

# **Estimativas por Tipo de Produto de Trabalho: uma Extensão da técnica PCU para o CMMI-SW Nível 2**

**Tatiana Cavalcanti Monteiro**  
Universidade de Fortaleza – UNIFOR, MIA  
Fortaleza, Ceará, Brasil  
[tatiana.m@uol.com.br](mailto:tatiana.m@uol.com.br)

e

**Carlo Giovano S. Pires**  
Instituto Atlântico, Dept. SPD  
Fortaleza, Ceará, Brasil  
[cgiovano@atlantico.com.br](mailto:cgiovano@atlantico.com.br)

e

**Arnaldo Dias Belchior**  
Universidade de Fortaleza – UNIFOR, MIA  
Fortaleza, Ceará, Brasil  
[belchior@unifor.br](mailto:belchior@unifor.br)

## **Resumo**

O CMMI-SW nível 2 recomenda a implantação de atividades de estimativas de tamanho, esforço, prazo e custo, como forma 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 PCU (Pontos por Caso de Uso), por exemplo, tem-se mostrado adequada para produtos de software orientados a objetos e baseados em casos de uso. No entanto, a granularidade dessa técnica para produtos de trabalho gerados nas atividades de planejamento e acompanhamento não se tem mostrado muito adequada. Este trabalho apresenta uma extensão da técnica PCU, para atender recomendações do CMMI-SW nível 2, permitindo uma visão mais detalhada das estimativas por tipo de produto de trabalho, possibilitando realizar refinamentos dessas estimativas ao longo do processo de desenvolvimento.

**Palavras-chave:** Estimativas, PCU, Casos de Uso, CMMI-SW.

## **Abstract**

The CMMI-SW level 2 requires estimation's use, including size, effort, schedule and cost, like the best way to improve software projects planning and tracking. Therefore there are many techniques for software estimations; this use isn't a common practice in software organizations. The UCP (Use Case Points), for example, is greatly suitable to object-oriented software and based in use case model. Nevertheless, the granularity offered by this method to the work's products types for software projects planning and tracking hadn't been adequate. This work presents an extension of UCP technique to attend CMMI-SW level 2, allowing detail view about the estimation by work's products types, offering to refine these estimations across the software development process.

**Keywords:** Estimation, UCP, Use Cases, CMMI-SW.

## 1. INTRODUÇÃO

Uma das práticas exigidas pelo CMMI-SW (*Capability Maturity Model Integration for Software*) nível 2 [2] é a realização de estimativas para tamanho, esforço, prazo e custo. Uma das grandes dificuldades na implantação das práticas de estimativa é a utilização de técnicas adequadas às exigências do modelo.

Algumas organizações têm adaptado técnicas de estimativas às suas necessidades, gerando técnicas proprietárias para atender a essas práticas, dificultando a troca de experiências, o uso de bases de conhecimentos de outras organizações, além do esforço despendido e do custo de definição dessas técnicas proprietárias.

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.

As estimativas podem ser enquadradas em dois grupos: *bottom-up* e *top-down*. As estimativas *bottom-up* são utilizadas para calcular cada item de trabalho, depois sumarizá-los ou agregá-los para se obter a estimativa total do projeto. Podem ser realizadas através da opinião de algum especialista ou através de uma base de dados, para determinar a complexidade e o esforço associados a uma determinada tarefa. As estimativas *top-down* avaliam o projeto como um todo, utilizando técnicas especializadas. A granularidade das estimativas dos produtos de trabalho é determinada pela estimativa de todo o projeto. Informações de projetos análogos são em geral utilizadas como base para a estimativa *top-down*.

Dentre as diversas técnicas para se estimar projetos de software, algumas delas tem-se consolidado como: Análise por Ponto de Função (*Functional Point Analysis* – FPA) [1], MKII *Function Points* [1] [9] e Pontos por Casos de Uso [11]. Esta última baseou-se nas duas primeiras técnicas e, apesar de ser uma técnica ainda recente, tem sido aprimorada e tem apresentado crescente utilização na indústria. A PCU é do tipo *top-down* e adequada a projeto que descrevem seus requisitos de software através de casos de uso.

Este trabalho apresenta uma extensão da técnica PCU, para atender recomendações do CMMI-SW nível 2 [2] – a TUPC (*Technical Use Case Point*) – permitindo uma visão mais detalhada das estimativas por tipo de produto de trabalho, possibilitando realizar refinamentos dessas estimativas ao longo do processo de desenvolvimento. Além disso, a proposta propõe calibrações nos fatores de produtividade por tipo de produto de trabalho, permitindo melhorar essas estimativas.

