

## **Pontos de Casos de Uso Técnicos para Manutenção de Software**

**Yara Maria Almeida Freire, Arnaldo Dias Belchior**

Universidade de Fortaleza – Mestrado de Informática Aplicada  
Av. Washington Soares 1321, 60811-341 – Fortaleza – CE – Brasil

yara@bnb.gov.br, belchior@unifor.br

**Resumo.** *As atividades de estimativas de tamanho, esforço, prazo e custo têm alcançado cada vez mais importância, uma vez que estimativas confiáveis relacionadas ao projeto auxiliam bastante seu planejamento e monitoração. Este trabalho propõe estimativas de manutenção de software, desenvolvido a partir da abordagem de casos de uso, estendendo a técnica TUCP (Pontos de Caso de Uso Técnico), proposta para estimativas de projetos de desenvolvimento de sistemas.*

**Abstract.** *Activities to estimate size, effort, schedule and cost have been gaining more importance, once more reliable projects estimates are helpful on monitoring and plan. This work proposes a software maintenance estimates, developed based on the approach of use cases, extending the technique TUCP (Technical Use Case Points), proposed for systems development projects estimates.*

### **1. Introdução**

Ao longo do ciclo de vida de um sistema, seus requisitos originais sofrem manutenção, para refletirem as necessidades de seus clientes e as mudanças de seus usuários. A manutenção pode envolver a evolução, a adaptação ou a correção de artefatos do sistema.

No início dos anos noventa, estatísticas revelaram que muitas organizações alocaram, no mínimo, 50% de seus recursos financeiros em manutenção de software [April, 1995; Bourque *et al.* 1996]. Na década atual, a manutenção de software tem chegado a 70% dos custos de um sistema de software, tornando-se um dos grandes desafios da engenharia de software [Webster *et al.* 2004]. Isto evidencia a importância de pesquisas sobre manutenção, através das quais técnicas e métodos são criados ou metodologias existentes são adaptadas, para contemplar as atividades inerentes a este processo, como é o caso das estimativas de tamanho de um projeto de manutenção.

Segundo McPhee (1999), a estimativa de tamanho de software é um processo pelo qual uma pessoa ou um grupo de pessoas estima o tamanho de um produto de software. Para Ross (2004), o tamanho geralmente tem impacto na solução técnica e na gestão do projeto, já que estimativas imprecisas podem levar ao fracasso do projeto. Neste contexto, a precisão das estimativas de tamanho torna-se fundamental para a elaboração de cronograma e orçamento realistas, pois essas estimativas constituem-se na base para a derivação das estimativas de esforço, prazo, e custo [CMMI 2006].

Este trabalho propõe estimativas de tamanho, esforço, prazo e custo de manutenção de *software*, desenvolvido a partir da abordagem de casos de uso. Esse

trabalho estende a técnica TUCP (Pontos de Caso de Uso Técnico), proposta por Monteiro *et al.* (2005) para estimativas de projetos de desenvolvimento de sistemas. Essa técnica por sua vez já é uma extensão da UCP (*Use Case Point*) [Karner 1993].

Este trabalho está organizado da seguinte forma: na seção 2, discorre-se sobre estimativas de manutenção de *software*; na seção 3, descreve-se a proposta de extensão da TUCP para projetos de manutenção de *software*, a TUCP-M; na seção 4, são mostrados alguns casos reais de utilização da TUCP-M; finalmente, na seção 5, apresentam-se as conclusões.

## **2. Estimativas de Manutenção de Software**

Um processo de manutenção diz respeito a um conjunto de etapas bem definidas, que direcionam a atividade de manutenção de *software*, com o objetivo primordial de satisfazer as necessidades dos usuários de maneira planejada e controlada [Pigosky 1996].

Segundo a NBR ISO/IEC 12207 (1998), o objetivo do processo de manutenção de *software* é modificar um produto de *software* existente, preservando sua integridade. Esse processo é ativado, quando o produto de *software* é submetido a alterações, a partir do momento em que é necessário efetuar modificações no código e na documentação do sistema devido a um problema, adaptação ou necessidade de melhoria [Sousa, Oliveira e Anquetil 2005].

A grande maioria dos sistemas, após sua implantação, sofre alterações durante todo seu ciclo de vida, especialmente os sistemas que suportam a área fim de uma empresa. Estes são os que mais sofrem manutenção, em virtude de que sua substituição implica normalmente em um risco maior e um maior custo para a empresa. Por isso, as empresas optam por realizar manutenções de seus sistemas mais críticos em vez de simplesmente substituí-los. Em geral, os sistemas que sofrem manutenção têm seus requisitos originais alterados para atender uma nova legislação, pelo surgimento de novas funcionalidades, pela mudança de *hardware* ou do ambiente operacional, pela correção de erros ou pela prevenção de possíveis falhas.

As empresas têm despendido um tempo bem maior no esforço de manutenção de seus aplicativos do que no desenvolvimento de novos sistemas. Isto implica que também o custo de manutenção desses aplicativos tem sido muito superior ao custo do desenvolvimento de novos sistemas. Assim sendo, existe uma grande necessidade por técnicas de estimativas para projetos de manutenção de *software*, para que essas empresas possam gerenciá-los adequadamente.

Embora as atividades de manutenção sejam críticas e duradouras ao longo do ciclo de vida de um sistema, ainda não há estudos conclusivos e largamente utilizados para se estimar o esforço despendido com essas atividades.

A existência de técnicas que possam estimar com simplicidade e precisão o tamanho, esforço, prazo e custo dos projetos relacionados à manutenção de *software*, surge como de vital importância no cenário das empresas que desenvolvem, consomem ou constroem seus próprios aplicativos. Entretanto, a maioria das técnicas existentes para estimativas de projetos é mais aplicável a projetos de desenvolvimento de *software*, havendo ainda lacunas a serem preenchidas no que se refere às estimativas de manutenção de *software*.

Segundo Ramil e Lehman (2000), a técnica de Análise de Pontos por Função (*Function Point Analysis*) é uma das mais conhecidas e utilizadas para se estimar o

tamanho de um projeto de software, sendo um importante insumo para as estimativas de custos. No entanto, a FPA [FPCPM, 1999] calcula o tamanho da manutenção de um *software* como sendo o mesmo tamanho para desenvolvê-lo. Isto eleva o cálculo da estimativa de tamanho da manutenção, tornando-a irreal.

