



Pontos de Caso de Uso Técnicos (TUCP): uma Extensão da Técnica UCP

Tatiana Cavalcanti Monteiro¹, Arnaldo Dias Belchior¹

¹Mestrado em Informática Aplicada - Universidade de Fortaleza (UNIFOR)
Av. Washington Soares, 1321, CEP 60811-341 - Fortaleza - CE – Brasil

tatiana.m@uol.com.br, belchior@unifor.br

Abstract. *The software engineering argues the implantation of activities of size estimates, effort, period and cost, as forms of improving the software projects planning and tracking. In spite of there being several techniques of estimates, the use of the same ones in software companies is not still such a common practice. The technique UCP (Use Case Points), for instance, it is adherent to object-oriented software products and based on use cases. However, they have been finding some situations where there are difficulties of obtaining resulted fully satisfactory when using the UCP. This work presents an extension of the technique UCP – that is TUCP – being looked for a calculation more refined for the projects effort. Besides, TUCP allows a more detailed vision of estimates in the main stages of the software cycle life, making possible the accomplishment of refinements of those estimates for an accompaniment more effective of the project.*

Keywords: *Estimation, UCP (Use Case Points), Use Cases.*

Resumo. *A engenharia de software recomenda a implantação de atividades de estimativas de tamanho, esforço, prazo e custo, como formas de melhorar o planejamento e o acompanhamento de projetos de software. Apesar de haver várias técnicas de estimativas, a utilização das mesmas em empresas de software ainda não é uma prática tão comum. A técnica UCP (Pontos por Caso de Uso), por exemplo, é aderente a produtos de software orientados a objetos e baseados em casos de uso. No entanto, têm-se encontrado algumas situações, onde há dificuldades de se obter resultados plenamente satisfatórios ao se utilizar a UCP. Este trabalho apresenta uma extensão da técnica UCP – a TUCP – buscando-se um cálculo mais acurado para o esforço de projetos. Além disso, a TUCP permite uma visão mais detalhada de estimativas nas principais etapas do ciclo de vida do software, possibilitando a realização de refinamentos dessas estimativas para um acompanhamento mais efetivo do projeto.*

Palavras-Chave: *Estimativas, UCP (Pontos por Caso de Uso), Caso de Uso.*

1. Introdução

As estimativas apóiam essencialmente as atividades de planejamento e acompanhamento de projetos de software. Estimativas eficientes permitem a verificação da viabilidade do projeto, a elaboração de propostas técnicas e comerciais, a confecção de planos e cronogramas detalhados, e o acompanhamento efetivo de projetos. Dentre as diversas técnicas para se estimar projetos de software, a UCP (*Use Case Points*), que surgiu nos anos noventa, tem sido



aprimorada e tem apresentado crescente utilização. Essa técnica é adequada a projetos, que descrevem seus requisitos de software através de casos de uso.

Este trabalho apresenta uma extensão da técnica UCP, a TUPC (*Technical Use Case Points*), propondo um cálculo mais acurado para o cálculo do esforço de projetos, e permitindo uma visão mais detalhada das estimativas por etapa do ciclo de vida do software, possibilitando a realização de refinamentos dessas estimativas para um acompanhamento mais efetivo do projeto. Além disso, a TUCP propõe calibrações nos fatores de produtividade nessas etapas, permitindo assim a obtenção de estimativas mais precisas.

A organização deste trabalho é feita em cinco seções. Na seção 2, descreve-se sobre os conceitos de casos de uso, que é a matéria prima para a estima proposta. Na seção 3, delinea-se sobre estimativas de projetos de software. Na seção 4, discorre-se sobre a extensão da UCP proposta, a TUCP. Na seção 5, apresenta-se um estudo de caso, utilizando-se a TUCP. Finalmente, na seção 6, são mostradas as conclusões.

2. Casos de Uso

Um caso de uso descreve o comportamento de um sistema, que produz um resultado mensurável de valor para um ator. Este expõe aquilo que os atores querem que o sistema execute para eles. Isso deve ser uma tarefa completa na perspectiva de um ator. Portanto, cada caso de uso identificado deverá estar associado à pelo menos um ator [Schneider *et al.* 2001].

Os casos de uso podem ser modelados a partir dos cenários obtidos para o sistema. Um cenário mostra uma seqüência de passos, que descreve as interações entre um ator e o sistema, isto é, uma alternativa do que pode acontecer. Cada passo deve ser uma declaração simples, evidenciando claramente quem o está executando. O passo deve apresentar a intenção do ator e não os mecanismos daquilo que o ator faz. Portanto, a interface com o caso de uso não deve ser descrita [Fowler 2005].

Várias têm sido as propostas de como se construir casos de uso efetivos, através de formulários e regras para a descrição dos mesmos [Anda *et al.* 2001; Ribu 2001; Anda 2002; OMG 2003; Cockburn 2001]. No entanto, ainda não há uma padronização para a descrição de casos de uso [Fowler 2005], especialmente por esta ser de natureza subjetiva e depender do grau de detalhamento e conhecimento das informações disponibilizadas. Este trabalho propôs um modelo de *Especificação de Casos de Uso* [Monteiro 2005a], a partir de pesquisas na literatura sobre o tema [Ribu 2001; OMG 2003; Cockburn 2001, RUP 2003; Schneider *et al.* 2001]. Isto porque a forma de especificar casos de uso é um fator preponderante no cálculo da estimativa de tamanho do software.

A seguir, serão apresentados os principais conceitos envolvidos em uma especificação de caso de uso.

