

COCOMO II - Um modelo para estimativa de custos de Gerência de Projetos

Pablo Ariel do Prado López¹

¹Bacharel em Informática, habilitação em Análise de Sistemas – Universidade do Vale do Rio dos Sinos (UNISINOS)

Caixa Postal 275 – 93.022-000 – São Leopoldo – RS – Brasil

pablo.ap.lopez@gmail.com

Resumo. Este artigo apresenta como o estabelecimento de estimativas se constitui em uma das principais atividades do planejamento do projeto de software. Propõe também, uma metodologia para a elaboração de estimativas utilizando o modelo Constructive Cost Model II (CoCoMo II) e apresenta o resultado da aplicação dessa metodologia considerando a comparação do custo estimado, o custo alcançado e o custo proposto pela metodologia em projetos concluídos, para justificar uma mudança de paradigma na elaboração de estimativas.

Abstract. This paper shows as estimates establishment constitutes one of the main activities of software project planning. It also considers a methodology for the elaboration of estimates using the model Constructive Cost Model II (CoCoMo II) and it is presented the result of the application of this methodology considering comparison of cost esteem for the company, the cost really reached and the cost considered for the methodology in finished projects, to justify a paradigm change in the estimates elaboration.

1. Introdução

Uma das funções do gerente de projetos de desenvolvimento de software é ter a capacidade de estimar e mensurar o custo, o tempo e esforço exigidos para os projetos de software a serem desenvolvidos ou em desenvolvimento. Para isto, ele deve conhecer a capacidade de sua equipe e os recursos com os quais pode contar para executar as atividades. Desta forma, adequando-se ao custo disponível e à qualidade desejada, o gerente poderá estabelecer prioridades (“trade-offs”)¹ para a realização dessas atividades.

Ao elaborar uma estimativa para o desenvolvimento de um projeto de software, é desejável que haja um conhecimento sobre técnicas de estimativas e uma visão global do projeto a ser gerenciado, o que capacita os profissionais da área de informática a quantificar, administrar e planejar mais efetivamente os sistemas a serem produzidos.

Planejamentos com base empírica podem tornar inviável o projeto de software, justamente por não atender requisitos básicos, que abordam as etapas do ciclo de

¹ Trade-offs são prioridades para a execução dos projetos; os mais utilizados são custo, tempo e qualidade, não necessariamente nesta ordem.

produção de software². As consequências se tornam visíveis no resultado final, em retrabalho³ durante a execução dos processos, ou o não atendimento das necessidades previstas e o atraso das atividades com o conseqüente atraso do projeto. Por esse motivo, é visível a necessidade de que as organizações realizem estimativas para o desenvolvimento de um projeto de software, para verificar a viabilidade do projeto, aumentando a competitividade e evitando perdas.

Na próxima seção, é apresentada a metodologia utilizada e o perfil de empresa a que estudado, na seção 3, o modelo *Constructive Cost Model (CoCoMo)* é brevemente apresentado, juntamente com suas características, inovações e limitações. O restante do artigo encontra-se assim organizado: na seção 4 é realizada uma identificação do problema de estimativas, na seção 5 são montadas as etapas de construção do modelo para as estimativas, a seção 6 apresenta uma simulação de um problema na elaboração de estimativas para que seja resolvido com o modelo CoCoMo II, na seção 7 é apresentado o produto a ser analisado e a aplicação do modelo é feita, e na seção 8, uma análise e comparação dos resultados alcançados é realizada, e por fim uma conclusão é apresentada na seção 9.

2. Metodologia de Pesquisa

A metodologia considerada mais apropriada ao presente artigo é a do estudo de caso qualitativo segundo Yin (1994), pois é considerado um método holístico para o desenvolvimento de pesquisas, e também permite que a contextualização do problema seja bem desenvolvida.

O perfil de empresa estudada caracteriza-se pela sua atividade principal ser a área de engenharia, utilizando a Tecnologia da Informação (TI) como ferramenta de suporte às suas atividades principais, utilizando normas e padrões de TI e aplicando modelos de engenharia de software para o planejamento e desenvolvimento dos seus processos de software, assim como no desenvolvimento de seus produtos de software. Desta forma o modelo CoCoMo II foi utilizado para avaliar o planejamento de desenvolvimento de produtos da empresa, comparando como todos esses processos são realizados atualmente, com o resultado da aplicação do modelo CoCoMo II.

Os critérios para escolha do perfil da empresa alvo para o estudo de caso seguem, segundo Gil (1994), o critério da conveniência, ou acessibilidade, no qual o pesquisador seleciona os dados a que tem acesso, e estes representam o todo dentro de sua pesquisa, ou estudo de caso, que é o realizado neste contexto.

