

GERÊNCIA DE PROJETOS

Estimando os Projetos com COCOMO II no RUP

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. Aqui iremos tratar da estimativa de 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 eles podem ser obtidos logo no início do ciclo de vida, junto aos requisitos ou especificações. Os PF são úteis para estimativas independentes de linguagem, realizadas no início dos projetos. Por outro lado, o uso das LOC para a previsão do esforço total continua tendo sucesso para uma ampla gama de projetos, envolvendo diversas linguagens.

A medida de tamanho mais usada nas estimativas iniciais dos projetos comerciais tem sido o PF. Um método de estimativa baseado em PF usado é a Contagem Indicativa NESMA. Com base nas premissas da Análise de Pontos de Função (APF), a NESMA recomenda a 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 não foram obtidos através de um procedimento estatístico, mas a partir de considerações sobre a relação entre os componentes do modelo da APF. Em uma seleção de 127 projetos de BD do ISBSG, encontramos o coeficiente de correlação (R de Pearson) de 78,4% entre o tamanho real em PFNA e o estimado aproximado 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:

- Modelos paramétricos – Assumem a existência de uma relação matemática entre tamanho, esforço e prazo. Os relacionamen-

tos 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, consiste em enumerar todas as atividades do projeto e estimar o esforço e o prazo para cada uma delas.
- Analogia – Esta técnica baseia-se na comparação das características do projeto com as 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, usam-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 usados para a coleta dos dados. Exemplo: estimar o esforço a partir de um modelo linear do tipo $esforço = tamanho \times produtividade$.

Normalmente, faltam dados históricos que possibilitem o uso de uma abordagem simplificada. Em tais casos pode ser interessante optar por um modelo paramétrico. Os modelos paramétricos mais amplamente utilizados para a determinação do esforço são o COCOMO e o SLiM. Tais modelos foram publicados e são respeitados tanto no meio acadêmico quanto na indústria. Ambos foram implementados através de diversos programas de computador e são mantidos pelos respectivos autores. Dentre os dois excelentes modelos acima relacionados, preferimos considerar o COCOMO II (CII), pelas seguintes razões:

- O CII foi e continua sendo desenvolvido na University of Southern California (USC), enquanto o SLiM é um produto da empresa Quantitative Software Management, Inc. (QSM).
- Tendo 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 suíte 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.
- Na USC, o Projeto CII está a cargo do USC Center for Software Engineering, o qual é mantido por diversas organizações filiais, dentre as quais a Rational. Como consequência, o Modelo COCOMO II é totalmente compatível com o Rational Unified Process – RUP.

UMA BREVE VISÃO DO COCOMO II

O modelo COCOMO II teve como precursor o COCOMO, também conhecido como COCOMO 81. Devido à idade dos projetos que embasaram o modelo, assim como sua incapacidade de lidar com ciclos de vida iterativos e com o uso de componentes Commercial-Off-The-Shelf (COTS), o COCOMO 81 é considerado obsoleto, tendo sido substituído por sua versão II, de 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 – Life Cycle Objectives – Ponto no qual é escolhida uma possível arquitetura para o projeto (não necessariamente aquela que será de fato usada). 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 (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 uso análogo; o circunflexo (^) denota a operação de exponenciação): $PM = A * (Size^E) * \text{Produtório}(i=1 \text{ até } n, EM(i)) [a]$, 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 PF, estimar a quantidade de LOC a partir de tabelas de back-firing, que permitem efetuar a conversão.
- EM(i) = Effort Multipliers = Até 17 fatores que irão adequar o modelo ao projeto específico. No início do projeto e até o final da elaboration usa-se o Early Design Model, com apenas sete fatores. Após a definição da arquitetura deve ser utilizado o Post-Architecture Model, com 16 fatores. Os fatores referem-se ao produto, à plataforma, ao pessoal e ao projeto.
- E = Expoente do esforço, dado pela fórmula [b] acima.
- B = Constante que deve ser calibrada a partir de dados históricos.
- 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. A equipe média é obtida através da divisão do esforço pelo prazo.

CALIBRAÇÃO DO MODELO

O modelo COCOMO II foi originalmente calibrado com dados de 161 projetos. Os mesmos foram selecionados dentre mais

de dois mil projetos candidatos. Para cada um dos 161 projetos escolhidos foram realizadas entrevistas e visitas, para 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, seu correto uso pressupõe a calibração para o ambiente-alvo. Na ausência de dados históricos disponíveis para o ambiente em questão, devem ser selecionados projetos equivalentes para efetuar a calibração. Os dados históricos selecionados devem ser validados antes de seu uso, 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 dez projetos disponíveis para a calibração.

CONCLUSÃO

Embora o uso 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 o efetivo uso de tais modelos deverá ser precedida de preparação adequada e cuidadoso estudo.

REFERÊNCIAS

- International Function Point Users Group – www.ifpug.org
- Netherlands Software Metrics Association – www.nesma.nl
- ISBSG – ISBSG Data Disk – Version 7 – International Software Benchmarking Standards Group, 2001.
- Boehm, B. et al. – Software Cost Estimation With COCOMO II – Prentice-Hall, 2000.
- 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.