2.1. Ator

Um ator representa um papel que uma pessoa, um dispositivo de hardware ou um outro sistema pode desempenhar em relação ao sistema. Os nomes dos atores devem denotar claramente o papel que estes desempenham e são representados por um *stickman* [RUP 2003].

Para se identificar os atores de um sistema, deve-se pensar primeiramente nos possíveis usuários que o utilizarão, e também como esses atores podem ser categorizados. Normalmente, é tida como uma boa prática pensar em alguns usuários (dois ou três) e verificar se os atores identificados vão cobrir as necessidades desses usuários.



A melhor forma de identificar casos de uso é considerar o que cada ator exige do sistema. Isto porque o sistema existe apenas para seus usuários e deve estar baseado nas necessidades destes. Muitas das necessidades dos atores serão reconhecidas através dos requisitos funcionais especificados no sistema. Essas necessidades podem ser modeladas através de fluxos de eventos [Kruchten 2001].

2.2. Fluxo de Eventos

As duas principais partes do fluxo de eventos são: o fluxo básico e os fluxos alternativos. Opcionalmente, podem-se ter também os fluxos de exceção. Esses fluxos podem ser capturados em seções de uma especificação de casos de uso.

O fluxo básico deve abordar o que "geralmente" ocorre quando o caso de uso é executado, enfatizando o cenário de sucesso do início ao fim (cenário *happy day*). É recomendável que ele seja relativamente pequeno e de fácil leitura, apresentando os passos necessários para alcançar o objetivo principal do caso de uso.

Os fluxos alternativos abordam o comportamento de caráter opcional ou excepcional em relação ao comportamento normal e também as variações do comportamento normal. Pode-se pensar nos fluxos alternativos como "desvios" ou variantes do fluxo básico, alguns dos quais voltarão ao fluxo básico e alguns finalizarão a execução do caso de uso [Kruchten 2001].

2.3. Tipos de Relacionamentos

Um caso de uso pode apresentar três tipos relacionamentos: inclusão, extensão, e generalização. Nesses relacionamentos, o caso de uso original, que é modificado, é chamado de *caso de uso base*. Todos os casos de uso abstratos são do tipo de inclusão, extensão, ou generalização. Portanto, não é necessário ter um ator associado a esses casos de uso.

O relacionamento de inclusão (*include*) conecta um *caso de uso base* a um *caso de uso de inclusão*. Um caso de uso de inclusão descreve um segmento de comportamento que é inserido em uma instância de caso de uso ao ser executado o caso de uso base. [Kruchten 2001].

O relacionamento de extensão (*extende*) estabelece a conexão entre um *caso de uso de extensão* e um *caso de uso base* [Schneider *et al.* 2001]. A extensão deve ser inserida fazendo referência a *pontos de extensão* no caso de uso base [RUP 2003].

Uma generalização de casos de uso é um relacionamento de um caso de uso filho com um caso de uso pai, especificando como um filho pode adotar todo o comportamento e as características descritas para o pai [OMG 2003].

3. Estimativas de Projetos de Software

Um dos principais riscos que atinge o processo de estimativas é a falta de credibilidade nas estimativas pelas equipes de desenvolvimento [Boehm 2000]. Isto ocorre quando as estimativas são irrealistas, isto é, os projetos são freqüentemente subestimados ou superestimados. 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 [SEI 2002].

Segundo Vasquez (2003), o processo de estimativa de um projeto de software envolve quatro atividades básicas: (i) estimar o tamanho do produto a ser desenvolvido; (ii) estimar o



esforço empregado na execução do projeto; (iii) estimar o prazo do projeto; e (iv) estimar o custo do projeto.

O cálculo do esforço do projeto pode ser obtido a partir da estimativa de tamanho do produto. O esforço total é obtido baseado no processo de desenvolvimento, que envolve muito mais do que a simples atividade de codificação do software – o fato é que a codificação é freqüentemente a menor parte do esforço real de todo o projeto. Elaboração de documentos, implementação de protótipos, projeto do produto a ser entregue, revisão e teste do código levam uma grande fatia de todo o esforço do projeto [Peters 1999].

Com as estimativas de tamanho e esforço calculadas para um projeto de software pode-se, então, determinar as estimativas de prazo. Isto geralmente envolve estimar quantas pessoas estarão envolvidas no projeto, que atividades serão executadas (*WBS*), e quando essas atividades iniciarão e serão finalizadas.

O custo é comumente proporcional ao esforço despendido para a construção do mesmo, onde o trabalho humano é o principal recurso a ser consumido. Conseqüentemente, o custo é com freqüência associado a homens-mês (h.m) ou homens-hora (h.h).

Atualmente, existem diversas métricas de estimativa de tamanho de projetos de software. Entretanto, não é trivial a seleção de uma métrica que seja a mais apropriada para uma organização. Podem-se citar algumas das métricas mais conhecidas e utilizadas atualmente: *Function Point Analysis* (Análise por Pontos de Função); *Bang*; *Features Points*; *Boeing's ED Function Point Analysis*, *MK II Function Point Analysis*, *COSMIC Full Function Point*, *Internet Points*, *Domino Points*, *Class-Method Points*, *Use Case Points* [Garmus and Herron 2000; Mcphee 1999; Roetzheim 2000].

A técnica *Use Case Points* (Pontos de Casos de Uso) ou UCP estima o tamanho de projeto de software orientado a objetos, criada por Kerner (1993), com base na FPA, MK II FPA e no processo “*Objectory*”, onde foi desenvolvida a técnica de diagramação para o conceito de casos de uso [Ribu 2001].