3. Constructive Cost Model (CoCoMo)

O *Constructive Cost Model*, (CoCoMo), segundo Boehm (2000), é um método que busca medir esforço, prazo, tamanho de equipe e custo necessário para o

² Segundo Pressman (1995), as etapas básicas do ciclo de produção de software são: a análise de requisitos, projeto, desenvolvimento, testes e implantação / manutenção.

³ O retrabalho, de acordo com Pressman (1995), caracteriza-se como ciclos em torno da mesma atividade, seja de análise, projeto, ou desenvolvimento de software.

desenvolvimento do software, desde que se tenha a dimensão do mesmo, através de um modelo de estimativa de tamanho de software, como Análise de Pontos de Função.

Devido à idade dos projetos que embasaram o modelo anterior, assim como sua incapacidade de lidar com ciclos iterativos e com a utilização de componentes *Commercial-Off-The-Shelf* (COTS)⁴, o CoCoMo 81, Boehm (1981), é atualmente considerado obsoleto, tendo sido substituído pela sua versão II, publicada inicialmente em 2000 e vista em Aguiar (2004).

Para o cálculo do custo deve-se conhecer o prazo e equipe de trabalho, para então chegar ao valor, sendo que para definir o tamanho do programa, torna-se necessário que se caracterize que medida será adotada (linhas de código, pontos por função ou pontos por caso de uso).

O CoCoMo II estima o custo e tempo baseado em pessoas/mês e meses, respectivamente, para a determinação do *baseline*⁵ de exigências de um produto para a conclusão de uma atividade. O modelo ainda prevê um adicional de 20% ao tempo computado, como margem de erro (análise de risco).

O CoCoMo II, divide-se em três (3) sub-modelos que abordam as diferentes fases em que o projeto ou atividade que está sendo realizada se encontra, onde a utilização de cada sub-modelo aumenta a precisão e fidelidade da estimativa ao longo do processo de planejamento e execução de um projeto: O modelo da *Application Composition* é o mais apropriado para o estágio de prototipação no ciclo de vida espiral. O modelo *Early Design* é apropriado quando as exigências são conhecidas e as alternativas de arquitetura do software foram exploradas. O modelo *Post-Architecture* é o modelo o mais detalhado e envolve as etapas de construção real do software e de manutenção.

- ***Application Composition*** (prototipação): O modelo *Application Composition* utiliza *Object Points* como medida de tamanho. Para estimar os *Object Points* no ambiente *Integrated Computer-Aided Software Engineering* (ICASE), inicia-se estimando o número de telas, relatórios, e componentes em 3GL que a aplicação conterá. Categorize estes *Object Points* como simples, médios, e difíceis e atribua pesos a cada objeto. Calcule a contagem do *Object Point* somando todos os objetos da aplicação. Para estimar o esforço, estime primeiramente os *new object points* (NOP) necessários para o desenvolvimento. Avalie, então, a taxa da produtividade, na razão de NOP/pessoa-mês. Para finalizar, compute a estimativa de pessoa-mês.
- ***Early Design*** (arquitetura): No modelo *Early Design*, o CoCoMo II usa drivers de custo para capturar aquelas características do desenvolvimento

⁴ Comercial-off-the-shelf (COTS) são componentes de software prontos para utilização, de terceiros, comercialmente disponíveis, e que se tornam importantes durante a criação do novo software, devendo ser utilizados preferencialmente durante a fase de pré-desenvolvimento do produto (software), segundo Boehm (2000).

⁵ Uma baseline é um conjunto de produtos aceitos e controlados, e que serão utilizados em atividades posteriores à sua aceitação, segundo Leite (1997)

do software que afetam o esforço necessário para terminar o projeto. Os drivers do custo contêm os níveis de avaliação que definem o impacto do driver do custo no esforço do desenvolvimento. Estas avaliações variam de extremamente baixo a extremamente altamente. Cada nível de avaliação tem um peso associado. Este peso é denominado como multiplicador do esforço. O modelo adiantado do projeto está usado obter estimativas ásperas do custo e da duração de um projeto depois que as exigências estão disponíveis e antes que a arquitetura inteira esteja determinada. O modelo é baseado em pontos de função não ajustados convertidos a *KSLOC* (*Kilo Source Line of Code* – Milhares de Linha de Código Fonte) e utiliza uma porcentagem sobre o código que será descartado pelas alterações ou volatilidade dos requisitos para ajustar o tamanho.

- ***Post-Architecture*** (desenvolvimento): O *Post-Architecture* é o modelo mais detalhado do CoCoMo II. É usado depois que os requisitos são definidos e a arquitetura total do projeto é desenvolvida. Tem novos drivers de custo, novas regras para contagem de linhas de código, e novas equações, o que torna a estimativa mais precisa em relação aos outros modelos pela quantidade e precisão das informações disponíveis para elaboração da mesma.