A seguir, será apresentada a técnica TUCP-M (Pontos de Caso de Uso Técnicos de Manutenção) para estimativas de manutenção de *software*.

### **3. Pontos de Caso de Uso Técnicos de Manutenção (TUCP-M)**

A proposta de estimativas de manutenção de *software* a partir de casos de uso está baseada na TUCP (Pontos de Caso de Uso Técnicos) [Monteiro *et al.* 2005], que originalmente foi concebida para tratar estimativas de caso de uso para projetos de desenvolvimento de *software*. A esta proposta, chamaremos de TUCP para manutenção de *software* ou TUCP-M. A utilização da TUCP-M objetiva um cálculo mais acurado das estimativas de projetos de manutenção de software a partir de casos de uso. Proporciona também uma visão mais detalhada de estimativas para as principais etapas do ciclo de vida do *software*, possibilitando um acompanhamento mais efetivo do projeto.

O grau de precisão da TUCP depende diretamente de a organização possuir um modelo que seja utilizado para a *Especificação de casos de uso*, como também é dependente do entendimento claro do que seja o conceito de *transação*. Estas questões influenciam diretamente no cálculo do tamanho do projeto.

No contexto deste trabalho, uma transação é cada passo dos fluxos de eventos (básico e ou alternativo) de uma especificação de caso de uso, em que ocorra um evento entre um ator e o sistema e que este deva ser executado por completo, ou ainda a realização de algum processamento complexo.

Quando uma manutenção de software é solicitada, o primeiro entendimento que se deve ter é quais requisitos novos estão sendo solicitados, quais deverão sofrer mudanças e quais deverão ser excluídos. Esses requisitos devem compor um ou mais casos de uso; sejam casos de uso novos ou já existentes.

A TUCP-M propõe, em linhas gerais, observar os seguintes itens:

- A elaboração de um documento de Solicitação de Mudanças, em que devem ser identificadas e detalhadas as mudanças solicitadas, isto é, os casos de usos afetados.
- Para cada caso de uso afetado, devem ser identificados os passos do fluxo de eventos que foram mantidos no artefato de *Especificação de Caso de Uso*. Esses passos podem ser do tipo: *incluído*, *alterado* ou *excluído*. Apenas esses passos dos fluxos de eventos dos casos de uso devem ter suas transações contadas.
- A partir do total das transações contadas para a manutenção dos casos de uso, efetua-se o cálculo da estimativa de tamanho da manutenção e, em seguida, do esforço, prazo, e custo.

#### **3.1 Estimativa de Tamanho de Manutenção**

A estimativa de tamanho da TUCP-M envolve as seguintes etapas:

- i. Contagem dos atores (UAW);
- ii. Contagem dos casos de uso (TUUCW);
- iii. Cálculo dos pontos de casos de uso não ajustados (TUUCP);
- iv. Cálculo dos fatores de complexidade técnica (TCF);
- v. Cálculo da *estimativa de tamanho* de manutenção (TUCP).

### Contagem dos Atores

Os atores devem ser classificados quanto à sua complexidade, atribuindo-se a cada um deles um peso correspondente – UAW (*Unadjusted Actor Weight*) – segundo a Tabela 1.

**Tabela 1. Classificação dos atores [Karner 1993]**

Tipo	Descrição	Peso
Simple	Aplicação com APIs definidas	1
Médio	Aplicação com interface baseada em protocolo ou interação de usuário baseado em linhas de comandos	2
Complexo	Interação de usuário através de interface gráfica ou página Web	3

Devem ser considerados para efeito de contagem apenas os atores afetados na manutenção, isto é, aqueles que efetivamente fazem parte dos casos de uso que serão mantidos.

### Contagem dos Casos de Uso

Os casos de uso devem ser classificados quanto à sua complexidade, atribuindo-se a cada um deles um peso correspondente – TUUCW (*Technical Unadjusted Use Case Points Weight*) – segundo a Tabela 2.

**Tabela 2. Contagem dos casos de uso**

Tipo	Descrição	Peso
simple	até 3 transações, incluindo os passos alternativos	5
médio	de 4 a 7 transações, incluindo os passos alternativos	10
complexo	de 8 a $t$ transações, incluindo os passos alternativos	15
$n$ -complexo	acima de $t$ transações	$P_x$

Os casos de uso com até  $t$  transações serão calculados de forma semelhante à UCP [Karner 1993]. Acima de  $t$  transações, o tipo de caso de uso será chamado de  $n$ -complexo, com o cálculo de seu peso ( $P_x$ ) apresentado nas equações abaixo [Monteiro *et al.* 2005].

$$TUUCW = 15n + p$$

$$n = T / t$$

Nas equações acima,  $T$  = número de transações do caso de uso, e  $p$  = o peso obtido, quando o resto ( $r$ ) da divisão de  $T / t$  é aplicado ao peso original (*simple*, *médio*, e *complexo*) (Tabela 2). Assim sendo, se  $r = 0$ ,  $p = 0$ ; se  $r \in [1, 3]$ ,  $p = 5$ ; se  $r \in [4, 7]$ ,  $p = 10$ ; e se  $r \in [8, (t - 1)]$ ,  $p = 15$ . A definição do valor de  $t$  em geral depende das características de cada organização, ou até mesmo das características de um dado tipo de projeto. Neste trabalho, foi utilizado  $t = 11$ , valor sugerido em Monteiro *et al.* (2005).

**SBES 2007**  
**XXI Simpósio Brasileiro de Engenharia de Software**

Para se calcular o tamanho de projetos de manutenção deve-se levar em consideração o total das transações calculadas, a partir dos *passos* dos fluxos de eventos dos casos de uso, que foram afetados pela manutenção:

- *Passos incluídos*: são os novos passos inseridos, que possam ser contados como transação.
- *Passos alterados*: são os passos que sofreram alguma modificação em sua estrutura, isto é, foram afetados pela manutenção, e que possam ser contados como transação.
- *Passos excluídos*: são os passos que tiveram sua descrição retirada do fluxo de evento, e que eram contados como transação.