A UCP é simples, rápida e fácil de usar [Damodaran; Washington s.d]. São contados atores e casos de uso e com base nessa contagem são calculados os UCP não ajustados. Posteriormente, são determinadas as complexidades dos *fatores técnicos* e dos *fatores ambientais* para o cálculo do tamanho e do esforço do projeto.

Na seção seguinte, será apresentada a TUCP que é uma extensão da UCP, buscando um cálculo mais acurado para o esforço despendido em projetos de software.

4. TUCP

A TUCP (Pontos de Caso de Uso Técnicos) é uma extensão da UCP (Pontos de Caso de Uso), proposta por [Monteiro 2005a], objetivando um cálculo mais acurado para as estimativas a partir de casos de uso. Toda esta seção está baseada em [Monteiro 2004a, 2004b, 2004c, 2004d, 2005a, 2005b].

A TUCP congrega as seguintes questões:

- Elabora um *Guia para a especificação de casos de uso*, uma vez que casos de uso bem elaborados podem influenciar fortemente no tamanho do projeto.
- Conceitua transação no contexto de casos de uso, por ser utilizada como unidade de medida para o cálculo do tamanho dos casos de uso.
- Amplia a contagem de pontos de casos de uso *complexos*, inserindo os chamados pontos de caso de uso *n-complexos*.



- Desatrela os *fatores técnicos* de ambiente (EF) do cálculo do tamanho, uma vez que o tamanho é uma grandeza física e não deve ter seu valor alterado em função de questões ambientais.
- Considera os *fatores ambientais* EFs apenas no cálculo do esforço, juntamente com o fator de produtividade (de forma semelhante à UCP).
- Granulariza o cálculo do esforço em etapas do processo de desenvolvimento e por caso de uso.

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.

Uma transação é cada passo dos fluxos de eventos (básico e ou alternativo) de um caso de uso, onde ocorra um evento entre um ator e o sistema, e que deva ser executado por completo ou ainda a realização de algum processamento complexo nesses fluxos de eventos. A seguir, serão apresentadas as estimativas de tamanho e esforço, segundo a TUCP.

4.1 Estimativa de Tamanho

A estimativa de tamanho da TUCP 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) e (v) Cálculo dos pontos de caso de uso técnicos (TUCP).

Contagem dos Atores

O peso total dos atores do sistema é calculado pela soma dos produtos dos atores de cada tipo pelo respectivo peso, UAW (*Unadjusted Actor Weight*), de forma semelhante à UCP [Karner, 1993], segundo a Tabela 1.

Tabela 1. Classificação dos atores

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

Contagem dos Casos de Uso

Através de experiências em trinta projetos de software utilizando-se a técnica UCP, em uma organização certificada SW-CMM, nível 2, constataram-se atrasos na entrega do produto final para o cliente. Esses atrasos, em geral, estavam relacionados a projetos que continham casos de uso complexos, especialmente aqueles com um grande número de transações (maior que 12 transações).

Com base nesta observação, uma adaptação no cálculo dos pesos para os casos de uso complexos foi elaborada. Uma nova tabela de valores para a classificação do peso foi montada para refletir essa questão apresentada.

Assim sendo, o peso total dos casos de uso do sistema é calculado a partir das Tabela 2, isto é, o TUUCW (*Technical Unadjusted Use Case Points Weight*).

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

Tipo	Descrição	Peso
simples	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 a UCP [Karnar, 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 a seguir.

$$TUUCW = 15n + p, \text{ onde } n = T / t.$$

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 (*simples*, *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 $r \in [8, (t - 1)]$, $p = 15$.

Por exemplo: um caso de uso com 13 transações, e com $t = 11$, temos $T = 13$, $n = 13 / 11$, como $r \in [1, 3]$, então $p = 5$, assim:

$$TUUCW = 15 \times 1 + 5 = 20$$

A definição do valor de t poderá depender das características da organização ou até mesmo das características de um dado tipo de projeto. Recomenda-se a calibração do valor de t , após ter sido levantado um histórico de (tipos) projetos concluídos, que possuam casos de uso do tipo n -complexo. Essa calibração deverá ser realizada baseada no esforço real de implementação do conjunto de projetos considerados.

Após a definição do valor de t , o peso de um caso de uso n -complexo vai sendo incrementado em um ciclo do tipo (3; 4; k). Isto significa dizer que a cada $(ft + 3)$, $((ft + 3) + 4)$, $((ft + 3) + 4) + k$, ... transações, e assim por diante, o valor do peso é modificado. O valor de $f \in \mathbb{N}$ e que k é dado pela Eq. 8.

$$k = t - 7 \text{ (Eq. 8)}$$

Na organização SW-CMM, nível 2, onde foram realizados os experimentos durante o período de aproximadamente dois anos, envolvendo uma média de sessenta projetos, o valor de k que se mostrou mais apropriado foi $k = 4$. Experimentos variando o valor de k entre 4 a 7 foram empreendidos nesses projetos. A partir da análise dos dados entre estimativas planejadas e realizadas foi verificado (depois de várias simulações), que o valor mais apropriado era $k = 4$. Na prática, é como se tivéssemos um caso de uso n -complexo transformado em n casos de uso complexos (com até 11 transações cada um deles).

No entanto, percebeu-se que alguns dos projetos que participaram dos experimentos tinham características peculiares (e.g.: utilização de *frameworks* de desenvolvimento, forte grau de reuso). Para esses projetos, um valor de $k > 4$ indicou resultados mais próximos do esforço real despendido nesses projetos.



