um mesmo período de aferição, o grupo com maior produtividade é aquele que atinge o maior índice.

O cálculo dos índices para uma aplicação é baseado na contagem de suas entradas, saídas e consultas, bem como tipos de arquivos e interfaces para outras aplicações. À medida em que os dados de projetos são coletados, as comparações entre os sistemas se tornam mais significativas. Assim, projetos que utilizam ferramentas específicas – como um editor interativo especial – podem ser contrastados com aqueles que não dispõem destas ferramentas. Da mesma forma, é possível comparar projetos desenvolvidos segundo linguagens distintas, ou por grupos de características contrastantes. Por exemplo, a coleta de estatísticas sobre sistemas segundo esta métrica já atingiu um ponto em que se pode prever o número de pontos de função gastos ao utilizar uma determinada linguagem para cerca de 50 linguagens diferentes [J87].

Em outras palavras, pontos de função fornecem uma estratégia de medidas que pode ser usada fora do ambiente de desenvolvimento. Por exemplo, grupos distintos de desenvolvimento de software podem ser comparados a partir do índice de pontos de função atingido. A medida de pontos de função apresenta as seguintes características:

- é baseada na visão externa do usuário da aplicação. Não depende da tecnologia, o que permite escolha da tecnologia usada internamente sem mudança da medida.

- as contagens podem ser determinadas ao fim da especificação funcional, o que permite que pontos de função sejam usados para estimar o custo do desenvolvimento;

- pode ser entendida e avaliada por usuários sem sofisticação técnica na revisão e avaliação do produto.

Para entender as medidas de produtividade, é necessário estabelecer parâmetros de esforço de desenvolvimento, baseado em tempo decorrido. Este tempo pode ser o tempo total contado a partir da análise de requisitos ou o tempo realmente dispendido na atividade (descontando ausências, feriados, etc.). A experiência mostra que o valor do tempo real corresponde a 75% do total [A683].

Deve-se manter um histórico de pontos de função para detectar tendências futuras de produtividade. A partir deste histórico, não apenas se pode prever os custos de desenvolver um determinado produto, mas também constatar problemas em sistemas que ultrapassem de muito a previsão. Behrens [B83] mediu aplicações ao longo de dois anos e sugeriu que o tamanho de um projeto e o ambiente de desenvolvimento são fatores determinantes na produtividade: em média, ambientes online reduziram o custo de um projeto em 9.25 horas por ponto de função, enquanto o aumento em 10 pontos de função aumentou o custo de um projeto em 0.43 horas por ponto de função. Em outro exemplo, estatísticas coletadas pela companhia Hallmark durante três anos [D85] permitem afirmar que um projeto nesta companhia custa 0.8 pontos de função por dia.

Num estudo mais amplo das aplicações desenvolvidas na IBM em 1984, Albrecht [A85] apontou os seguintes resultados:

- . produtividade média de 21.5 pontos de função por mês (num total de 24100 pontos implementados em 11200 homens/mes de trabalho). A medida variou entre diversos centros IBM de 11 a 65 pontos por mês;

- . a manutenção de 1 115 000 pontos de função instalados custou um esforço anual de 6200 homens/mes de trabalho, correspondendo a uma unidade média de trabalho de 0.72 horas anuais de manutenção para cada ponto de função instalado.

A obtenção do valor do índice segue os seguintes passos:

- identificar e contar o número de funções de usuário em um sistema, multiplicando-as por fatores de complexidade pre-estabelecidos, que indicam o nível de processamento em que são empregados pelo usuário (baixo, médio ou alto);

- ajustar o resultado de forma a considerar fatores que influenciam a complexidade do processamento, definidos a partir de características gerais da aplicação, chegando-se, assim, ao índice de pontos de função para a aplicação. Neste segundo passo, leva-se em consideração o impacto de fatores como, por exemplo, atualizações em tempo real, ou distribuição de processamento.

A classificação e contagem de funções, primeiro passo do algoritmo, é realizado classificando cada uma das cinco funções a seguir segundo seu nível de complexidade (alto, médio ou baixo). Detalhes sobre o valor dado a este nível são discutidos por Behrens [883]. As funções consideradas são aquelas disponíveis ao usuário durante o desenvolvimento e teste da aplicação. Parte-se do pressuposto que existe uma fronteira que define os limites entre a aplicação e o mundo externo. O qualificativo "interna (externa)" para cada tipo de função salienta sua localização relativa à fronteira (processada pelo software versus utilizada por outras aplicações). A contagem, baseada em funções que atravessam esta fronteira, segue as seguintes regras:

Entradas externas (entradas que atravessam a fronteira modificando dados): contar cada tipo de dado ou de controle que vindo do lado externo da camada adiciona ou muda dados internos. Um tipo de dado difere dos demais por seu formato ou pela forma como é processado. Não são incluídos na contagem dados pertencentes a transações do usuário ou a outras transações. Não devem ser incluídos consultas ou tipos introduzidos para satisfazer a tecnologia usada. A classificação em simples, médio ou complexo varia de acordo com o volume esperado de cada tipo.

