

ESTIMANDO OS PROJETOS COM COCOMO II

por Mauricio Aguiar

Neste artigo, abordaremos um assunto que, por sua própria natureza, exige algum tratamento matemático, embora *light*. A justificativa é que os benefícios alcançados através de sua utilização certamente compensarão o esforço envolvido: falaremos das técnicas para estimar projetos de software.

O problema das estimativas em projetos de software envolve, na maioria das vezes, a previsão de quatro variáveis: tamanho, esforço, prazo e qualidade. O esforço é normalmente obtido a partir do tamanho, o prazo e o custo a partir do esforço. Neste artigo, iremos tratar da estimativa das variáveis tamanho, esforço e prazo, deixando a qualidade para uma outra oportunidade.

Estimando o Tamanho do Projeto

As duas medidas de tamanho de software mais comuns são os Pontos de Função do IFPUG (PF) e as Linhas de Código (LOC – *Lines of Code*). Outras medidas têm sido consideradas, sem alcançar nível de utilização significativo.

Uma vantagem dos PF sobre as LOC é que os Pontos de Função podem ser obtidos logo no início do ciclo de vida, diretamente dos requisitos ou especificações [1]. Os PF são úteis para estimativas independentes de linguagem, realizadas no início do ciclo de vida [2]. Por outro lado, a utilização das LOC na previsão do esforço total de um projeto continua tendo sucesso para uma ampla quantidade de projetos, envolvendo diversas linguagens [3], conforme veremos adiante.

Devido às razões apontadas, a medida de tamanho mais utilizada nas estimativas iniciais dos projetos comerciais tem sido o Ponto de Função do IFPUG [4]. Um método de estimativa baseado em Pontos de Função bastante utilizado é a Contagem Indicativa da NESMA [5]. Neste método, são identificados os grupamentos de dados relativos à natureza do negócio, segundo a perspectiva do usuário. Tais grupamentos são classificados como: Internos (I), se mantidos pela aplicação objeto do estudo; ou Externos (E), se apenas referenciados ou consultados pela aplicação. Com base nas premissas da Análise de Pontos de Função (APF), a NESMA recomenda a seguinte fórmula para estimar o tamanho de uma aplicação, em Pontos de Função Não Ajustados (PFNA): **$PFNA = (35 * I) + (15 * E)$** . Note-se que os fatores 35 e 15 acima não foram obtidos através de qualquer procedimento estatístico, mas sim a partir de considerações sobre a relação entre os diferentes componentes do modelo da APF. Em uma seleção de 127 projetos do banco de dados do ISBSG [6], encontramos o coeficiente de correlação (R de Pearson) de 78,4% entre o tamanho real em PFNA e o estimado por uma aproximação da Contagem Indicativa NESMA.

Estimando o Esforço e o Prazo

Para estimar esforço e prazo, é preciso que seja selecionada uma abordagem para a obtenção de estimativas. As abordagens existentes podem ser divididas em [1]:

Modelos Paramétricos – Assumem a existência de uma relação matemática entre tamanho, esforço e prazo. Tal relação é afetada por parâmetros de performance. Os relacionamentos são baseados em suposições teóricas e/ou dados históricos. Exemplos de modelos paramétricos são COCOMO (*COnstructive COst Model*) e SLiM (*Software Life Cycle Model*) .

Modelos Baseados em Atividades – Também chamada estimativa *bottom-up*, esta modalidade consiste em enumerar todas as atividades do projeto e estimar o esforço e prazo para cada uma delas.

Analogia – Esta técnica baseia-se na comparação das características do projeto com a de outros projetos concluídos. As diferenças são identificadas, sendo introduzidas as mudanças necessárias para produzir as estimativas.

Relações Simples de Estimativas – Trata-se de uma simplificação dos modelos paramétricos. Neste caso, utilizam-se relações matemáticas simples, baseadas em dados históricos locais, ao invés de modelos matemáticos abrangentes. De uma forma geral, os relacionamentos deste tipo não são aplicáveis a organizações e contextos diferentes dos originalmente utilizados para a coleta dos dados. Exemplo: Estimar o esforço a partir de um modelo linear do tipo $\text{Esforço} = \text{Tamanho} \times \text{Produtividade}$.

Normalmente, faltam dados históricos que permitam a utilização de uma abordagem simplificada. Nesse caso, pode ser interessante optar por um modelo paramétrico. Os modelos paramétricos mais amplamente utilizados [7] para a determinação do esforço são o COCOMO (atualmente em sua segunda versão) e o SLiM. Tais modelos foram publicados [8,9] e são respeitados tanto no meio acadêmico quanto na indústria.