Cálculo dos Pontos de Casos de Uso não Ajustados

O cálculo de TUUCP (*Technical Unadjusted Use Case Points*) é efetuado pela equação abaixo.

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

Cálculo dos Fatores de Complexidade Técnica

Os 13 fatores de complexidade técnica (TCF) são calculados de forma similar à UCP [Kaner, 1993]. Na equação abaixo, TFactor é o somatório dos produtos entre o peso e a nota atribuída a cada um dos 13 fatores de complexidade técnica.

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

Estimativa de Tamanho (TUCP)

O cálculo da TUCP (*Technical Use Case Points*) ajustada é dado pela equação abaixo.

$$TUCP = TUUCP * TCF$$

Estimativa de Tamanho por Etapa do Ciclo de Vida do Projeto (TUCP)

Como o esforço de um projeto é diretamente proporcional a seu tamanho, então, o tamanho de um caso de uso por etapa do ciclo de vida pode ser baseado no percentual de esforço empregado para seu desenvolvimento. Assim sendo, e baseado em Meneses (2001), definiu-se a estimativa de tamanho do caso de uso por etapa do ciclo de vida na equação abaixo.

$$TUCP_{(UC_etapa)} = \left(\left(\frac{TUCP}{\sum TUUCW} \right) * TUUCW_{(UC)} \right) * \text{Percentual de Esforço}_{(etapa)}$$

$TUCP_{(UC_etapa)}$ é o tamanho do caso de uso por etapa do ciclo de vida, sendo definido proporcionalmente pelo seu fator de complexidade e seu percentual de distribuição de esforço para a etapa do ciclo de vida considerada. $TUUCW_{(UC)}$ é o peso do caso de uso que está sendo calculado.

O tamanho TUCP do caso de uso como um todo ao longo do projeto é mostrado abaixo.

$$TUCP_{(caso\ de\ uso)} = \left(\frac{TUCP}{\sum TUUCW} \right) * TUUCW_{(UC)}$$

O tamanho TUCP de uma etapa do projeto é dado pela equação abaixo.

$$TUCP_{(etapa)} = TUCP * \text{Percentual de Esforço}_{(etapa)}$$

4.2 Estimativa de Esforço

A estimativa do esforço da TUCP envolve as seguintes etapas: (i) cálculo dos fatores de ambiente (EF), (ii) cálculo da produtividade (PROD) e (iii) cálculo do esforço técnico (Esforço).



Cálculo dos Fatores Ambientais

Os 8 fatores ambientais (EF) são calculados de forma similar à UCP [Karner, 1993]. Na equação abaixo, EFator é o somatório dos produtos entre o peso e a nota atribuída a cada um dos 8 fatores ambientais.

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

Cálculo da Produtividade

Este cálculo pode ser calibrado de acordo com a produtividade da equipe do projeto. Karner (1993) propôs uma produtividade de 20 homens/hora (h.h) por UCP para projetos onde a equipe seja considerada estável e experiente, e 28 h.h por UCP para projetos em que os requisitos não sejam estáveis e com uma equipe não experiente. Anda *et al.* (2001) mostrou-se que este esforço pode variar de 15 h.h por UCP para projetos onde a equipe seja considerada estável e experiente, e 30 h.h por UCP para projetos em que os requisitos não sejam estáveis e com uma equipe não experiente.

Estimativa do Esforço

O cálculo do esforço do projeto é calculado de forma semelhante à UCP.

$$\text{Esforço} = \text{TUCP} * \text{EF} * \text{PROD}$$

De forma semelhante, pode-se calcular o esforço por caso de uso.

$$\text{Esforço}_{(UC)} = \text{TUCP}_{(UC)} * \text{EF} * \text{PROD}$$

Estimativa de Esforço por Etapa do Ciclo de Vida do Projeto

Na UCP [Karner 1993], o fator produtividade é único para todo o projeto. No entanto, constatou-se que o fator de produtividade pode variar dependendo da etapa do ciclo de vida do projeto. Isto porque a experiência da equipe de desenvolvimento pode variar de uma etapa para outra, já que, normalmente, as pessoas envolvidas em uma etapa não são as mesmas de uma outra etapa. Além disso, outros fatores podem influenciar na produtividade como: tipo de equipe, linguagem, experiência, utilização de *frameworks*, etc.

A calibragem para a produtividade para cada etapa do ciclo de vida deve ser obtida da base histórica organizacional. Assim, o esforço por etapa do ciclo de vida do projeto é calculado na equação abaixo.

$$\text{Esforço}_{(etapa)} = \text{TUCP}_{(etapa)} * \text{EF} * \text{PROD}_{(etapa)}$$

A seguir, será apresentada uma proposta para o acompanhamento do progresso funcional ao longo do desenvolvimento de software.

4.3 Progresso Funcional

Com o objetivo de auxiliar no acompanhamento e monitoramento do sistema, este trabalho propôs também uma visão de progresso funcional ao longo do ciclo de vida do sistema. Assim sendo, é mensurado o percentual da funcionalidade de todo o sistema, a partir da análise do progresso de cada um de seus casos de uso.

A visão funcional tem como informação essencial a métrica global, μ_{Sistema} , definida por Champeaux (1997) *appud* Meneses (2001), que indica quantitativamente o progresso funcional do sistema. Essa métrica permitirá, entre outros aspectos, visualizar o progresso da funcionalidade atual, comparar seu crescimento com incrementos e fases



anteriores, identificar possíveis dificuldades existentes no sistema, e evidenciar casos de uso mal projetados.