Saídas externas (geradas pelo sistema) contar cada tipo que atravessa a fronteira. Incluir relatórios, mensagens e arquivos enviados a outras aplicações. Para relatórios, a classificação de complexidade depende do número de colunas e correlações exigidas entre seus dados. Arquivos de saída e respostas a consultas não devem ser considerados.

Arquivo lógico interno (dados gerados e mantidos pela aplicação, não necessariamente armazenados ao seu final): contar cada grupo de dados ou de controle processado internamente pela aplicação. Incluir todos os dados que, sob o ponto de vista do usuário, sejam gerados e usados internamente. Não incluir dados que não sejam visíveis ao usuário através de consultas.

Arquivo de interface (criado apenas para ser enviado a outro sistema ou lido de outro sistema): arquivos passados ou compartilhados entre aplicações devem ser considerados na contagem de cada uma delas. Arquivos gerados internamente e enviados para outra aplicação devem ser contados duas vezes: como arquivos lógicos e como arquivos de interface.

Consulta externa (par entrada/saída sem modificações de dados): contar cada combinação de entrada/saída. Um tipo de consulta é considerado diferente dos demais se diferir no formato de entrada ou de saída ou na lógica de processamento. O nível de complexidade de cada consulta deve considerar as complexidades da entrada e da saída, sendo escolhida a maior das duas.

Uma vez classificadas e contadas as funções, deve-se realizar novos ajustes para considerar a influência de 14 tipos de características do sistema. Cada característica recebe uma nota de 0 a 5, de acordo com seu grau de influência no sistema (0 corresponde a influência nula e 5 a influência considerável). As características consideram:

- volume de dados e informação de controle enviados ou recebidos através de redes (comunicação)
- dados ou funções distribuídas como parte da aplicação
- objetivos de desempenho que influenciem o desenvolvimento
- utilização da aplicação em configuração com alto índice de uso
- frequência e volume de transações previsto
- funções de entrada online
- funções online visando eficiência do ponto de vista do usuário
- existência de atualizações online
- processamento complexo, como manipulação intensiva de equações, ou tratamento exaustivo de exceções para corrigir transações incompletas
- desenvolvimento de aplicações visando reusabilidade
- facilidade de conversão e instalação
- facilidade operacional, incluindo procedimentos para recuperação, e minimização de atividades manuais e consulta a manuais
- projeto leva em conta instalação em vários ambientes, ou utilização por diferentes espectros de usuários
- projeto visa suporte a facilidade de mudanças.

Convém fazer algumas observações:

- nem sempre a complexidade de um software pode ser medida exclusivamente em função destes índices. Por exemplo, um software que reaproveita código de outro projeto tem sua medida em pontos de função reduzida, em função da economia obtida em tempo de desenvolvimento. Para poder haver comparação com outros, pode ser preciso incorporar à medida final o valor dos pontos de função do código reutilizado;

- o número de linhas de código correspondentes a um ponto de função varia de projeto a projeto e não é possível fazer correlação direta entre os dois fatores, já que pontos de função independem da linguagem utilizada. No entanto, projetos com contagens de função altas requerem menos linhas de código por ponto de função.

• se o pessoal envolvido na produção de um determinado software precisa ser treinado em alguma atividade específica, o índice de pontos de função cai.

As experiências publicadas sobre a medida de pontos de função não revelam surpresas: protótipos mostram um índice maior de produtividade que aplicações desenvolvidas de forma tradicional; pacotes comprados fornecem um melhor índice de utilização que os desenvolvidos na empresa; funcionários que requerem treinamento custam mais a um projeto; projetos grandes parecem menos produtivos que pequenos projetos. O que é significativo é que pontos de função não contradizem as hipóteses feitas, o que fornece credibilidade a este tipo de medida. Quanto mais forem usados, maior a confiança que teremos neste tipo de indicador.

3. PAFUNCio – PRODUTIVIDADE ASSOCIADA A FUNÇÕES

A partir do exposto anteriormente, foi desenvolvida uma ferramenta (PAFUNCio) que suporte o acompanhamento de pontos de função por múltiplos grupos, com manutenção de vários tipos de histórico ao longo do desenvolvimento dos projetos. Esta ferramenta pode ser instalada em qualquer microcomputador do tipo PC-compatível.