Ambos foram implementados através de diversos programas de computador e são suportados pelos respectivos autores. Dentre os dois excelentes modelos acima relacionados, preferimos considerar o COCOMO II (CII), pelas seguintes razões:

i) O CII foi e continua sendo desenvolvido em uma universidade, a *University of Southern California* (USC), enquanto o SLiM é um produto da empresa *Quantitative Software Management, Inc.* (QSM).

ii) Devido a ter sido desenvolvido por uma organização de caráter educacional, o CII é suportado por uma ferramenta de software gratuita, disponibilizada pela própria USC, enquanto o SLiM é implementado através de uma suite de ferramentas da QSM que custa milhares de dólares. Adicionalmente, existem várias implementações sofisticadas do CII, para aqueles que desejarem mais funcionalidade e estiverem dispostos a pagar o preço.

iii) Na USC o Projeto CII está a cargo do USC *Center for Software Engineering*, o qual é mantido por diversas organizações filiadas, dentre as quais a Rational. Como consequência, o Modelo COCOMO II é totalmente compatível com o RUP. Acrescente-se que o prefácio da referência [8] foi escrito por Walker Royce, Vice-Presidente da Rational.

Uma Breve Visão do COCOMO II

O Modelo COCOMO II teve como precursor o COCOMO, também conhecido como COCOMO 81 [10]. Devido à idade dos projetos que embasaram o modelo, assim como sua incapacidade de lidar com ciclos de vida iterativos e com a utilização de componentes

Commercial-Off-The-Shelf (COTS), o COCOMO 81 é atualmente considerado obsoleto, tendo sido substituído por sua versão II, publicada em 2000.

O COCOMO II aplicado ao RUP estima o esforço, prazo e equipe média para as fases de *Elaboration* e *Construction*. As fases de *Inception* e *Transition* são estimadas como percentuais da soma *Elaboration+Construction*. Os marcos que caracterizam o escopo abrangido pelo CII são:

LCO – *Lyfe Cycle Objectives* – Ponto no qual é escolhida uma possível arquitetura para o projeto (não necessariamente aquela que será de fato utilizada). Ocorre ao final da *Inception*.

IOC – *Initial Operational Capability* – Ponto no qual é concluído o desenvolvimento do software, estando o sistema pronto para entrega e teste final.

O CII produz estimativas para a região do ciclo de vida do RUP situada entre os marcos LCO e IOC. Estabelecido o escopo de estimativa do CII, vejamos como o modelo prevê o Esforço.

O CII define o Esforço (PM = *Persons-Month*), em pessoas-mês, como

$PM = A * (Size^E) * \text{Produtório } (i=1 \text{ até } n, EM(i))$ [a], onde

$E = B + 0,001 * \text{Somatório } (j=1 \text{ até } 5, SF(j))$ [b].

A = Constante que deve ser calibrada para o ambiente, a partir de dados históricos.

Size = Tamanho do software em LOC. Caso o tamanho seja dado em Pontos de Função, estimar a quantidade de LOC a partir de tabelas de *backfiring* [8], que permitem converter Pontos de Função para LOC.

EM(i) = *Effort Multipliers* = Até 17 fatores que irão adequar o modelo ao projeto específico. A quantidade de fatores utilizados dependerá do momento no qual será feita a estimativa. No início do projeto e até o final da *Elaboration* utiliza-se o *Early Design Model*, com apenas 7 multiplicadores. Após a definição da arquitetura deve ser utilizado o *Post-Architecture Model*, com 16 multiplicadores. Os multiplicadores referem-se a fatores relativos ao produto, à plataforma, ao pessoal e ao projeto. Um fator freqüentemente considerado em separado é aquele que diz respeito à compressão do prazo de desenvolvimento, ou seja, a exigência de que o projeto seja concluído antes do que seria o prazo normal. O significado de cada um dos fatores e a orientação para a sua determinação podem ser encontrados em [8].

E = Expoente do Esforço, dado pela fórmula [b] acima.

B = Constante que deve ser calibrada a partir de dados históricos.

¹ A notação “*Produtório* ($m=1$ até n , $F(m)$)” indica o produto de todos os elementos $F(m)$, m variando de 1 a n , inclusive. A notação “*Somatório*” tem utilização análoga. O circunflexo (“^”) denota a operação de exponenciação.

SF(j) = *Scale Factors* = Cinco fatores que irão ajustar o expoente do Esforço, de modo a adequar a fórmula a um projeto específico. Deve-se tomar cuidado ao ajustar estes fatores, uma vez que o impacto dos mesmos sobre as estimativas é exponencial.

Por sua vez, o Prazo (TDEV), em meses, é calculado como

$TDEV = C * PM^F$ [c], onde

$F = D + (0,2 * 0,01) * \text{Somatório } (j=1 \text{ até } 5, SF(j))$ [d]

e **C** e **D** são constantes que devem ser calibradas a partir de dados históricos. Quando cabível, deverá ser aplicada compressão de prazo sobre a fórmula [c] acima.