Este trabalho está organizado em cinco seções. Na seção 2, descreve-se como as estimativas são apresentadas no CMMI-SW nível 2. Na seção 3, delinea-se a técnica PCU. Na seção 4, discorre-se sobre a extensão da PCU proposta, a TUCP. Na seção 5, apresenta-se um estudo de caso, utilizando-se a TUCP. Finalmente, na seção 6, são delineadas as conclusões do trabalho.

## 2. ESTIMATIVAS NO CMMI-SW NÍVEL 2

O CMMI-SE é um modelo para avaliação e melhoria da maturidade dos processos de uma organização, como uma integração e evolução dos modelos: SW-CMM (*Capability Maturity Model for Software*), SECM (*System Engineering Capability Model*), e IPD-CMM (*Integrated Product Development CMM*) [2].

O CMMI-SW fornece às organizações de software um guia de como obter controle em seus processos para desenvolver e manter software, e como evoluir em direção a uma cultura de engenharia de software e excelência de gestão. O CMMI foi projetado para guiar as organizações de software no processo de seleção das estratégias de melhoria, determinando a maturidade atual do processo e identificando as questões mais críticas para a qualidade e melhoria do processo de software [2].

O CMMI tendo em vista a realidade e as necessidades das organizações, e visando fornecer uma maior flexibilidade, admite duas abordagens: uma por estágio, como o CMM tradicional, e outra contínua, semelhante a ISO/IEC 15504 [3]. A representação por estágios contém cinco níveis de maturidade: 1-inicial, 2-gerenciado, 3-definido, 4-gerenciado quantitativamente, e 5-otimizado.

O CMMI é organizado em cinco níveis de maturidade que definem uma escala ordinal para medir a maturidade de um processo de software da organização e para avaliar a sua capacidade. Os níveis também ajudam uma organização a priorizar seus esforços de melhorias. Cada nível compreende um conjunto de objetivos de processos que, quando satisfeitos, estabilizam um componente importante do processo de software. Alcançando cada nível da estrutura de maturidade, estabelecem-se diferentes componentes no processo de software, resultando em um crescimento na capacidade de processo da organização.

A representação por estágio do CMMI-SW possui as seguintes áreas de processo para o nível 2 (gerenciado): gerência de requisitos, planejamento de projeto, monitoração e controle de projeto, gerência de acordo com fornecedores, medição e análise, garantia da qualidade do processo e do produto, e gerência de configuração. A representação contínua do CMMI-SW possui seis níveis de maturidade: incompleto, executado, gerenciado, definido, gerenciado quantitativamente, e otimizado [2].

As áreas de processo do CMMI-SW podem ser agrupadas em quatro categorias: gerenciamento de processo, gerenciamento de projeto, engenharia, e suporte. As áreas de processos relacionadas à gerência de projetos contêm as atividades de projetos referentes ao planejamento, acompanhamento e controle do projeto. São elas: planejamento de projeto, acompanhamento e controle do projeto, gestão de acordos com fornecedores, gestão integrada de projeto, gestão de riscos, integração da equipe, gestão quantitativa de projeto, e gestão integrada de fornecedores [2].

No CMMI-SW, como também no SW-CMM [2], o processo de estimativas estabelece uma base formal para o planejamento e o acompanhamento de projetos de software, a partir de quatro atividades: (i) estimar o tamanho do produto a ser gerado; (ii) estimar o esforço empregado na execução do projeto; (iii) estimar a duração do projeto; e (iv) estimar o custo do projeto. Essas atividades são referenciadas no CMMI-SW nível 2 [2], nas áreas de processo de planejamento de projeto (*Project Planning* - PP) e monitoração e controle de projeto (*Project Monitoring and Control* - PMC) .

No CMMI-SW nível 2, as estimativas de tamanho devem ser feitas para todas as principais atividades e produtos de software, e realizadas sobre produtos com granularidade fina e adequada para um acompanhamento contínuo. No entanto, este modelo não especifica o tipo de medição que deve ser aplicada, nem a granularidade em que os produtos de trabalho devem ser decompostos. Para que um projeto seja melhor estimado, seus produtos de trabalho devem ser decompostos até uma granularidade necessária para se alcançar os objetivos da estimativa [2].

O CMMI-SW exige que se planeje e acompanhe as atividades do processo de desenvolvimento de software, através da adoção de métodos e modelos de ciclo de vida que guiem as atividades de desenvolvimento, a documentação dos produtos de trabalho e o planejamento [2].

O planejamento deve englobar todas as atividades do desenvolvimento, além de estabelecer indicadores e métricas de qualidade. Esses indicadores permitem acompanhar o desempenho real dos projetos, comparando-os aos seus valores estimados e planejados. Nos casos em que os resultados reais divergem dos planejados, ações corretivas devem ser tomadas e gerenciadas até a sua efetiva conclusão.