Para o acompanhamento e o monitoramento mais adequado do sistema, sentiu-se a necessidade de uso das seguintes visões: (i) progresso funcional do sistema; (ii) progresso funcional da etapa do ciclo de vida; (iii) progresso funcional do caso de uso.

Progresso Funcional do Sistema

Como os casos de uso frequentemente apresentam complexidade variada (TUCP) e necessitam de níveis de esforço distintos para sua realização ao longo das etapas do ciclo de vida, pode-se obter também o progresso funcional do sistema através da equação abaixo. O μ_{Etapa} indica o progresso do caso de uso de uma etapa do ciclo de vida em valor percentual.

$$\mu_{Sistema} = \frac{\sum (TUCP_{etapa} * \mu_{etapa})}{\sum TUCP_{etapa}}$$

Progresso Funcional da Etapa do Ciclo de Vida

A métrica relacionada ao progresso funcional da etapa do ciclo de vida do sistema pode ser calculada através da equação abaixo. Neste caso, μ_{UC_Etapa} é um valor percentual que indica o progresso funcional do caso de uso em uma dada etapa do ciclo de vida. Esse progresso pode ser obtido em função da elaboração dos produtos de trabalho naquela etapa, isto é, o quanto cada produto de trabalho já foi realizado.

$$\mu_{etapa} = \frac{\sum (TUCP_{UC_Etapa} * \mu_{UC_Etapa})}{\sum TUCP_{UC_Etapa}}$$

Progresso Funcional do Caso de Uso

A métrica relacionada ao progresso funcional de um caso de uso pode ser calculada através da equação abaixo. Assim, μ_{UC} é um valor percentual que indica o progresso funcional do caso de uso, considerando-se suas etapa do ciclo de vida. Seja, por exemplo: *Req* (Requisitos), *A&P* (Análise e Projeto), *Cod* (Codificação) e *Tst* (Teste).

$$\mu_{UC} = \frac{\sum_{Etapa \in \{Req, A\&P, Cod, Tst\}} (TUCP_{UC_Etapa} * \mu_{UC_Etapa})}{\sum_{Etapa \in \{Req, A\&P, Cod, Tst\}} TUCP_{UC_Etapa}}$$

A seguir, será apresentado um estudo de caso para um melhor entendimento das estimativas propostas. Os resultados expostos foram obtidos em projetos reais que utilizaram a TUCP.

5. Estudo de Caso

Este trabalho foi aplicado em uma organização de pesquisa e desenvolvimento, que era SW-CMM nível 2 e foi certificada CMMI nível 3 em fevereiro de 2006.

Neste trabalho, serão apresentados os resultados de apenas três projetos, que serão chamados de Projeto *A*, Projeto *B*, e Projeto *C*. No entanto, apenas o Projeto *A* será detalhado para fins de demonstração de cálculo da TUCP. Nesta organização, foi considerado o valor de $t = 11$, após vários refinamentos do valor de k (seção 4.1).

O Projeto *A* foi desenvolvido na plataforma J2EE com uma equipe de projeto composta por seis pessoas consideradas experientes. Esse projeto possuía vinte e quatro casos



de uso, sendo sete simples, dez intermediários, um complexo e seis n -complexos. Este sistema tinha apenas um ator complexo (Tabela 1). O valor TUUCP deste sistema será apresentado a seguir.

$$TUUCP = \sum UAW + \sum TUUCW = 3 + 310 = 313$$

O tamanho TUCP deste sistema foi calculado a partir do tamanho de cada caso de uso e com base nas características do projeto, conforme abaixo.

$$TUCP = TUUCP * TCF = 313 * 1,125 \cong 352,1$$

A partir do cálculo do fator de complexidade ambiental (EF) e da produtividade da equipe do projeto de 20 h.h (Tabela 4), obteve-se o esforço do projeto:

$$\text{Esforço} = TUCP * EF * PROD = 352,1 * 0,815 * 20 \cong 5739,6 \text{ h.h}$$

Com a TUCP é possível se ter estimativas de tamanho e esforço por etapa do ciclo de vida para cada caso de uso. Como o Projeto *A* possui vinte e quatro casos de uso, será apresentado apenas o cálculo para um único caso de uso (UC_01), detalhado por etapa do ciclo de vida. A Tabela 3 apresenta o UC_01 do sistema com a sua distribuição de tamanho e esforço por etapa do ciclo de vida (Requisitos (*Req*), Análise e Projeto (*A&P*), Codificação (*Cod*) e Teste (*Tst*)).

Tabela 3. Detalhamento do UC_1 do Projeto A por etapa do ciclo de vida

Caso de Uso	Req (TUCP)	Req (h.h)	A&P (TUCP)	A&P (h.h)	Cod (TUCP)	Cod (h.h)	Tst (TUCP)	Tst (h.h)	Tamanho (TUCP)	TUCP (h.h)
UC_01	1,1	18,5	1,4	23,1	2,6	41,7	0,6	9,3	5,7	92,6

O cálculo do tamanho do UC_01 é dado abaixo (seção 4.1).

$$TUCP_{(UC_{01})} = \left(\left(\frac{352,1}{310} \right) * 5 \right) \cong 5,7$$

Tanto a produtividade da equipe, quanto o percentual de esforço foram obtidos a partir da base histórica organizacional (Tabela 4).