Finalizada a apuração da quantidade de transações requeridas para inclusão ( $t_i$ ), alteração ( $t_a$ ), ou exclusão ( $t_e$ ) de passos do fluxo de eventos de cada caso de uso, que foram afetados pela manutenção, pode-se obter o total de transações de manutenção ( $t_m$ ) para cada caso de uso. Assim sendo, tem-se a equação abaixo. A partir do valor de  $t_m$  de cada caso de uso é calculada sua complexidade a partir da Tabela 2.

$$t_m = \sum t_i + \sum t_a + \sum t_e$$

**Cálculo dos Pontos de Casos de Uso não Ajustados**

O cálculo dos pontos de caso de uso técnicos não ajustados, TUUCP (*Technical Unadjusted Use Case Points*), é efetuado pela equação abaixo [Monteiro *et al.* 2005].

$$TUUCP = \sum UAW + \sum TUUCW$$

**Cálculo dos Fatores de Complexidade Técnica**

Os Fatores de complexidade técnica (TCF) são fatores de ajuste para o cálculo da estimativa do tamanho do projeto, e relacionam-se com as funcionalidades em geral do projeto. São apresentados na Tabela 3 com seus respectivos pesos.

**Tabela 3 - Fatores de complexidade técnica [Ribu 2001]**

Fator	Descrição	Peso	Nota
T1	Sistemas Distribuídos	2.0	
T2	Desempenho da Aplicação	1.0	
T3	Eficiência do Usuário-Final ( <i>on-line</i> )	1.0	
T4	Processamento Interno Complexo.	1.0	
T5	Reusabilidade do código em outras aplicações	1.0	
T6	Facilidade de Instalação	0.5	
T7	Usabilidade (facilidade operacional)	0.5	
T8	Portabilidade	2.0	
T9	Facilidade de Manutenção	1.0	
T10	Concorrência	1.0	
T11	Características especiais de segurança	1.0	
T12	Acesso direto para terceiros	1.0	
T13	Facilidades especiais de treinamento	1.0	

Cada um desses fatores deve ser avaliado de acordo os termos lingüísticos apresentados na Tabela 4.

**Tabela 4 – Termos lingüísticos**

Nota	Termo lingüístico	
0	Não	Sem influência (ou relevância)
1	Pouco (a)	Fraca influência (ou relevância)
2	Moderado (a)	Moderada influência (ou relevância)
3	Médio (a)	Média influência (ou relevância)
4	Alto (a)	Alta influência (ou relevância)
5	Total	Total influência (ou relevância)

O TCF é obtido através da equação abaixo [Karner 1993], onde TFactor corresponde ao somatório dos produtos entre o *peso* e a *nota* atribuída de cada um dos fatores de complexidade técnica.

$$TCF = 0,6 + (0,01 * TFactor)$$

### **Cálculo da Estimativa de Tamanho de Manutenção**

Segundo a TUCP [Monteiro *et al.* 2005], os Fatores técnicos ambientais (EF) devem ser desatrelados do cálculo do tamanho, uma vez que o tamanho é uma grandeza física e não deve ter seu valor alterado em função de fatores ambientais.

O cálculo da estimativa de tamanho de manutenção, isto é a TUCP de manutenção, que correspondente aos pontos de caso de uso técnicos ajustados, é dado pela equação abaixo.

$$TUCP = TUUCP * TCF$$

Em um projeto de manutenção, considerando-se suas *n* etapas de ciclo de vida (definidas no processo de software), o tamanho de um projeto de manutenção (*TUCP*) corresponde ao total do tamanho da manutenção realizada em cada uma de suas etapas (*E*).

$$TUCP = \sum_{i=1}^n TUCP_{Ei}$$

Esse tamanho também pode ser calculado a partir do total do tamanho da manutenção de cada um de seus *m* casos de uso (*UC*). Neste contexto, para a manutenção de um caso de uso, há artefatos diretamente relacionados a ele, como por exemplo, a *Especificação de casos de uso*. Há também artefatos que se relacionam com o conjunto de casos de uso do projeto, como por exemplo, o *Documento de arquitetura*.

$$TUCP = \sum_{j=1}^m TUCP_{UCj}$$

Estimar o tamanho da manutenção por etapa do ciclo de vida proporciona a obtenção de uma granularidade maior da estimativa, tornando mais fácil o acompanhamento do projeto e permitindo ações corretivas antes do final de cada etapa do ciclo de vida desse projeto.

A partir dos dados de projetos já concluídos da organização, podem-se levantar percentuais de realização ( $\mu_E$ ) de cada etapa do ciclo de vida (*E*), que em geral seguem uma média histórica própria da organização, dependendo das características de cada projeto e de seu domínio de aplicação.

$$TUCP_E = \mu_E * TUCP$$

### 3.2 Estimativa de Esforço de Manutenção

A estimativa de esforço de manutenção envolve as seguintes etapas:

- i. Cálculo dos fatores de ambiente (*EF*);
- ii. Cálculo da produtividade (*Prod*);
- iii. Cálculo da *estimativa de esforço* de manutenção.

#### Cálculo dos Fatores Ambientais

Os Fatores ambientais (*EF*) são fatores de ajuste para o cálculo da estimativa de esforço, e relacionam-se com os requisitos não-funcionais do projeto. São apresentados na Tabela 5 com seus respectivos pesos.

**Tabela 5 - Fatores ambientais [Ribu 2001]**

Fator	Descrição	Peso	Nota
F1	Familiaridade com o processo de desenvolvimento	1,5	
F2	Experiência com a aplicação em desenvolvimento	0,5	
F3	Experiência com orientação a objeto	1	
F4	Capacidade de análise	0,5	
F5	Motivação	1	
F6	Requisitos estáveis	2	
F7	Colaboradores de meio período	-1	
F8	Dificuldade na linguagem de programação	-1	

Cada um desses fatores deve ser também avaliado de acordo com os termos lingüísticos apresentados na Tabela 4, que variam de 0 a 5. O *EF* é obtido através da equação abaixo [Karner 1993], onde *EFactor* corresponde ao somatório dos produtos entre o *peso* e a *nota* atribuída de cada um dos fatores ambientais.

$$EF = 1,4 + (-0,03 * EFactor)$$

#### Cálculo da Produtividade

O fator de produtividade (*Prod*) está diretamente relacionado com a *estabilidade dos requisitos* e a *experiência da equipe do projeto*. Karner (1993) propôs uma produtividade de 20 h.h (homens/hora) por ponto de caso de uso, para projetos com requisitos estáveis e com equipe experiente, e 28 h.h por ponto de caso de uso, para projetos com requisitos não estáveis e com equipe não experiente.