A definição inicial do CoCoMo II vem sendo refinada com pesquisas e dados adicionais sobre projetos que são coletados e analisados pela equipe da USC. Podemos então definir os objetivos primários do CoCoMo II como sendo:

- Um modelo de estimativa de custo para o desenvolvimento de software de acordo com o modelo de ciclo de vida de software vista em Jones (1986), e as práticas de desenvolvimento da última década.
- Criar ferramentas de suporte capazes de fornecer melhoramentos do modelo, através de manutenção de informações sobre o desenvolvimento de software em uma estrutura de base de dados.
- Fornecer um *framework* analítico, e um conjunto de ferramentas e técnicas para avaliação dos efeitos de melhoria na tecnologia e nos custos despendidos no ciclo de vida de desenvolvimento de software.

Estes objetivos suportam as necessidades primárias da estimativa de custo, realizadas pela equipe da USC. Como prioridade, essas necessidades devem fornecer suporte para a etapa de planejamento de projetos, equipe de desenvolvimento necessária, preparação inicial do projeto, replanejamento, negociação de contratos, avaliações de propostas, necessidades e níveis de recursos (tecnológicos e humanos), exploração dos conceitos e avaliação de design. A seguir serão apresentadas as características do modelo CoCoMo II.

3.1. Calibração do Modelo

A calibração do modelo para um ambiente consiste no ajuste da variável “A”, que indica o percentual de reuso de código, que será vista na seção 3.4 deste artigo, da fórmula de esforço. O objetivo da calibração é obter a distribuição da produtividade e

atividade de desenvolvimento para um ambiente específico. O modelo CoCoMo II foi originalmente calibrado com dados de 161 projetos. Os mesmos foram selecionados entre 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 CoCoMo II possa ser executado com os parâmetros nominais, sua correta utilização pressupõe a calibração para o ambiente-alvo. Com a calibração em projetos do mesmo ramo é possível modificar o valor das constantes das fórmulas padrões. 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, 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 o modelo quando houver menos de 10 projetos disponíveis para a calibração, conforme Boehm (2000).

3.2. Fatores de Escala

Os modelos de estimativa de custo de software possuem um fator exponencial para considerar os gastos (despesas) e economias relativas as escalas encontradas em projetos de software de tamanhos distintos. Para determinar se o projeto apresenta gastos ou economias, uma variável é utilizada como expoente da equação de esforço, $b = 0,91 + 0,01 \times \sum SF_j$. A equação como um todo será vista com mais detalhes na seção 3.4 deste artigo.

A partir do valor de “b”, pode-se capturar os efeitos do projeto e chegar as seguintes premissas:

b < 1,0: O projeto apresenta economia de escala. Se por ventura o tamanho do produto dobra, o esforço relativo ao projeto é menor que o dobro. A produtividade do projeto aumenta à medida que aumenta o tamanho do produto. Pode-se alcançar algumas economias de escala do projeto com ferramentas de projeto específicas. Para projetos pequenos, fixar custos de saída, tais como reuniões periódicas e ainda relatórios administrativos, a rigor são uma fonte de economia.

b = 1,0: As economias e gastos estão em um nível de equilíbrio. Este modelo linear é utilizado a rigor para a estimativa de custo de projetos pequenos.

b > 1,0: O projeto apresenta gastos de escala. Isto se deve normalmente a dois fatores principais: o crescimento do gasto em comunicações e o gasto em crescimento da integração de um grande sistema. Os projetos maiores alocam mais recursos humanos, e conseqüentemente uma maior comunicação e interação interpessoal produzindo despesas. Integrar um produto pequeno como parte de um maior não requer apenas esforço de desenvolver o produto pequeno como também a despesa adicional de esforço para projetar, manter, integrar e testar suas interfaces com o resto do produto.

O expoente “b” é obtido através dos fatores de escala. A seleção dos fatores de escala é embasada na razão de que são recursos significativos de variação exponencial

de esforço ou variação da produtividade do projeto. Cada fator tem um intervalo de níveis de valores que vão desde Muito Baixo até Extra Alto. Cada nível de valores possui um peso, SF, e o valor específico do peso é chamado Fator de Escala. Um fator de escala de um projeto é calculado realizando o somatório de todos os fatores e é utilizado para determinar o expoente “b”.

3.3 Multiplicadores de Esforço

Os drivers de custo são utilizados para capturar características do desenvolvimento de software que afetam o esforço para completar o projeto. Os drivers de custo possuem um nível de medida que expressa o impacto do driver no esforço de desenvolvimento. Esses valores variam desde Extra Baixo até Extra Alto. Para o propósito de análise quantitativa, o nível de medida de cada driver de custo possui um peso associado. A este peso se dá o nome de Multiplicador de Esforço (EM). A medida atribuída a um driver de custo é 1.0 e o nível de medida associado com este peso é chamado de nominal. Se um nível de medida produz mais esforço de desenvolvimento de software, então seu correspondente EM possui valor acima de 1.0. Reciprocamente se o nível de medida reduz o esforço, então o correspondente EM possui valor abaixo de 1.0.