A seguir, será apresentada a técnica de pontos por casos de uso (PCU), que será utilizada como base para o planejamento e acompanhamento de projetos em organizações focadas no CMMI-SW.

### 3. PONTOS POR CASOS DE USO (PCU)

Os projetos de software orientado a objetos já utilizam, comumente, os diagramas de casos de uso (use cases) para descrever as funcionalidades do sistema de acordo com a forma de utilização por parte dos usuários. Tomando como base os casos de uso, uma técnica para estimativa de projeto foi proposta em 1993 por Gustav Karner [4]. Essa técnica permite estimar o tamanho do sistema ainda na fase de levantamento de casos de uso, utilizando-se dos próprios documentos gerados como subsídio para o cálculo dimensional.

Apesar de ser uma técnica recente, o PCU tem sido estudado por vários pesquisadores no meio acadêmico e na indústria. Em [11], os resultados da aplicação de PCU na estimativa de alguns projetos comerciais são relatados.

Para essa técnica ser eficaz, os casos de uso devem ser descritos em um nível de detalhamento apropriado, pois isto influencia diretamente na qualidade final da estimativa. Portanto, deve haver a preocupação em medir casos de uso em nível de sistema – onde seja possível diferenciar transações e interações com o usuário. Desta forma, casos de uso de nível muito alto (modelagem de negócio do sistema) ou muito baixo (decomposição funcional dos casos de uso) não se demonstram adequados para a medição.

Várias pesquisas têm sido realizadas nos últimos anos, discutindo o nível de decomposição ideal dos casos de uso para a aplicação da técnica. Este é o principal problema da técnica PCU: na maioria dos casos, caberá aos analistas decidir a granularidade ideal dos casos de uso utilizados para a medição. Uma técnica de descrição dos casos de uso muito utilizada em [8] apresenta como deve ser representado um requisitos através de um caso de uso.

Uma vez que os casos de uso principais do sistema sejam levantados e descritos, é possível estimar o tamanho total do software baseando-se em um conjunto simples de métricas. As atividades necessárias para a geração da estimativa baseada na técnica PCU são decompostas nas seguintes atividades:

- Classificar os atores;
- Classificar os casos de uso;
- Definir os fatores técnicos e de ambiente.

A primeira atividade é classificar os atores envolvidos em cada caso de uso, obtendo-se um somatório dos pontos não-ajustado. A classificação desses atores está baseada na Tabela 1. O peso total dos atores do sistema (*Unadjusted Actor Weight* - UAW) é calculado pela soma dos produtos do número de atores de cada tipo pelo respectivo peso.

**Tabela 1 - Classificação dos Atores**

| Tipo de Ator  | Descrição                                                                            | Peso |
|---------------|--------------------------------------------------------------------------------------|------|
| Simple        | Aplicação com API definida                                                           | 1    |
| Intermediário | Outro sistema interagindo através de um protocolo de comunicação, como TCP/IP ou FTP | 2    |
| Complexo      | Um usuário interagindo através de uma interface gráfica ( <i>stand-alone</i> ou Web) | 3    |

Uma vez calculado o peso dos atores do sistema, a atividade seguinte é classificar os casos de uso e calcular o peso bruto dos mesmos (*Unadjusted Use Case Weight* - UUCW). Para fins de cálculo, dividem-se os casos de uso em três níveis de complexidade, de acordo com o número de transações envolvidas em seu processamento.

Por transação, entende-se como uma série de processos que devem ser realizados em conjunto ou cancelados em sua totalidade, caso não seja possível concluir o processamento. A Tabela 2 apresenta o peso para cada um dos tipos de caso de uso classificados.

**Tabela 2 - Classificação dos Casos de Uso**

| Tipo de Caso de Uso | Número de Transações  | Peso |
|---------------------|-----------------------|------|
| Simple              | Até 3 transações      | 1    |
| Intermediário       | De 4 a 7 transações   | 2    |
| Complexo            | Acima de 7 transações | 3    |

O cálculo do UUCW é realizado de forma similar ao cálculo do peso dos atores, somando os produtos da quantidade de casos de uso classificados em cada tipo pelo peso nominal do tipo em questão.

Para calcular os pontos de casos de uso não ajustados (*Unadjusted Use Case Points* - UUCP) apresentados na Eq. (1), deve-se somar o valor obtido na medição de atores com o valor obtido na medição de casos de uso.

$$UUCP = UAW + UUCW \quad \text{Eq. (1)}$$

A atividade seguinte trata de definir os fatores técnicos (*Technical Complexity Factor* – TCF) e o de ambiente (Environmental Factors - EF). Os fatores técnicos (TCF) medem a complexidade do projeto em relação aos requisitos não-funcionais. Esses fatores técnicos influenciam no resultado do PCU, apresentados em Eq.(2). Segundo Karner [4], os fatores de complexidade do projeto são as características relacionadas à performance, portabilidade, segurança, reusabilidade de código, entre outras.