A Equipe Média é obtida através da divisão do Esforço pelo Prazo. O CII considera um mês equivalente a 152 horas de trabalho, já excluídos fins de semana e feriados. Este valor pode ser modificado para um ambiente específico.

Calibração do Modelo

O Modelo COCOMO II foi originalmente calibrado com dados de 161 projetos. Os mesmos foram selecionados dentre mais de 2000 projetos candidatos. Para cada um dos 161 projetos escolhidos foram realizadas entrevistas e visitas, a fim de garantir a consistência das definições e suposições do modelo. O modelo nominal vem calibrado para esses projetos, cuja natureza pode diferir daquele que se deseja estimar.

Embora o CII possa ser executado com os parâmetros nominais, sua correta utilização pressupõe a calibração para o ambiente-alvo. Na ausência de dados históricos disponíveis para o ambiente-alvo em questão, devem ser selecionados projetos equivalentes para efetuar a calibração. Os dados históricos selecionados devem ser validados antes de sua utilização, alimentando-os no software escolhido, calculando os coeficientes calibrados e, posteriormente, verificando se a diferença percentual (**estimado** – **real**)/**estimado** encontra-se compatível com o nível de erro pretendido para as estimativas. Devido ao seu impacto exponencial, não é recomendável calibrar os coeficientes B e D nas equações [b] e [d] acima quando houver menos de 10 projetos disponíveis para a calibração [8].

Roteiro para Estimativa

A seguir, fornecemos um roteiro simplificado para a obtenção de estimativas com o COCOMO II no início do projeto, utilizando o software USC CII: (1) Calibrar o CII para a base histórica de projetos obtida, salvando os coeficientes obtidos em um modelo; (2) Caso a organização-alvo não trabalhe com LOC, estimar o tamanho do aplicativo em Pontos de Função, utilizando o método da NESMA, obtendo a quantidade de LOC a partir de *backfiring*, utilizando a tabela contida no USC CII; (3) Determinar os 7 *Effort Multipliers* do Early Design Model e entrar com eles no USC CII; (4) Determinar o Esforço, o Prazo e a Equipe Média; (5) Obter a distribuição do Esforço e Prazo por fase do projeto (*Inception*, *Elaboration*, *Construction* e *Transition*, se for utilizado o RUP – notar que o USC CII também trabalha com o modelo *Waterfall* - Cascata).

Conclusão

Embora a utilização de modelos paramétricos exija um tratamento matemático algo sofisticado, tais modelos são os recursos mais poderosos para a obtenção de estimativas nos projetos de software. Dentre os existentes, destacam-se o COCOMO II e o SLiM, estando disponíveis, no caso do primeiro, software e documentação gratuitos. Cabe alertar, no entanto, que a efetiva utilização de tais modelos deverá ser precedida de preparação adequada e cuidadoso estudo.

Referências

- [1] McGarry, J. et al. – *Practical Software Measurement* – Addison-Wesley, 2002.
- [2] Royce, W. – *Software Project Management – A Unified Framework* – Addison-Wesley, 1998.
- [3] Roetzheim, W.H. e Reyna, A.B. – *Software Project Cost & Schedule Estimating* – Prentice-Hall, 1998.
- [4] International Function Point Users Group – <http://www.ifpug.org>
- [5] Netherlands Software Metrics Association – <http://www.nesma.nl>
- [6] ISBSG – *ISBSG Data Disk – Version 7 – International Software Benchmarking Standards Group*, 2001.
- [7] Futrell, R.T. et al. – *Quality Software Project Management* – Prentice-Hall, 2002.
- [8] Boehm, B. et al. – *Software Cost Estimation With COCOMO II* – Prentice-Hall, 2000.
- [9] Putnam, L. e Myers, W. – *Measures For Excellence – Reliable Software On Time, Within Budget* – Prentice-Hall, 1992.
- [10] Boehm, B. – *Software Engineering Economics* – Prentice-Hall, 1981.

URLs

www.psmc.com - PSM Support Center – site sobre *Practical Software Measurement*
www.ifpug.org – *International Function Point Users Group* – sobre Pontos de Função
www.bfpug.com.br – *Brazilian Function Point Users Group* – sobre Pontos de Função
sunset.usc.edu/research/COCOMOII/ - site do COCOMO II na USC, de onde pode ser baixado o USC CII
www.nesma.nl – site da NESMA
www.isbsg.org.au – site do ISBSG

Mauricio Aguiar é Diretor da **ti MÉTRICAS** e representa o **PMI-ISSIG** no Rio de Janeiro.

Esta coluna é produzida especialmente para a *Developers' Magazine* pelo ISLIG-Rio - Iniciativa Local do PMI-ISSIG, Grupo de Interesse em Sistemas de Informação do PMI.