Os EM são utilizados para ajustar o esforço Pessoas/Mês nominal. Existem 7 (sete) multiplicadores de esforço para o modelo *Early Design* e 17 para o modelo *Post-Architecture*, e a sua influência dentro da fórmula de esforço dá-se da seguinte forma: $PM = A \times (Size)^b \times \prod EM_i^6$.

3.4 Realizando Medições através do CoCoMo II

O modelo CoCoMo II permite estimar e determinar o Custo, o Esforço e o Tempo de um projeto ou atividade de software a partir de um modelo de estimativa de tamanho do software. A particularidade neste modelo é que os pontos de função são considerados quando ainda não foram ajustados, ou seja, não é o resultado final da estimativa de tamanho de software, visto que na maioria dos casos é difícil determinar exatamente o número de linhas de código que o projeto ou a atividade conterá, isto é apresentado na figura 1. Isto acarreta que ambos os modelos – de estimativa de tamanho e CoCoMo II, sejam perfeitamente compatíveis e complementares, de acordo com Cocomo Manual (2004).

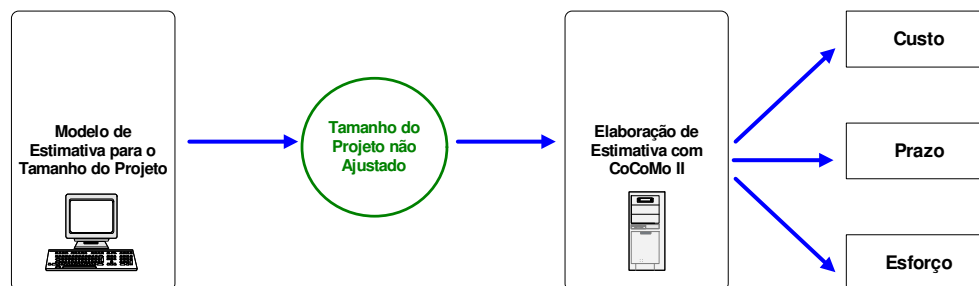


Figura 1. Esquema da estimativa realizada no CoCoMo II

⁶ Π significa produtório, que simboliza o total de diversas multiplicações.

A seguir serão apresentados os passos a serem seguidos para utilização do método CoCoMo II para se obter a estimativa do Custo, Esforço e Tempo para o projeto ou atividade a ser desenvolvida, o que, de acordo com a fase em que o projeto se encontra, utiliza-se o modelo adequado.

1º Passo – Calibrar o CoCoMo II para a base histórica de projetos obtida, salvando os coeficientes em um modelo; - esta etapa é comum a todos os tipos de projeto e aos 3 modelos.

2º Passo – Aplicar a fórmula para o cálculo do esforço de desenvolvimento, esta etapa é dependente do modelo de desenvolvimento que estiver sendo utilizado, e isto é definido de acordo com o tipo de projeto.

Utilizando o modelo *Application Composition*

- ✓ Computar o número de Pontos de Objeto (OP – *Object Points*) para o sistema inteiro;
- ✓ Estimar o percentual de reutilização de código e computar o número de Novos Pontos de Objeto (NOP – *New Object Points*) requeridos;
- ✓ Determinar a taxa de produtividade (PROD), que é o número de novos pontos objetos por mês que a equipe de projeto pode produzir;
- ✓ Estimar o esforço através da equação **$E = NOP/PROD$**

Utilizando o modelo *Early Design*

- ✓ Calcular pontos de função não ajustados (PFNA) para o sistema;
- ✓ Converter os PFNA para KSLOC, usando o fator de ajuste específico do ambiente de desenvolvimento;
- ✓ Ajustar a estimativa inicial através do conjunto de drivers de custo;
 1. Complexidade e confiabilidade do produto
 2. Reutilização requerida
 3. Dificuldade da plataforma
 4. Experiência da equipe
 5. Capacidade da equipe
 6. Facilidades disponíveis
 7. Programação
- ✓ Estimar o esforço “E” como: **$E = A \times KSLOC^b \times \Pi$** (drivers de custo)

Utilizando o modelo *Post-Architecture*

- ✓ A fórmula de estimativa para este modelo é dada pela equação:
 $E = a \times A \times KSLOC^b \times \Pi$ (drivers de custo);
- ✓ O fator “a” é determinado pelo tipo de projeto;
- ✓ O fator “A” determina o percentual de reutilização de código; e possui o valor de 2,94, conforme a calibração do CoCoMo II em Cocomo Manual(2004),