Anda *et al.* (2001) mostraram que este esforço pode variar entre 15 h.h por ponto de caso de uso para projetos com requisitos estáveis e com equipe experiente, e 30 h.h por ponto de caso de uso para projetos com requisitos não estáveis e com equipe não experiente. Neste trabalho, serão utilizados esses valores de referência para a produtividade.

No entanto, utilizar, por exemplo, o fator de produtividade entre 15 e 30, pode significar até duplicar o esforço do projeto. Como se obter (ou calibrar), então, o fator de produtividade mais adequado para o projeto, se os membros da equipe possuem produtividades diferentes? Como combinar isto com a estabilidade dos requisitos do projeto?

**SBES 2007**  
*XXI Simpósio Brasileiro de Engenharia de Software*

O fator de produtividade pode ser calculado a partir de resultados de projetos já concluídos e armazenados em uma base histórica da organização. No entanto, o valor obtido embute as variáveis de experiência da equipe e estabilidade dos requisitos. Como então utilizar esse valor histórico, se a equipe de um projeto de manutenção nem sempre é a mesma que desenvolveu esse projeto? Além disso, cada projeto de manutenção de um mesmo produto de software não possui o mesmo grau de estabilidade de requisitos.

Neste contexto, sugere-se que o fator de produtividade seja calculado a partir dos seguintes fatores ambientais (EF):

- *Para a Experiência da equipe de projeto:*
  - **F1:** Familiaridade com o processo de desenvolvimento (peso = 1,5);
  - **F2:** Experiência com a aplicação em desenvolvimento (peso = 0,5);
  - **F3:** Experiência com orientação a objeto (peso = 1).
- *Para a Estabilidade dos requisitos:*
  - **F6:** Requisitos estáveis (peso = 2).

Assim sendo, considerando-se os fatores ambientais acima, pode-se obter a equação abaixo:

$$x_1 = (1,5 F1 + 0,5F2 + F3) \rightarrow \text{variável experiência da equipe}$$

$$x_2 = 2 F6 \rightarrow \text{variável estabilidade dos requisitos}$$

$$x = x_1 + x_2$$

Para  $F1 = F2 = F3 = F6 = 0$ , tem-se a produtividade mais baixa ( $b$ ), isto é, equipe sem experiência e requisitos não estáveis. Neste caso,  $x = 0$  e  $Prod = b$ .

Para  $F1 = F2 = F3 = F6 = 5$ , tem-se a produtividade mais alta ( $a$ ), isto é, equipe experiente e requisitos estáveis. Neste caso,  $x = 25$  e  $Prod = a$ .

A partir dos dados acima, pode-se calcular a produtividade ( $Prod$ ), através da equação linear abaixo:

$$Prod = b - [(b - a) / 25] x$$

Como neste trabalho, a produtividade varia de 15 a 30, então  $a = 15$  e  $b = 30$ , tem-se a equação abaixo:

$$Prod = 30 - 0,6 x$$

Considerando-se que a *experiência da equipe* ( $x_1$ ) de projeto está diretamente relacionada ao papel desempenhado em uma *etapa do ciclo de vida* ( $E$ ), pode-se então calcular a produtividade a partir desses papéis. Assim sendo, ter-se-ia a *variável experiência da equipe alocada àquela etapa de ciclo de vida* (e.g.: experiência dos analistas de sistemas, experiência dos arquitetos, etc.).

A produtividade do projeto ( $Prod$ ) pode ser dada pela *média ponderada* da produtividade das etapas de seu ciclo de vida ( $Prod_E$ ), considerando-se o tamanho de cada etapa ( $TUCP_E$ ):

$$Prod = \frac{\sum_{i=1}^n (Prod_{Ei} * TUCP_{Ei})}{TUCP}$$

Considerando-se também que a *variável estabilidade de requisitos* ( $x_2$ ) pode variar por caso de uso, que agrupa o comportamento de um subconjunto de funcionalidades (requisitos), pode-se então calcular a produtividade a partir da *estabilidade dos requisitos de cada caso de uso*.

A produtividade do projeto (*Prod*) pode ser dada pela *média ponderada* da produtividade dos casos de uso ( $Prod_{UC}$ ), considerando-se o tamanho de cada caso de uso ( $TUCP_{UC}$ ):

$$Pr od = \sum_{i=1}^n (Pr od_{UCj} * TUCP_{UCj}) / TUCP_{UC}$$

### **Cálculo de Esforço do Projeto**

A estimativa de esforço para o projeto pode ser obtida a partir da equação abaixo [Monteiro *et al.* 2005].

$$Esforço = TUCP * EF * Pr od$$

A estimativa do esforço de manutenção (*Esforço*) corresponde ao esforço total realizado em suas  $n$  etapas do ciclo de vida do projeto ( $E$ ), conforme as equações abaixo.

$$Esforço = \sum_{i=1}^n Esforço_{Ei}$$

$$Esforço_E = TUCP_E * EF * Pr od_E$$

O esforço de um projeto de manutenção pode ser dado também pelo esforço para a manutenção de seus casos de uso ( $UC$ ).

$$Esforço = \sum_{j=1}^m Esforço_{UCj}$$

$$Esforço_{UC} = TUCP_{UC} * EF * Pr od_{UC}$$

Em um processo de manutenção, etapas do ciclo de vida ( $E$ ) do projeto podem ser parcialmente executadas (manutenções corretivas, por exemplo). Desta forma, o esforço de manutenção de cada etapa do ciclo de vida do projeto, para cada caso de uso, pode ser acrescido de um fator de manutenção ( $f_{UC\_Ei}$ ), pertencente ao intervalo [0, 1]. Neste caso, quando  $f_{UC\_E} = 0$ , o esforço de manutenção da etapa considerada para o caso de uso em questão é nulo e, quando  $f_{UC\_E} = 1$ , o esforço de manutenção é de 100%. O novo esforço de manutenção do caso de uso ( $Esforço'_{UC}$ ) é dado pelas equações abaixo.

$$Esforço'_{UCj} = \sum_{i=1}^n (f_{UCj\_Ei} * Esforço_{UCj\_Ei})$$

$$Esforço_{UCj\_Ei} = \mu_{Ei} * Esforço_{UCj}$$

$$Esforço'_{UCj\_Ei} = \mu_{Ei} * Esforço'_{UCj}$$

O novo esforço de manutenção por etapa do ciclo de vida ( $Esforço'_{Ei}$ ) é dado por:

$$Esforço'_{Ei} = \sum_{j=1}^m (f_{UCj\_Ei} * Esforço_{UCj\_Ei})$$

O novo esforço de um projeto de manutenção de software (*Esforço'*), que utiliza casos de uso, poderá ser expresso pela equação abaixo.