Tabela 4. Produtividade e percentual de esforço por etapa do ciclo de vida

Etapa	Percentual de Esforço	Produtividade
Requisitos	20%	20h.h
Análise e Projeto	25%	20 h.h
Codificação	45%	20 h.h
Testes	10%	20 h.h

O tamanho do caso de uso UC_01 nas etapas do ciclo de vida (*Req*, *A&P*, *Cod*, e *Tst*) foi calculado abaixo (seção 4.1).

$$1. TUCP_{(UC_{01_Req})} = \left(\left(\frac{352,1}{310} \right) * 5 \right) * 0,20 \cong 1,1$$

$$2. TUCP_{(UC_{01_A \& P})} = \left(\left(\frac{352,1}{310} \right) * 5 \right) * 0,25 \cong 1,4$$

$$3. TUCP_{(UC_{01_Cod})} = \left(\left(\frac{352,1}{310} \right) * 5 \right) * 0,45 \cong 2,6$$



$$4. \text{TUCP}_{(\text{UC}_{01_Tst})} = \left(\left(\frac{352,1}{310} \right) * 5 \right) * 0,10 \cong 0,6$$

O cálculo do tamanho TUCP de cada etapa do ciclo de vida do projeto (*Req, A&P, Cod, e Tst*) foi realizado se segue (seção 4.1).

$$1. \text{TUCP}_{(\text{REQ})} = 352,1 \times 0,20 \cong 70,4$$

$$2. \text{TUCP}_{(\text{A \& P})} = 352,1 \times 0,25 \cong 88,0$$

$$3. \text{TUCP}_{(\text{COD})} = 352,1 \times 0,45 \cong 158,5$$

$$4. \text{TUCP}_{(\text{TST})} = 352,1 \times 0,10 \cong 35,2$$

A partir do tamanho TUCP de cada etapa do ciclo de vida do projeto, o esforço por etapa foi calculado abaixo (seção 4.2).

$$1. \text{Esforço}_{(\text{Req})} = 70,4 * 0,815 * 20 \cong 1147,9$$

$$2. \text{Esforço}_{(\text{A \& P})} = 88,03 * 0,815 * 20 \cong 1434,9$$

$$3. \text{Esforço}_{(\text{Cod})} = 158,5 * 0,815 * 20 \cong 2582,8$$

$$4. \text{Esforço}_{(\text{Tst})} = 35,2 * 0,815 * 20 \cong 573,9$$

Na Tabela 5, são mostrados os dados do Projeto A: (i) etapas do ciclo de vida; (ii) o tamanho em TUCP; (iii) o esforço estimado; (iv) o esforço real; (v) a produtividade estimada; (vi) a produtividade real; (vii) o percentual de esforço realizado de cada etapa do ciclo de vida do projeto; e (viii) o percentual de erro em relação ao esforço estimado. Pode ser percebido que a equipe foi muito produtiva, pois sua média de produtividade em torno 15 h.h.

Para este projeto não foi preciso o ajuste no fator de produtividade, já que a produtividade foi de 20 h.h para todas as etapas do ciclo de vida. Entretanto, se um projeto tivesse uma equipe não experiente com as atividades de codificação, por exemplo, poderia ter sido feito um ajuste no fator de produtividade do esforço ($\text{PROD}_{(\text{Etapa})}$) para 26 h.h.

Tabela 5. Dados do Projeto A calculados com base na TUCP

Etapas do Ciclo de Vida	Tamanho (TUCP)	Esforço Estimado (H.H)	Esforço Real (H.H)	Produtividade Estimada (H.H/TUCP)	Produtividade Real (H.H/TUCP)	% Esforço Real	% Erro Estimado
Requisitos	70,4	1147,9	956,2	20	13,58	18,19	-20
Análise e Projeto	88,0	1434,9	1007,8	20	11,45	19,17	-42
Codificação	158,5	2582,8	2734,5	20	17,26	52,02	6
Testes	35,2	574,0	558,0	20	15,85	10,62	-3
Total	352,1	5739,6	5256,5	20	14,93	100	-9

O cálculo para o percentual de erro estimado, considerando o esforço real e esforço estimado, para cada etapa utilizou a métrica SER (*Symmetric Relative Error*), proposta por M. Jorgensen e D. Sjobeg (Ribu, 2001):

- $\text{SER} = (\text{Real} - \text{Estimado}) / \text{Real} \Leftrightarrow \text{Real} \leq \text{Estimado}$
- $\text{SER} = (\text{Real} - \text{Estimado}) / \text{Estimado} \Leftrightarrow \text{Real} \geq \text{Estimado}$



Na Tabela 6 e Tabela 7 são apresentados os dados dos três projetos considerados, utilizando-se a técnica TUCP e a UCP, respectivamente. Esses dados foram comparados com os valores reais obtidos nos projetos. Pode ser percebido que o percentual de erro estimado em TUCP foi menor do que o percentual de erro em UCP para dois dos projetos, e no Projeto C o percentual foi o mesmo. Neste caso, o Projeto não apresentava casos de uso do tipo *n*-complexo (Tabela 8).