- ✓ O expoente “b” é derivado da soma dos 5(cinco) fatores de escala utilizando a fórmula: $b = 1,01 + 0,01 \times \sum W_i$,

Onde cada fator “W” possui um intervalo de valores entre 5 (baixo) e 0 (alto).
Os fatores são:

Precedência - Flexibilidade de desenvolvimento - Resolução do risco/arquitetura
- Coesão da equipe - Maturidade do processo

3º Passo – Aplicar o cálculo do tempo de desenvolvimento, e estimar o prazo.

$$TDEV = C \times \{PM^F\}$$

Onde:

C é uma constante multiplicativa para tempo de desenvolvimento.

(C = 3.67 conforme calibragem do CoCoMo II – Cocomo Manual(2004))

$$F = [D + 0.2 \times (B - 1.01)]$$

D é uma constante multiplicativa para tempo de desenvolvimento.

(D = 0.28 conforme calibragem do CoCoMo II – Cocomo Manual(2004))

B assume o mesmo valor da fórmula para cálculo de esforço – Somatório dos Fatores de Escala

PM é o esforço calculado

“TDEV” é expresso em quantidade de meses.

4º Passo – A Equipe Média é obtida através da divisão do Esforço pelo Prazo. O CoCoMo II considera um mês de trabalho equivalentes a 152 horas de trabalho, excluídos finais de semana e feriados. Este valor pode ser alterado para um ambiente específico.

$$P = PM/TDEV$$

Onde:

PM é o esforço calculado

TDEV é o prazo calculado

“P” é expresso em quantidade de homens

5º Passo – Cálculo do custo.

Com o prazo (TDEV) e Equipe Média (P) é possível estimar o custo do software, conhecendo, além dessas variáveis, o valor hora de cada integrante envolvido no projeto.

4. Identificação do Problema

Os principais problemas enfrentados por um Gerente de Projetos de Software estão relacionados aos processos de estimativa dos custos e prazos de entrega dos projetos.

Influenciando esses prazos e suas estimativas têm-se, ainda, a composição da equipe de projeto e desenvolvimento e o acompanhamento das etapas efetuadas pela

mesma. Some-se a isso as crescentes mudanças no mercado e as novas tecnologias disponíveis para se projetar e desenvolver softwares. Eis aí um ambiente competitivo, extremamente dinâmico, ao qual as empresas que desenvolvem software, mesmo que não seja o seu foco, devem se adaptar para continuarem competitivas e principalmente manter sua fatia de mercado, duramente conquistada.

O Gerente de Projetos de Software torna-se, portanto, figura fundamental na sobrevivência e crescimento dessas empresas no mercado. Porém estimar o prazo, custo e esforço de projeto e desenvolvimento de um software é um processo delicado e que envolve muita sensibilidade do Gerente de Projetos de Software e, principalmente, uma excelente capacidade de estimativa.

Por maior que seja a experiência do profissional atuando como Gerente de Projetos de Software, ele fatalmente irá deparar-se com projetos de software com os quais ainda não tenha trabalhado, nem mesmo com algo similar ou ainda, iniciar trabalhos com uma equipe da qual ele não conhece os seus membros e nem suas capacidades. A Engenharia de Software fornece, como ferramentas possíveis de auxílio, as métricas de software e métricas para projetos, ferramentas capazes de gerar dados para serem analisados e utilizados como apoio pelos Gerentes de Projetos de Software.

Dentro deste contexto, o problema que se apresenta é o fato que, com relação aos produtos de software e das atividades que fazem parte de um projeto, as mensurações e estimativas são realizadas seguindo um modelo empírico, baseado e embasado na percepção e experiência dos responsáveis pela fase de análise dos requisitos e também pelo conhecimento de quem realiza a atividade. Isso dificulta a obtenção de informações precisas sobre os projetos realizados, para efeitos de comparação do que foi estimado com o que foi realizado, bem como para a realização de estimativas para os novos projetos que serão realizados.

Para estes novos produtos, supõe-se a utilização de uma estimativa por semelhança, seguindo a percepção dos gerentes e analistas; de maneira totalmente empírica o gerente ou analista, ao receberem as especificações do projeto, levam-nas ao projetista e conjuntamente avaliam e discutem a melhor maneira de realizá-las, sendo que a definição da estimativa é realizada nesse momento.

5. Etapas de Construção do Modelo

O primeiro obstáculo a ser superado para que as métricas de software e as métricas de projetos sejam utilizadas, é a efetiva implementação dessas métricas nas empresas que desenvolvem projetos de software para gerarem os dados necessários e essenciais.