$$Esforço' = \sum_{i=1}^m Esforço'_{Ei}$$

A partir do esforço estimado e esforço real do projeto, pode ser calculado seu erro relativo simétrico (*ERS*), em que:

$$ERS = (Real - Estimado) / Real, \text{ se } Real \leq Estimado$$

$$ERS = (Real - Estimado) / Estimado, \text{ se } Real \geq Estimado$$

### 3.3 Estimativas de Prazo e Custo

As estimativas de prazo e de custo podem ser obtidas a partir do esforço estimado na seção anterior, disponibilidade de recursos, restrições do projeto, entre outras questões. O uso de técnicas do tipo PERT (*Programme Evaluation Review Technique*) e CPM (*Critical Path Method*) pode auxiliar na determinação mais precisa do prazo do projeto de manutenção, levando em consideração caminhos críticos, alocação de recursos, seqüenciamento de atividades, e dependência de tarefas.

Além do prazo, a estimativa de custo do projeto pode ser também determinada utilizando-se a equação abaixo [Monteiro *et al.* 2005], em que o “Valor” corresponde ao desembolso monetário realizado para uma unidade de tamanho. Por exemplo, 1 TUCP = R\$ 80,00.

$$Custo_{(unidade\_monetária)} = TUCP * Valor_{(TUCP)}$$

Em geral, uma etapa do ciclo de vida é realizada mais fortemente por um determinado papel. Por exemplo, na etapa de “Requisitos e Análise”, o analista de sistemas desempenha o papel mais importante na equipe do projeto. Assim sendo, tendo-se o valor médio por hora pago aos analistas de sistemas, por exemplo, pode-se calcular o custo da etapa “Requisitos e Análise” a partir do esforço estimado para essa etapa.

$$Custo_{Ei(unidade\_monetária)} = Esforço_{Ei} * Valor_{(papel)}$$

A partir do cronograma do projeto, em geral elaborado a partir da WBS (*Work Breakdown Structure*), onde estão distribuídos todos os esforços por papéis, pode-se estimar de forma mais acurada o custo do projeto, a partir do valor médio por hora pago a cada profissional envolvido.

$$Custo_{(unidade\_monetária)} = \sum_{i=1}^n Custo_{Ei}$$

A seguir, serão apresentadas algumas aplicações das estimativas de manutenção de software propostas.

## 4 Aplicação das Estimativas de Manutenção de Software

A técnica TUCP-M está sendo aplicada em um conjunto de quatorze projetos de manutenção em uma empresa pública federal, que desenvolve e mantém software para o cliente interno a mais de trinta anos. Desse conjunto, seis já foram concluídos ou parcialmente concluídos. Atualmente, essa empresa possui em torno de trezentos

produtos de software (sistemas), que suportam seus negócios, envolvendo várias plataformas de desenvolvimento.

Desde 2005, essa organização possui um processo de desenvolvimento e manutenção de software baseado no RUP (2003). Atualmente, conta com uma equipe de aproximadamente quatrocentos profissionais de informática, sendo em torno de 50% terceirizados, e cerca de 80% do esforço desses profissionais é em manutenção de software.

Neste trabalho, foi desenvolvida uma ferramenta em planilha eletrônica, que automatizou a TUCP e a TUCP-M. Essa ferramenta está sendo utilizada pelos coordenadores de projeto desta organização, facilitando assim os cálculos das estimativas de projeto de software, e permitindo a coleta dos valores efetivamente realizados nesses projetos.

A seguir, serão apresentados de forma mais detalhada três projetos de manutenção de software, que foram estimados com a utilização da TUCP-M. Para esses projetos foram consideradas as seguintes etapas (*E*) do ciclo de vida de desenvolvimento: (i) Requisitos e Análise (*R&A*); (ii) *Design* (*Des*); (iii) Codificação (*Cod*); e (iv) Teste (*Tst*). Nesses projetos, a produtividade foi calculada, considerando-se cada uma dessas etapas ( $Prod_E$ ), isto é, a partir dos papéis nelas desempenhados. Para a organização, em que foi realizada esta aplicação, foram considerados os seguintes percentuais de esforço por etapa de ciclo de vida ( $\mu_E$ ): (i)  $\mu_{R\&A} = 0,27$ ; (ii)  $\mu_{Des} = 0,18$ ; (iii)  $\mu_{Cod} = 0,40$ ; e (iv)  $\mu_{Tst} = 0,15$ .

#### **4.1 Projeto A**

O *Projeto A* de manutenção foi realizado em um sistema de médio porte, desenvolvido na plataforma Cliente/Servidor, duas camadas, utilizando a linguagem *Powerbuilder*, que trata de aplicações financeiras referentes a fundos de investimentos. Esse projeto de manutenção possui um analista de sistemas, com bastante experiência e conhecimento no sistema, e um programador júnior, sem conhecimento prévio dos padrões de codificação adotados pela empresa e sem conhecimento no sistema.

Fazem parte do *Projeto A* treze casos de uso, que foram estimados segundo a TUCP-M. Entretanto, somente quatro desses casos de uso foram finalizados até o momento: UC\_01, UC\_04, UC\_05 e UC\_13.

Após a apuração da quantidade de transações requeridas para inclusão ( $t_i$ ), alteração ( $t_a$ ), ou exclusão ( $t_e$ ) de passos do fluxo de eventos, que foram afetados pela manutenção, obteve-se o total de transações de manutenção ( $t_m$ ) para cada um deles. Assim sendo, obteve-se: UC\_01 = 2  $t_m$ , UC\_04 = 3  $t_m$ , UC\_05 = 3  $t_m$ , e UC\_13 = 1  $t_m$ .

Nesse projeto, apenas 4 atores foram afetados na manutenção, com o peso total não ajustado dos atores (*UAW*) de 7. O cálculo final dos fatores de complexidade técnica foi de  $TCF = 0,96$ . O cálculo final dos fatores ambientais foi de  $EF = 0,96$ . A produtividade calculada para as quatro etapas de ciclo de vida foram: (i)  $Prod_{R\&A} = 17$ ; (ii)  $Prod_{Des} = 17$ ; (iii)  $Prod_{Cod} = 23$ ; e (iv)  $Prod_{Tst} = 19$ .