Os pontos de função são medidos em dois momentos do ciclo de vida de um projeto, a saber:

- ao final da especificação funcional;

- ao final da implementação. Devem também ser atualizados durante a fase de manutenção do sistema.

A produtividade de desenvolvimento é medida contando os pontos de função adicionados ou mudados durante a implementação (ou seja, a diferença entre os valores constatados ao fim da especificação e ao fim da implementação). A produtividade de manutenção considera os pontos de função suportados após a instalação do software: o índice ao fim do desenvolvimento é ajustado de forma a considerar funções que hajam sido incorporadas ou eliminadas durante a manutenção. Este reajuste deve ser efetuado periodicamente.

O responsável pela especificação funcional de um sistema é quem realiza a contagem do número de ocorrências das cinco funções, definindo igualmente o grau de influência das características externas. Estes valores são fornecidos de forma interativa à ferramenta, que se encarrega de manter históricos de desenvolvimento separados por aplicação e por grupo de projeto. Estes históricos permitem controlar as alterações efetuadas a partir do início do desenvolvimento do projeto. Além de calcular o índice de pontos de função de um sistema, a ferramenta também permite calcular o esforço gasto em produzi-lo. O índice de pontos de função é fornecido tanto para a fase de desenvolvimento quanto para a fase de manutenção.

A ferramenta mantém um banco de dados sobre pontos de função e projetos, suportando as seguintes operações:

- manutenção de dados gerais sobre o ambiente de desenvolvimento;
- manutenção de dados sobre pesos empíricos observados;
- cálculo de índices de produtividade e de pontos de função tanto para desenvolvimento quanto para manutenção do software;
- estatísticas sobre um sistema específico;
- estatísticas sobre um grupo específico.

Os relatórios e consultas referentes às estatísticas podem ser tanto globais quanto se referir a um período de tempo específico. É possível, também, obter resultados comparativos entre diversos grupos de desenvolvimento ou entre sistemas. A interação com o usuário é feita através de menus auto-explicativos. O pre-requisito para o uso da ferramenta é estar familiarizado com a técnica, embora um resumo das suas principais características pode ser examinado através de telas de HELP.

As funções disponíveis permitem

A) Inserção de um código de sistema. Usada quando do início de desenvolvimento de um sistema, recebe um valor que identifica unívocamente o novo sistema a ser desenvolvido.

B) Alteração de dados de um sistema. Modifica os dados de um sistema específico no banco de dados, a partir da identificação de seu código.

C) Eliminação de um sistema específico do banco de dados.

D) Cálculo de índices de pontos de função. Calcula os índices de desenvolvimento ou manutenção. Os índices de desenvolvimento podem se referir tanto à primeira quanto à segunda contagem. No primeiro caso, o valor se refere à situação ao término da especificação funcional, sendo que as entradas são as contagens de função (por tipo) especificadas. A segunda contagem é feita quando o sistema está implementado. Neste caso, devem ser informadas as alterações efetuadas desde a contagem anterior. Para cálculo do índice de manutenção, o sistema usa como entrada o valor obtido na segunda contagem, e as funções cujos parametros foram alterados desde que esta contagem foi efetuada. Para isto é utilizado o histórico de alterações.

E) Emissão de relatórios. Podem ser por grupo de desenvolvimento ou por sistema, sendo que o período coberto pelo relatório pode ser opcionalmente indicado. Isto permite projeções e estatísticas, incluindo o cálculo de produtividade para um determinado grupo.

F) Pesquisa. Resultado de consulta sobre um determinado sistema ou grupo (índices; produtividade) ou sobre pesos atribuídos a um determinado fator.

6) Auxílio ao usuário. Conjunto de instruções que ensinam o usuário a utilizar a ferramenta, além de informações básicas sobre a métrica de pontos de função.

As páginas a seguir mostram cópias de algumas telas do PAFUNCIO.

4. CONCLUSOES

Este artigo apresentou a métrica de pontos de função como adequada para medir a produtividade de um software, descrevendo a implementação de uma ferramenta automatizada que suporta a utilização desta métrica por grupos de projeto. Uma vantagem indiscutível do método é a sua fácil aplicação e aprendizado, aliada à independência de tecnologia. Um dos resultados mais úteis do uso deste tipo de técnica é a habilidade de definir objetivos racionais de produtividade de software para desenvolvimento e manutenção.

Com esta ferramenta, pretende-se avaliar a produtividade de grupos de desenvolvimento de software formados por alunos da UNICAMP, bem como traçar objetivos a serem alcançados em termos de produtividade futura. Uma vez que se tenha pontos de função suficientes para vários projetos, estas medidas podem ser usadas para definir "benchmarks" de produtividade para as equipes. Analisando-se os diversos índices, poder-se-á identificar quais os grupos mais produtivos e inclusive observar quais linguagens, metodologias, ferramentas surtiram mais efeito. Os resultados obtidos deverão ser de grande valia não só do ponto de vista didático, mas também para permitir confronto de atributos que influenciam positiva ou negativamente os projetos desenvolvidos.