O segundo obstáculo a superar é a obtenção de uma ferramenta que auxilie o Gerente de Projetos de Software na compilação e análise dos dados gerados, que não são poucos. Tudo isso seria impossível se não existissem métricas de software e métricas para o projeto pelo meio qual o software é desenvolvido.

Basicamente, as métricas mostram dois indicadores básicos muito importantes:

- **Produtividade:** A produtividade depende de inúmeros fatores como a dimensão e complexidade dos sistemas a desenvolver, as linguagens de programação, o grau de reutilização ou a experiência e motivação dos participantes no processo de desenvolvimento. Para constatar uma variação

de produtividade resultante da influência desses fatores é necessário medir e quantificar a produtividade.

- **Qualidade:** A qualidade dos produtos de software é traduzida através de características como a correção, eficiência, confiabilidade, portabilidade ou facilidade de manutenção. A obtenção desses dados quantitativos relativos a essas características é assim fundamental para introduzir melhorias no processo de desenvolvimento.

Para realizar a modelagem do projeto, ou da atividade a ser realizada no projeto, a melhor maneira é utilizar as ferramentas e metodologias apropriadas, que nos permitem realizar essas estimativas de forma mais precisa. Dessa forma será possível obter melhores resultados e contar com um histórico para atividades subsequentes ou futuras.

6. Descrição do Caso – Metodologia de Elaboração de Estimativas

Nesta seção é apresentada a maneira como são desenvolvidas e realizadas atualmente as estimativas de custo e prazo dos projetos e atividades desenvolvidas na empresa “case” estudada.

1) O Gerente de Projetos de Software recebe do departamento de vendas, ou diretamente do cliente, através de um contato realizado, uma solicitação para criação de um novo projeto, ou solicitação de uma nova funcionalidade, ou para uma alteração de funcionalidade de um projeto existente.

2) O Gerente de Projetos, após analisar a viabilidade técnica da solicitação, repassa-a ao Analista de Sistemas, que, de posse dessas informações, elabora uma especificação funcional mais detalhada da solicitação, repassando-a ao Projetista de Softwares, para que elabore o projeto da solução e apresente o mesmo para o Analista.

3) O Gerente de Projetos, o Analista e o Projetista em comum acordo verificam quem dos recursos humanos disponíveis (essa disponibilidade não significa que o integrante não esteja realizando alguma atividade no momento, e sim em relação ao número de atividades e complexidade que os integrantes estejam desenvolvendo no momento), teria capacidade de realizar a atividade, definindo assim o número de pessoas que realizará a atividade.

4) De maneira informal, as pessoas envolvidas na atividade, Analista, Projetista e desenvolvedor(es), discutem as premissas necessárias para a atividade, que foram especificadas e definidas pelo projetista na etapa 2 do modelo, como tecnologia a ser utilizada, alterações e/ou adequações necessárias nas aplicações, bibliotecas e componentes adjuntas ao projeto e que podem interferir de alguma forma na realização da nova atividade.

5) A estimativa de definição de tempo é passada para o(s) desenvolvedor(es) envolvidos, que podem ou não concordar com a estimativa realizada e estipular um novo tempo de acordo com a sua percepção e repassar para o Gerente de Projetos e Analista, que vão analisar e retornar essa perspectiva, sendo que este ciclo permanece até que se chegue a um denominador comum entre todas as partes envolvidas nesse processo.

6) Após essa discussão, há a definição do custo do desenvolvimento, o qual pode ser definido seja por algum projeto ou atividade anteriormente realizado e que possa ter alguma similaridade, mas que geralmente são definidas através da percepção do Gerente de Projetos e do Analista, sendo que a estimativa de tempo é repassada para o Projetista para que seja repassada ao(s) desenvolvedor(es), enquanto a estimativa de custo é utilizada para realização de orçamento para o cliente.

6.1. Escopo das Estimativas Realizadas

Qualidade de Projeto é um conceito diferente de qualidade do produto. A literatura comumente aborda a qualidade do produto em TI ao passo que a qualidade do projeto é um tópico mais difícil de ser pesquisado. Qualidade de Projeto é um conceito amplo que engloba muitos aspectos da qualidade do produto. O que se deve ter em mente quando se considera a qualidade do produto é que a mesma se torna intangível sem um bom gerenciamento da qualidade do projeto, segundo Prado (2000).

Partindo desta premissa, os produtos que serão estimados neste estudo de caso, as estimativas serão referentes ao **custo dos produtos em questão**, e não da solução completa.

O custo da solução completa engloba outros aspectos que estão além do propósito deste trabalho, como por exemplo, as visitas realizadas, as etapas de análise, testes, implantação e manutenção do projeto, o custo do hardware que acompanha os produtos, materiais de marketing e materiais de treinamento, confecção e elaboração de manuais, os deslocamentos de pessoal a clientes para consultorias, visitas ou outras atividades, eventuais treinamentos que tenham de ser realizados para aprendizado de uma metodologia ou tecnologia, ou ainda, a contratação de pessoal ou terceirização de serviços.