Tabela 6. Tamanho e esforço dos Projetos A, B e C pela TUCP

Projetos	Tamanho (TUCP)	Esforço Estimado (H.H)	Esforço Real (H.H)	Produtividade Estimada (H.H/TUCP)	Produtividade Real (H.H/TUCP)	% Erro Estimado (TUCP)
Projeto A	352,1	5739,6	5256,5	20	14,93	9
Projeto B	74,7	1352,5	1477,0	20	19,77	9
Projeto C	103,3	1833,8	1871,0	24	18,12	2

Tabela 7. Tamanho e esforço dos Projetos A, B e C pela UCP

Projetos	Tamanho (UCP)	Esforço Estimado (H.H)	Esforço Real (H.H)	Produtividade Estimada (H.H/UCP)	Produtividade Real (H.H/UCP)	% Erro Estimado (UCP)
Projeto A	222,8	4457,0	5256,5	20	23,59	18
Projeto B	62,08	1241,7	1477,0	20	23,79	19
Projeto C	76,4	1833,8	1871,0	24	24,49	2

Tabela 8. Tipos e quantidades de casos de uso nos Projetos A, B e C

Projetos	Tipo de Caso de Uso				Total
	Simples	Intermediário	Complexo	N-Complexo	
Projeto A	7	10	1	6	24
Projeto B	3	2	0	1	6
Projeto C	1	3	4	0	8

Vale salientar que a comparação da eficácia da técnica UCP em relação à TUCP dá-se apenas através do cálculo do esforço. Não se podem comparar os tamanhos da UCP e da TUCP, por terem fórmulas de cálculo distintas.

No Projeto A o progresso funcional do sistema é calculado com base na técnica TUCP conforme mostrado a seguir. Este projeto é composto de 24 casos de uso, mas serão considerados apenas os casos de uso UC_01, UC_02, e UC_03. Serão atribuídos, a título de exemplo, os valores percentuais do subconjunto {0,00; 0,25; 0,50; 0,75; 1,00} para indicar o progresso funcional (0% a 100%) de cada um desses três casos de uso por etapa do ciclo de vida (Tabela 9).

Tabela 9 - Progresso funcional de caso de uso por etapa de ciclo de vida

Caso de Uso	μ_{Req}	μ_{AdP}	μ_{Cod}	μ_{Tst}	μ_{UC}
UC_01	1,00	1,00	1,00	1,00	1,00
UC_02	1,00	1,00	0,75	0,50	0,84
UC_03	0,50	0,75	0,00	0,00	0,33

Portanto, foi calculado abaixo o progresso funcional dos casos de uso UC_01, UC_02, e UC_03 levando em consideração o percentual funcional de cada caso de uso (seção 4.3).



$$\mu_{UC_01} = \frac{1,1*1,00 + 1,4*1,00 + 2,6*1,00 + 0,6*1,00}{5,7} = 1,00$$

$$\mu_{UC_02} = \frac{1,057*1,00 + 1,391*1,00 + 2,781*0,75 + 0,334*0,50}{5,56} = 0,84$$

$$\mu_{UC_03} = \frac{1,891*0,50 + 3,672*0,75 + 3,8672*0,00 + 1,1891*0,00}{11,13} = 0,33$$

A partir dos dados acima, em relação ao que foi estimado para o progresso funcional dos casos de uso considerados, o caso de uso UC_01 já foi concluído; o caso de uso UC_02 foi 84% realizado; e o caso de uso UC_03 foi 33% realizado.

O progresso funcional de cada fase do ciclo de vida do sistema é calculado abaixo (seção 4.3):

$$\mu_{Req} = \frac{1,1*1,00 + 1,057*1,00 + 1,891*0,50}{70,4} = 0,044$$

$$\mu_{A\&P} = \frac{1,4*1,00 + 1,391*1,00 + 3,672*0,75}{88} = 0,063$$

$$\mu_{Cod} = \frac{2,6*1,00 + 2,781*1,00 + 3,672*0,50}{158,5} = 0,045$$

$$\mu_{Tst} = \frac{0,6*1,00 + 0,334*0,50 + 1,891*0,00}{35,2} = 0,02$$

A partir dos dados acima, o progresso funcional da etapa de requisitos (*Req*) foi de 4,4%; a etapa de Análise e Projeto (*A&P*) em 6,3%; a etapa de Codificação (*Cod*) em 4,5%; e a etapa de Teste (*Tst*) em 2%. O cálculo do progresso funcional do sistema, $\mu_{Sistema}$, é obtido abaixo (seção 4.3).

$$\mu_{Sistema} = \frac{70,4*0,044 + 88*0,063 + 158,5*0,045 + 35,2*0,02}{352,1} = 0,046$$

A partir dos dados acima, em relação ao que foi estimado, o progresso funcional do sistema foi de 4,6%. Conclui-se que o Projeto *A* está em seu início.

5. Conclusões

Este trabalho apresentou a técnica TUCP, que é uma extensão da técnica UCP, tendo sido validada em uma organização certificada CMMI nível 3 em fevereiro de 2006. Em todos os mais de setenta projetos estimados com a TUCP, todos eles apresentaram um cálculo mais acurado para o esforço de projetos em relação à UCP, conforme dados da própria organização. A margem de erro nas estimativas calculadas nos projetos de software, que utilizam casos de uso (em torno de 10%), tem sido também semelhante ou superior à técnica de análise de pontos por função. Além disso, a TUCP permite uma



visão mais detalhada das estimativas em etapas do ciclo de vida do software, possibilitando um acompanhamento mais efetivo do projeto.