As Tabelas 6, 7, 8, e 9 apresentam o esforço de manutenção estimado e realizado para os casos de uso UC\_01, UC\_04, UC\_05 e UC\_13, respectivamente.

**SBES 2007**  
*XXI Simpósio Brasileiro de Engenharia de Software*

**Tabela 6 – Esforço de Manutenção do UC\_01 – Projeto A**

Caso de Uso	TUCP	Esforço Estimado Inicial	Fator ( $f_{UC,E}$ )	Esforço' Estimado Final	Esforço Realizado	ERS (erro)
UC_01	6,48	82,96 h.h.	25%	20,40 h.h.	35,00 h.h.	71,57%
R&A	1,75	19,33 h.h.	30%	5,79 h.h.	5,00 h.h.	-15,80%
Des	1,17	12,88 h.h.	30%	3,86 h.h.	3,00 h.h.	-28,67%
Cod	2,59	38,75 h.h.	20%	7,75 h.h.	25,00 h.h.	222,58%
Tst	0,97	12,00 h.h.	25%	3,00 h.h.	2,00 h.h.	-50,00%

**Tabela 7 – Esforço de Manutenção do UC\_04 – Projeto A**

Caso de Uso	TUCP	Esforço Estimado Inicial	Fator ( $f_{UC,E}$ )	Esforço' Estimado Final	Esforço Realizado	ERS (erro)
UC_04	6,48	82,96 h.h.	24%	20,13 h.h.	34,60 h.h.	71,88%
R&A	1,75	19,33 h.h.	35%	6,76 h.h.	6,00 h.h.	-12,67%
Des	1,17	12,88 h.h.	25%	3,22 h.h.	2,30 h.h.	-40,00%
Cod	2,59	38,75 h.h.	20%	7,75 h.h.	25,00 h.h.	222,58%
Tst	0,97	12,00 h.h.	20%	2,40 h.h.	1,30 h.h.	-84,62%

**Tabela 8 – Esforço de Manutenção do UC\_05 – Projeto A**

Caso de Uso	TUCP	Esforço Estimado Inicial	Fator ( $f_{UC,E}$ )	Esforço' Estimado Final	Esforço Realizado	ERS (erro)
UC_05	6,48	82,96 h.h.	25%	20,40 h.h.	35,00 h.h.	-68,27%
R&A	1,75	19,33 h.h.	30%	5,79 h.h.	4,00 h.h.	-44,75%
Des	1,17	12,88 h.h.	30%	2,57 h.h.	2,00 h.h.	-28,50%
Cod	2,59	38,75 h.h.	20%	7,75 h.h.	3,00 h.h.	158,33%
Tst	0,97	12,00 h.h.	20%	2,40 h.h.	2,00 h.h.	-20,00%

**Tabela 9 – Esforço de Manutenção do UC\_13 – Projeto A**

Caso de Uso	TUCP	Esforço Estimado Inicial	Fator ( $f_{UC,E}$ )	Esforço' Estimado Final	Esforço Realizado	ERS (erro)
UC_13	6,48	82,96 h.h.	12%	10,21 h.h.	7,50 h.h.	-36,13%
R&A	1,75	19,33 h.h.	20%	3,86 h.h.	3,00 h.h.	-28,67%
Des	1,17	12,88 h.h.	10%	1,28 h.h.	1,00 h.h.	-28,00%
Cod	2,59	38,75 h.h.	10%	3,87 h.h.	3,00 h.h.	-29,00%
Tst	0,97	12,00 h.h.	10%	1,20 h.h.	0,50 h.h.	-140,00%

A Tabela 10 apresenta o esforço total de manutenção realizado para os quatro casos de uso do *Projeto A*.

**Tabela 10 – Esforço de Manutenção do Projeto A**

Projeto A	TUCP	Esforço Estimado Inicial	Fator ( $f_{UC,E}$ )	Esforço' Estimado Final	Esforço Realizado	ERS (erro)
Etapas	25,92	331,84 h.h.	21%	69,25 h.h.	88,10 h.h.	27,22%
R&A	7,00	77,32 h.h.	29%	22,20 h.h.	18,00 h.h.	-23,33%
Des	4,67	51,52 h.h.	21%	10,93 h.h.	8,30 h.h.	-31,69%
Cod	10,37	155,00 h.h.	18%	27,12 h.h.	56,00 h.h.	106,49%
Tst	3,89	48,00 h.h.	19%	9,00 h.h.	5,80 h.h.	-55,17%

A partir das informações coletadas, observou-se um erro expressivo na etapa de codificação de 106,49% (subestimado). Isto se deu porque esses casos de uso foram codificados

**SBES 2007**  
**XXI Simpósio Brasileiro de Engenharia de Software**

por desenvolvedores juniores e com pouca experiência no sistema. Para esse projeto, foi contabilizado indevidamente para esse desenvolvedor o tempo de aprendizado em padrões, metodologia e bibliotecas da organização.

Já na etapa de teste, houve um erro de -55,17 (superestimado). Os casos de uso foram testados pelo próprio analista, com larga experiência em teste nesse sistema e na organização. Isto justifica o menor esforço realizado nesta etapa, uma vez que o analista que definiu o caso de uso foi o mesmo que realizou os testes. Para os quatro casos de uso considerados, o esforço real total foi de 88,10 h.h, indicando um erro de 27,22% (subestimado).

#### 4.2 Projeto B

O *Projeto B* de manutenção foi realizado em um sistema de médio porte, desenvolvido na plataforma Cliente/Servidor, duas camadas, utilizando linguagem *Visual Basic*, que operacionaliza recebimento de propostas de crédito. Esse projeto possui um analista de sistemas com bastante experiência e médio conhecimento no sistema e um programador.

Fazem parte do *Projeto B* doze casos de uso, que foram estimados segundo a TUCP-M. Entretanto, somente dois desses casos de uso foram finalizados até o momento: UC\_50, e UC\_51. Obteve-se a seguinte quantidade de transações de manutenção: UC\_50 = 4  $t_m$ , UC\_51 = 3  $t_m$ .