É relevante ressaltar, também, que o custo estimado, não é o que está sendo repassado ao cliente final, e sim o custo que a empresa terá para o desenvolvimento do produto, este custo do produto será adicionado aos demais custos citados no parágrafo acima, sendo que este cálculo de custo total do projeto não está no escopo do presente trabalho.

7. Produto de Software Analisado

Para realizar as estimativas com o modelo CoCoMo II, foi utilizado um produto que já estava com seus requisitos e arquitetura totalmente concluídos, desta forma foi estimado utilizando o sub-modelo *Post-Architecture*, visto que por ser o mais detalhado dos sub-modelos, nos gerará uma estimativa mais precisa. O produto consiste em uma solução de gerenciamento de um sistema de automação, portanto pode ser considerado de complexidade alta.

7.1. Aplicação Prática do Modelo CoCoMo II

Para realizarmos o cálculo da estimativa, consideraremos os seguintes dados do produto de software, utilizando as equações já citadas na seção 3.4:

Total de Pontos de Função não Ajustados (PFNA): 1029

Encontrando número de KDSI através do ajuste de PFNA com o valor do peso da linguagem utilizada, Visual Basic® = 32: $(1029 * 32) / 1000 = 32,928$.

Encontrando o Esforço de Desenvolvimento

Calculando o valor de PM nominal através da equação $A \times (SIZE)^b$: 145 meses

Calculando o valor de PM ajustado através da equação $PM_{(nominal)} \times \Pi EM$: 133 meses

Encontrando o Tempo de Desenvolvimento

Calculando o valor de TEDV através da equação $3.67 \times PM_{(ajustado)}^{(0,28 + 0,2 \times (b - 1,01))}$: 16 meses

Encontrando a Estimativa de RH

Calculando o valor de P através da equação $PM_{(ajustado)} / TDEV$: 8 pessoas

Encontrando o Custo do Produto

$TEDV = 16$ meses

152 horas mensais (de acordo com Cocomo Manual (2004))

8 horas diárias

19 dias úteis mensais, descontando finais de semana e eventuais feriados.

Total de dias = 304

Total de horas = 2432 horas por integrante

2432×8 integrantes = 19.456

Custo aproximado da hora de desenvolvimento de cada integrante = 5

$19.456 \times 5 = 97.280$

Custo estimado = R\$ 97.280 (Noventa e sete mil e duzentos e oitenta reais)

8. Análise dos Resultados

Para realizarmos esta análise, devemos considerar o resultado estimado da forma atual, e o resultado encontrado, para que possa haver uma comparação dos resultados (estimado modelo X estimado atual X realizado).

A estimativa atual é realizada da seguinte forma:

Nº de dias estimados x nº de pessoas utilizadas na atividade x valor hora dos recursos x 8 horas diárias.

A tabela 1 apresenta um comparativo das estimativas realizadas.

Tabela 1 – Estimativa do produto de software através do modelo Post-Architecture

	Estimado Empresa	Estimado CoCoMo II	Realizado
Status produto			Concluído
Tempo (dias)	400	304	325

Esforço (nº pessoas)	8	8	8
Tamanho (KDSI)	Não aplicado	33.09	56.8502
Custo (em reais)	128.000	97.280	104.000

De acordo com as estimativas realizadas, a empresa previu que com 400 dias úteis de trabalho e 8 pessoas teria sido possível concluir o desenvolvimento do produto, a um custo de R\$ 128.000,00.

Pela estimativa elaborada no modelo, o ideal para o desenvolvimento do produto poderia ter sido realizado com 304 dias úteis de trabalho, e 8 pessoas a um custo de R\$ 97.280,00.

Visto que é um produto que já está concluído, é possível ter uma idéia mais concreta de como a utilização do modelo pode beneficiar as estimativas:

A estimativa de esforço teve a precisão mais acertada, visto que as estimativas e o realizado foram sem erros.

Em relação ao tempo de desenvolvimento, a defasagem entre estimado e realizado pela empresa foi de 22,07%, visto que o produto levou menos tempo de desenvolvimento do que o previsto, entretanto na comparação com a estimativa pelo CoCoMo II, a diferença entre estimado e realizado é de 6,46%, tempo este a mais que pode ter sido causado, por exemplo, pela alocação de recurso em outro produto, problemas pessoais, enfim, alguma eventualidade não prevista no cronograma.

No aspecto tamanho do produto, visto que a empresa não estima atualmente esta característica, não há maneira de comparar o estimado com o realizado, mas em relação à estimativa elaborada pelo modelo, a diferença é de 41%, e uma das razões principais é que há muita repetição de código nos diversos módulos do produto, o que justifica o fato de que não foi estimada a reutilização de código.