Como conclusões mais importantes deste trabalho, podem-se destacar:

- Uma base histórica de estimativas da organização deve ser implementada (como recomenda a literatura de métricas de software), para que sirva de fundamentação nas calibrações e nos fatores de produtividade para projetos futuros.
- A inexistência de padrões aceitos universalmente para a especificação de casos de uso dificulta a comparação entre projetos de diferentes organizações. Portanto, não há como garantir que os valores em UCP ou TUCP estarão medindo a mesma coisa, se os critérios utilizados para a especificação de casos de uso forem muito diversificados.
- A especificação de caso de uso deve ser descrita em um nível de detalhamento adequado (seção 2), para que a estimativa baseada em TUCP seja eficaz.
- Os fatores de ambiente (EF) passaram a estar relacionados apenas com o esforço; assim, o tamanho passa a depender somente de requisitos funcionais e não-funcionais.

Referências Bibliográficas

- Anda, B; et al. (2001) “Estimating software development effort based on use cases: experiences from industry”, In: International Conference on UML2001, 4. Proceedings. Toronto, Oct. 1 – 5.
- Anda, B. (2002) “Comparing effort estimates based on Use Case Points with expert Estimates”, Empirical Assessment in Software Engineering (EASE 2002). Keele, UK, p. 8-10.
- Boehm, B., (2000) “Software Cost Estimation With COCOMO II”, Prentice Hall, New Jersey.
- Champeaux, D. (1997) Object-Oriented Development Process and Metrics. Prentice Hall,.
- Cockburn, A. (2001) Writing effective: use cases. Addison-Wesley Boston.
- Damodaran, M; Washington A. (s.d.) “Estimation Using Use Case Points”, Computer Science Program. Texas – Victoria; University of Houston. Acesso em 11/04/2004: http://bfpug.com.br/Artigos/UCP/Damodaran-Estimation_Using_Use_Case_Points.pdf.
- Fowler, M;Scott, K. (2005) “UML Essencial - Um Breve Guia para a Linguagem-PAD”, Bookman.
- Garmus, D., Herron, D. (2000) “Function Point Analysis: Measurement practices for successful software projects”, Addison-Wesley: EUA.
- Karner, G. (1993) “Metrics for Objectory”, Diploma thesis, University of Linköping, Sweden. No. LiTH-IDA-Ex-9344:21.
- Kruchten, P. (2001) “The Rational Unified Process: an introduction”, Addison-Wesley.
- McPhee, C. (1999) “SENG 621: Software process management: software size estimation”, University of Calgary. 11p. Acesso em 18/08/04: http://sern.ucalgary.ca/~cmcphee/SENG621/Software_Size_Estimation.html.



- Meneses, J. B. (2001) *Inspector: Um Processo de Avaliação de Progresso para Projetos de Software*. Dissertação de Mestrado da UFPE. Recife
- Monteiro, T.C.; Pires, C.G.S.; Belchior, A.D. (2004a). “Estimations by Work Product Type: An extension of the UCP technique for the CMMI-SW level 2 and 3”. Metrics News issue, Berlin, Königs Wusterhausen. Disponível em: http://ivs.cs.uni-magdeburg.de/sweng/agruppe/forschung/mnews/news2004_1.pdf
- Monteiro, T.C.; Pires, C.G.S.; Belchior, A.D. (2004b). “An extension of estimates based on UCP attending the necessities of SW-CMM level 2”. Proceedings of International Conference on Computing, Communications and Control Technologies - CCCT 2004, Austin (Texas), USA.
- Monteiro, T.C.; Pires, C.G.S.; Belchior, A.D. (2004c). “Estimativas por Tipo de Produto de Trabalho”. Proceedings of XXX Latin American Conference of Informatics – CLEI, Arequipa, Peru.
- Monteiro, T.C.; Pires, C.G.S.; Belchior, A.D. (2004d). “Estimations by Work Product Type: An extension of the UCP technique for the CMMI-SW level 2 and 3”. METRICS NEWS, MetriKon 2004 (DASMA Metrik Kongress). Berlin, Königs wusterhausen. Disponível em: <http://iws2004.cs.uni-magdeburg>
- Monteiro, T. C. (2005a). “Pontos de Caso de Uso Técnicos (TUCP): uma Extensão da Técnica UCP”. Dissertação de Mestrado, Universidade de Fortaleza.
- Monteiro, T.C.; Pires, C.G.S.; Belchior, A.D. (2005b). TUCP: “Uma Extensão da Técnica UCP”. IV Simpósio Brasileiro de Qualidade de Software – SBQS 2005, Porto Alegre – RS.
- OMG. (2003) “Unified Modeling Language Specification”, version 1.5. March 2003. Disponível em: <http://www.omg.org>. Acesso em: 17/09/2004.
- Peters, K. (1999) “Software Project Estimation”, Software Productivity Centre Inc. (SPC) in Vancouver, British Columbia, Canada. Disponível em: <http://www.spd.ca/downloads/resources/estimates/estbasics.pdf>.
- Ribu, K. (2001) “Estimating Object-Oriented Software Projects with Use Cases”, Master of Science Thesis, University of Oslo, United States.
- Roetzheim, W. H. (2000) “Estimating Internet Development”, Software Development Magazine.
- RUP (2003) Rational Software Corporation, Rational Unified Process, Version 2003.06.00.65, CD-ROM, Rational Software, Cupertino, California, 2003.
- Schneider, G., Winters, J. (2001) “Applying Use Case: A Practical Guide”, 2nd ed. Addison-Wesley.
- SEI (2002). “CMMI-SW for Systems Engineering/Software Engineering ,Version 1.1 - CMU/SEI-2002-TR-012”, Disponível em: <http://www.sei.cmu.edu/publications/documents/02.reports/02tr002.html>.
- Vasquez, C. E. (2003) “Análise de ponto de função: medição, estimativas e gerenciamento de projetos de software”, 1ed. São Paulo: Érica.