Nesse projeto, apenas 3 atores foram afetados na manutenção, com o peso total não ajustado dos atores (*UAW*) de 3 (todos os atores são de complexidade simples). O cálculo final dos fatores de complexidade técnica foi de  $TCF = 0,80$ . O cálculo final dos fatores ambientais foi de  $EF = 0,905$ . A produtividade calculada para as quatro etapas de ciclo de vida foram: (i)  $Prod_{R\&A} = 25$ ; (ii)  $Prod_{Des} = 25$ ; (iii)  $Prod_{Cod} = 25$ ; e (iv)  $Prod_{Tst} = 20$ . A Tabela 11 e a Tabela 12 apresentam o esforço de manutenção estimado e realizado para os casos de uso UC\_50 e UC\_51, respectivamente.

**Tabela 11 – Esforço de Manutenção do UC\_50 – Projeto B**

Caso de Uso	TUCP	Esforço Estimado Inicial	Fator ( $f_{UC\_E}$ )	Esforço' Estimado Final	Esforço Realizado	ERS (erro)
UC_50	9,60	210,67 h.h.	69%	117,06 h.h.	102,00 h.h.	-14,76%
R&A	2,59	58,64 h.h.	50%	29,32 h.h.	20,00 h.h.	-46,60%
Des	1,73	39,09 h.h.	100%	39,09 h.h.	36,00 h.h.	-8,58%
Cod	3,84	86,88 h.h.	50%	43,44 h.h.	40,00 h.h.	-8,60%
Tst	1,44	26,06 h.h.	20%	5,21 h.h.	6,00 h.h.	15,16%

**Tabela 12 – Esforço de Manutenção do UC\_51 – Projeto B**

Caso de Uso	TUCP	Esforço Estimado Inicial	Fator ( $f_{UC\_E}$ )	Esforço' Estimado Final	Esforço Realizado	ERS (erro)
UC_51	4,80	105,33 h.h.	27%	28,12 h.h.	32,00 h.h.	13,80%
R&A	1,30	29,32 h.h.	50%	14,66 h.h.	21,00 h.h.	43,25%
Des	0,86	19,54 h.h.	0%	0,00 h.h.	0,00 h.h.	0,00%
Cod	1,92	43,44 h.h.	25%	10,86 h.h.	8,00 h.h.	-35,75%
Tst	0,72	13,03 h.h.	20%	2,60 h.h.	3,00 h.h.	15,38%

A Tabela 13 apresenta o esforço total de manutenção realizado para os dois casos de uso do *Projeto B*.

**SBES 2007**  
**XXI Simpósio Brasileiro de Engenharia de Software**

**Tabela 13 – Esforço de Manutenção do Projeto B**

<i>Projeto B</i>	TUCP	<i>Esforço Estimado Inicial</i>	<i>Fator (<math>f_{UC,E}</math>)</i>	<i>Esforço' Estimado Final</i>	<i>Esforço Realizado</i>	<b>ERS (erro)</b>
<b>Etapas</b>	14,40	316,00 h.h.	46%	145,18 h.h.	134,00 h.h.	-8,34%
<i>R&amp;A</i>	3,89	87,96 h.h.	50%	43,98 h.h.	41,00 h.h.	-7,27%
<i>Des</i>	2,59	58,63 h.h.	67%	39,09 h.h.	36,30 h.h.	-8,58%
<i>Cod</i>	5,76	130,32 h.h.	42%	54,30 h.h.	48,00 h.h.	-13,13%
<i>Tst</i>	2,16	39,09 h.h.	20%	7,81 h.h.	9,00 h.h.	-15,24%

Para os dois casos de uso considerados, o esforço real total foi de 134,00 h.h, indicando um erro de - 8,34% (superestimado).

### 4.3 Projeto C

O *Projeto C* de manutenção foi realizado em um sistema de médio porte, desenvolvido na plataforma Cliente/Servidor, duas camadas, utilizando a linguagem *Powerbuilder*, que trata de aplicações com debêntures, no mercado financeiro. Esse projeto de manutenção possui um analista de sistemas, com experiência e conhecimento no sistema, e um programador com bastante experiência.

Fazem parte do *Projeto C* apenas dois casos de uso, que foram estimados segundo a TUCP-M: UC\_01, e UC\_02. Obteve-se a seguinte quantidade de transações de manutenção: UC\_01 = 1  $t_m$ , UC\_02 = 1  $t_m$ .

Nesse projeto, apenas um ator foi afetado na manutenção, com peso (*UAW*) de 7. O cálculo final dos fatores de complexidade técnica foi de  $TCF = 0,835$ . O cálculo final dos fatores ambientais foi de  $EF = 0,65$ . A produtividade calculada para as quatro etapas de ciclo de vida foram: (i)  $Prod_{R\&A} = 18,5$ ; (ii)  $Prod_{Des} = 19,5$ ; (iii)  $Prod_{Cod} = 16$ ; e (iv)  $Prod_{Tst} = 17$ . A Tabela 14 e Tabela 15 apresentam o esforço de manutenção estimado e realizado para os casos de uso UC\_01, e UC\_02, respectivamente.

**Tabela 14 – Esforço de Manutenção do UC\_01 – Projeto C**

<b>Caso de Uso</b>	TUCP	<i>Esforço Estimado Inicial</i>	<i>Fator (<math>f_{UC,E}</math>)</i>	<i>Esforço' Estimado Final</i>	<i>Esforço Realizado</i>	<b>ERS (erro)</b>
<b>UC_01</b>	5,43	61,56 h.h.	53%	32,57 h.h.	36,00 h.h.	10,53%
<i>R&amp;A</i>	1,47	17,62 h.h.	80%	14,09 h.h.	18,00 h.h.	27,75%
<i>Des</i>	0,98	12,38 h.h.	0%	0,00 h.h.	0,00 h.h.	0,00%
<i>Cod</i>	2,17	22,57 h.h.	70%	15,79 h.h.	15,00 h.h.	-5,27%
<i>Tst</i>	0,81	8,99 h.h.	30%	2,69 h.h.	3,00 h.h.	11,52%

**Tabela 15 – Esforço de Manutenção do UC\_02– Projeto C**

<b>Caso de Uso</b>	TUCP	<i>Esforço Estimado Inicial</i>	<i>Fator (<math>f_{UC,E}</math>)</i>	<i>Esforço' Estimado Final</i>	<i>Esforço Realizado</i>	<b>ERS (erro)</b>
<b>UC_02</b>	5,43	61,56 h.h.	40%	24,58 h.h.	26,00 h.h.	5,78%
<i>R&amp;A</i>	1,47	17,62 h.h.	50%	8,81 h.h.	8,00 h.h.	0,00%
<i>Des</i>	0,98	12,38 h.h.	0%	0,00 h.h.	0,00 h.h.	0,00%
<i>Cod</i>	2,17	22,57 h.h.	50%	11,28 h.h.	16,00 h.h.	41,84%
<i>Tst</i>	0,81	8,99 h.h.	50%	4,49 h.h.	2,00 h.h.	-124,50%

A etapa de teste desse projeto foi realizada pelo analista responsável e pelo usuário gestor. O tempo gasto na etapa de teste do UC\_02 foi superestimado, em virtude

de o usuário não necessitar dispensar um tempo maior para verificar se as novas informações apresentadas neste caso de uso estavam corretas.