Em termos de custo, verificando os dados obtidos, verifica-se que a defasagem entre o estimado pela empresa e o efetivamente gasto com o produto foi de 23,07%, a empresa gastou menos do que estimou para o desenvolvimento, entretanto na comparação com a estimativa pelo CoCoMo II, a diferença entre estimado e o que foi gasto é de 6,46%, que seria uma economia considerável de custo se o projeto tivesse seu curso normal estimado, esta diferença de custo é em decorrência dos fatores tempo e tamanho do produto, que influenciaram de forma diretamente proporcional.

9. Conclusão

No decorrer deste artigo procurou-se explorar o modelo CoCoMo II, bem como as características e variáveis necessárias e que influenciam no sucesso da mudança de paradigma e implantação do modelo CoCoMo II. À luz de recomendações levantadas pelos autores na revisão bibliográfica, foram formulados os aspectos necessários para elaboração de estimativas com embasamento, que foram investigadas e aplicadas em um estudo de caso em uma empresa com perfil pré-definido e, finalmente, relatadas e discutidas na seção de análise de resultados.

Sobre o estudo de caso, pode-se inferir que grande parte das diferenças nos valores encontrados para as variáveis estudadas entre os produtos nas estimativas

realizadas e nos resultados encontrados, justifica-se pelo fato de que os processos de desenvolvimento de software ainda estão sofrendo os efeitos da padronização ora sendo adotada. Assim, por exemplo, reutilização e modularização, podem acarretar a repetição de instruções escritas nos diferentes produtos estudados, ao invés de um encapsulamento de rotinas comuns a todos os produtos.

Os fatores levantados nas referências bibliográficas e as evidências colhidas no estudo de caso levam à conclusão que o sucesso na mudança de paradigma e implantação de uma metodologia para elaboração de estimativas com embasamento teórico, neste caso o modelo estudado, o CoCoMo II, depende, dentre outros aspectos, das seguintes variáveis:

- Comparação de custo estimado pela empresa, o custo realmente alcançado e o custo proposto pela metodologia em projetos concluídos. Tal comparação é importante, pois, desta forma existe a possibilidade de verificar se há e onde há problemas nos processos de desenvolvimento de software, e desta forma ter a possibilidade de encontrar maneiras de melhorá-los ou atualizá-los.
- No aspecto tecnológico e de projetos, comparação do custo estimado de projetos em andamento ou em planejamento, não apenas em projetos concluídos, para justificar a mudança de paradigma na elaboração de estimativas. Isso viabiliza a possibilidade de aplicar a metodologia e o modelo CoCoMo II como forma de comparação e em projetos que estejam em andamento, possibilitando que haja uma noção real de como a estimativa pode trazer benefícios aos processos de estimativa e elaboração do custo dos produtos e projetos, pois estando em andamento é possível direcionar o andamento do desenvolvimento em relação à estimativa, ou seja, se for detectado algum desvio, pode-se solucionar em meio ao desenvolvimento (iteração), e não trabalhar apenas com hipóteses, que é o resultado da aplicação em projetos concluídos.

Essas foram as variáveis que se apresentaram como as mais relevantes na elaboração do estudo de caso. Entretanto, é possível haver outras variáveis, como por exemplo, a receptividade da equipe à mudança, que podem influenciar nos resultados de planejamento e desenvolvimento de software, apresentando desta forma resultados diferentes e estimativas alteradas.

10. Referências Bibliográficas

- Aguiar, Maurício (2004) “Estimando os Projetos com COCOMO II”. TIMétricas. <http://www.timetricas.com.br>, Maio.
- Boehm, Barry W. (1981) “Software Engineering Economics”. Prentice Hall.
- Boehm, Barry et al. (2000) “Software Cost Estimation With COCOMO II”. Prentice Hall
- Cocomo Manual (2004) “Cocomo II Model Definition Manual”. <http://sunset.usc.edu/research/COCOMOII>, Agosto.
- Gil, Antonio Carlos (1994) “Métodos e Técnicas de Pesquisa Social”. São Paulo. Atlas.

- Jones, Capers (1986) "A Short History of Functions Points and Feature Points Software Productivity Research". Inc. Beurlington MA
- Leite, J.C.S.P. et al. (1997) "Enhancing a Requirements Baseline with Scenarios. Requirements Engineering", Springer-Verlag London
- Prado, Darcy (2000) "Gerenciando Projetos nas Organizações", Belo Horizonte MG, EDG.
- Pressman, Roger S. (1995) "Engenharia de Software". São Paulo. Atlas.
- Yin, Robert K. (1994) "Case Study Research, Design and Models". California. Sage.