Outro benefício será a possibilidade de se usar esta ferramenta para estimar o custo de novos projetos, em função do histórico e da correspondência que se pode obter entre o índice de pontos de função e o esforço dispendido.

5. BIBLIOGRAFIA

[A85] Albrecht, A. "Function points helps managers assess applications". Computerworld, ago. 1985.

[AG83] Albrecht, A.J. e Gaffney, J.E. "Software function, source lines of code, and development effort prediction: a software science validation". IEEE Transactions on Software Engineering, SE-9(6), nov. 1983.

[B81] Boehm, B.W. "Software engineering economics". New Jersey, Prentice Hall, 1981.

[B83] Behrens, C.A. "Measuring the productivity of computer systems development activities with function points". IEEE Transactions on Software Engineering, SE-9(6), nov. 1983.

[D85] Drummond, S. "Measuring applications development performance". Datamation, 15(2), fev. 1985.

[DA86] Davis, M.J. e Addleman, D.R. "A practical approach to specification technology selection". Journal of Systems and Software, 6, 285-294, 1986.

[GA86] Gaffney, J.E. "The impact on software development costs of using HOL's". IEEE Transactions on Software Engineering, SE-12(3), mar. 1986.

[H77] Halstead, M.H. "Elements of software science". NY, Elsevier Press, 1977.

[J87] Jones, C. "Function-point metrics: key to improved productivity." InformationWEEK, fev. 1987.

FUNCTION POINTS - MEDICAO PRODUTIVIDADE EM DESENVOLVIMENTO DE SOFTWARE	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> ENTRADA/MANUT. A. Insere B. Atualiza C. Elimina </div> <div style="border: 1px solid black; padding: 5px;"> RELATORIOS F. Por grupo desenv. G. Por data especifica </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> CALCULO FP D. Desenvolvimento E. Manutencao. </div> <div style="border: 1px solid black; padding: 5px;"> CONSULTAS H. Educacao em F. Points I. Dados de um sistema Luis A. Iaderoza - RA.878364 MC 690 - Eng. Software. UNICAMP </div>
[Entre Opcao --> (A - I, ou X p/ terminar) :]:	

FPOINTS Calculo de F.Points para desenvolvimento release 1. Entre com o numero de funcoes para o sistema						
TIPOS DE FUNCAO		COMPLEXIDADE				
			BAIXA		MEDIA	
Input Externo		0		12		0
Output Externo		3		0		0
Arquivos Logicos		0		0		1
Arquivos de Interface		0		5		0
Funcoes Pesquisa		1		2		0

CARACTERISTICAS		GERAIS	
Comunicacao de dados	: 5	Funcoes distribuidas	: 1
Performance	: 4	Config. muito usada	: 3
Taxa de transacoes	: 2	Entrada dados online	: 5
Eficiencia Usuario	: 3	Atualizacao online	: 5
Processamento complexo	: 3	Utilizados outros sist.:	: 4
Facilidade implement.	: 3	Facilidade operacional	: 4
Multiplos locais	: 1	Facilidade mudancas	: 4

-----> Atualizacao efetuada OK <-----

MSG ==> Function points p/ software e' : 135.52
TECLE ENTER PARA CONTINUAR.

FPOINTS		PESQUISA DADOS POR SISTEMA	
Cod. Sistema	: MLPS	FP contagem 1	: 135.5
		FP contagem 2	: 169.1
		FP manutencao	: 384.1
Descricao	: MATERIAL LOG. PURCH. SYS.	Data contagem 1:	: 010280
Esforo desenv.:	: 20.00	Data contagem 2:	: 010480
Esforo manut.:	: 30.00	Data ult.contg.:	: 010280
Grupo Desenv.:	: 03		
---- Caracteristicas Gerais ----			
Comunicacao de dados:	: 5	Funcoes distribuida	: 1
Performance	: 4	Configuracao muito usada:	: 3
Taxa de transacoes	: 2	Entrada de dados online	: 5
Eficiencia usuario	: 3	Atualizacao online	: 5
Processam. complexo	: 3	Utilizavel outros sist.:	: 4
Facilidade implem.:	: 3	Facilidade operacional	: 4
Multiplos loc. Usuar:	: 1	Facilidade p/mudancas	: 4
MSG ==> Pesquisa para o sistema terminada			
Deseja outra Pesquisa/FIM (S/F)			