A Tabela 16 apresenta o esforço total de manutenção realizado para os dois casos de uso do *Projeto C*.

**Tabela 16 – Esforço de Manutenção do Projeto C**

<i>Projeto C</i>	TUCP	<i>Esforço Estimado Inicial</i>	<i>Fator (<math>f_{UC,E}</math>)</i>	<i>Esforço' Estimado Final</i>	<i>Esforço Realizado</i>	<b>ERS (erro)</b>
<b>Etapas</b>	10,86	123,12 h.h.	46%	57,15 h.h.	62,00 h.h.	8,49%
<i>R&amp;A</i>	2,93	35,24 h.h.	65%	22,90 h.h.	26,00 h.h.	13,54%
<i>Des</i>	1,95	24,76 h.h.	0%	0,00 h.h.	0,00 h.h.	0,00%
<i>Cod</i>	4,34	45,14 h.h.	60%	27,07 h.h.	31,00 h.h.	14,52%
<i>Tst</i>	1,63	17,98 h.h.	40%	7,18 h.h.	5,00 h.h.	-43,60%

Para os dois casos de uso, o esforço real total foi de 62,00 h.h, indicando um erro de 8,49% (subestimado). A seguir serão apresentadas as conclusões deste trabalho.

## 5. Conclusão

As estimativas são parte essencial em um projeto de desenvolvimento de *software*, em especial em um plano de projeto. Quanto mais realistas forem essas estimativas, mais contribuirão para o sucesso do projeto.

Este trabalho estendeu a TUCP (*Technical Use Case Point*) [Monteiro *et al.* 2005], que realiza estimativas para o desenvolvimento de software, para a TUCP-M, que trata de projetos de manutenção de software.

A TUCP-M leva em consideração apenas os casos de uso afetados com a manutenção e os passos do fluxo de eventos que foram incluídos, alterados ou excluídos em virtude dessa manutenção. Desta forma, a estimativa de esforço realizada no projeto deste estudo de caso, para as aplicações realizadas têm-se mostrado mais acurada.

O resultado da aplicação da TUCP-M na organização piloto é considerado satisfatório, com um erro nas estimativas de manutenção em torno de  $\pm 20\%$  nos seis projetos concluídos. Vale ressaltar que o percentual de erro aceitável nas estimativas de projetos de software com a utilização da APF é também em torno de  $\pm 20\%$ . Apesar disso, continuam sendo coletadas e analisadas as informações dos outros oito projetos de manutenção em curso nessa organização.

Em função dos resultados obtidos, essa organização já iniciou um projeto de desenvolvimento de um sistema *Web* (em Java), para automatizar a TUCP e a TUCP-M. O objetivo é utilizar esse aplicativo em seus projetos de desenvolvimento e de manutenção de software, já estando concluída a *fase de iniciação* de seu processo de desenvolvimento.

## Referências

Anda, B. *et al.* (2001) Estimating software development effort based on use cases: experiences from industry. In: International Conference on the Unified Modeling Language (UML2001). Toronto.

**SBES 2007**  
*XXI Simpósio Brasileiro de Engenharia de Software*

- April, A., Abran, A. (1995) Industrial Research in Software Maintenance: Development of Productivity Models, Guide Summer '95 Conference and Solutions Fair, Boston.
- Bourque P., Maya M., Abran A. (1996) A Sizing Measure for Adaptive Maintenance Work Products, IFPUG Spring Conference, Atlanta.
- FPCPM. (1999) Function Point Counting Practices Manual, Version 4.1.
- Karner, G. (1993) Metrics for Objectory. Diploma thesis, University of Linköping, Sweden. No. LiTH-IDA-Ex-9344:21.
- McPhee, C. S. (1999) "Software process management: software size estimation". University of Calgary. Disponível em [http://sern.ucalgary.ca/~cmcphee/SENG621/Software\\_Size\\_Estimation.html](http://sern.ucalgary.ca/~cmcphee/SENG621/Software_Size_Estimation.html). Acesso em 2007.
- Monteiro, T. C., Pires, C. G. S., Belchior, A. D., (2005) "TUCP: Uma Extensão da Técnica UCP", IV Simpósio Brasileiro de Qualidade de Software, Porto Alegre.
- NBR ISO/IEC 12207, (1998) Tecnologia da Informação: processos de ciclo de vida de software, ABNT, Rio de Janeiro.
- Pigosky, Thomas. (1996) "Practical Software Maintenance – Best Practices for Managing Your Software Investments", WILEY Computer Publishing, Nova York.
- Ramil, J. F., Lehman, M. M. (2000) "Cost Estimation and Evolvability Monitoring for Software Evolution Process", Workshop on Empirical Studies of Software Maintenance, San Jose, CA, USA.
- Ribu, K. (2001) Estimating Object-Oriented Software Projects with Use Cases. Master of Science Thesis. University of Oslo, Norway.
- Ross, M. (2004) "Size does Matter: Continuous Size Estimating and Tracking". Quantitative Software Management. Disponível em <http://www.qsm.com>.
- RUP. (2003) Rational Software Corporation, Rational Unified Process, Version 2003.06.00.65, CD-ROM, Rational Software, Cupertino, California..
- CMMI. (2006) CMMI for Development, V1.2 model, CMU/SEI-2006-TR-008. Software Engineering Institute.
- Sousa, K. D., Oliveira, K. M., Anquetil, N. (2005) "Uso do GQM para avaliar implantação de processo de manutenção de software", IV Simpósio Brasileiro de Qualidade de Software, Porto Alegre.
- Webster K. P.B., Marçal de Oliveira K., Anquetil N. (2004) "Priorização de Riscos para Manutenção de Software" (short paper), *Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento*, Madrid, Espanha.