Já os fatores de ambiente (EF) estão relacionados com a familiaridade com o processo de desenvolvimento a ser utilizado no projeto, a experiência na aplicação, motivação, requisitos estáveis, entre outros.

Para calcular o fator de complexidade técnica do sistema (TCF), deve-se primeiro calcular o TFactor. O TFactor é o somatório do produto entre o valor atribuído no projeto para cada item e seu peso. O TCF é calculado por:

$$TCF = 0,6 + (0,01 * TFactor) \quad \text{Eq. (2)}$$

Para calcular os fatores de ambiente - EF apresentado em Eq.(3) deve-se primeiro calcular o EFactor. O EFactor é o somatório do produto entre o valor atribuído no projeto para cada item e seu peso [11]. O EF é calculado por:

$$EF = 1,4 + (-0,03 * EFactor) \quad \text{Eq. (3)}$$

Os pontos de caso de uso (PCU) são calculados por:

$$PCU = UUCP * TCF * EF \quad \text{Eq. (4)}$$

O Esforço apresentado em Eq.(5) é obtido pelo produto do tamanho em PCU pela produtividade, que é calculado por:

$$\text{Esforço} = \text{PCU} * \text{Produtividade} \quad \text{Eq. (5)}$$

As contagens de PCU em Eq.(4) podem variar entre organizações e indivíduos, devido à mencionada variação nos estilos de casos de uso. É então razoável supor que a produtividade associada ao desenvolvimento de 1 PCU (20 homens-horas, no trabalho original de Karner [4]) também varie bastante. Desta forma, a obtenção de estimativas confiáveis de esforço exigiria a padronização dos estilos de casos de uso em um extenso trabalho de calibração do modelo de estimativas baseado em PCU.

A seguir, será desenvolvida a abordagem para estimativas de produtos de trabalho baseado em PCU, a TUCP no contexto do CMMI-SW.

#### 4. UMA ABORDAGEM PARA ESTIMATIVAS DE PRODUTOS DE TRABALHO BASEADA EM PCU

As estimativas podem ser realizadas em vários pontos do ciclo de vida, e em um subconjunto dos principais produtos de trabalho, dependendo de características e estratégias do projeto. Quanto aos pontos do ciclo de vida, é possível realizar as estimativas na fase de proposta com requisitos de alto nível e heurísticas [11], depois refinar após detalhamento de requisitos, ou refinar após a especificação de casos de uso, ou até mesmo após análise e projeto.

Conforme descrito na seção 2, o CMMI-SW nível 2 não define quais são os produtos de trabalho e nem o nível de granularidade necessário para se alcançar os objetivos das estimativas. As empresas são responsáveis por determinar esse nível de granularidade em seus produtos de trabalho de acordo com os objetivos e as estratégias necessárias de cada organização.

A técnica original PCU apresenta a estimativa de tamanho para todo o projeto em PCU e a estimativa de esforço pode ser obtida através de fatores de produtividade. No entanto, esta granularidade não permite um planejamento detalhado e nem um acompanhamento efetivo por produtos de trabalho.

Neste contexto, este trabalho propõe uma extensão da técnica PCU, para atender recomendações do CMMI-SW nível 2, permitindo uma visão mais detalhada das estimativas por tipo de produto de trabalho, possibilitando realizar refinamentos dessas estimativas ao longo do processo de desenvolvimento. Essa extensão, a TUCP (Technical Use Case Point) compreende os quatro pontos a seguir:

- Estimativa de tamanho do projeto;
- Estimativas de tamanho por produtos de trabalho;
- Estimativa de esforço com fator de produtividade diferenciado;
- Estimativas de prazo e custo.

Os quatro pontos propostos pela técnica TUCP serão apresentados abaixo, juntamente com um exemplo de utilização da mesma.

##### 4.1 Estimativa de Tamanho do Projeto

Pontos de casos de uso ajustados (PCU) apresentados em Eq.(4) consideram o peso dos diferentes tipos de interface através dos atores, o peso dos requisitos funcionais através dos casos de uso, o peso dos requisitos não-funcionais através dos fatores técnicos, e o peso da equipe e ambiente de desenvolvimento através dos fatores de ambiente. Um problema identificado com esta abordagem é que o uso dos fatores de ambiente - EF pode levar a diferentes tamanhos, dependendo da equipe ou empresa que desenvolver o projeto. Apenas os fatores técnicos e não-técnicos deveriam ser levados em consideração para a estimativa de tamanho, evitando gerar dependência do tamanho do software com características de empresa e equipe de desenvolvimento. De acordo com a proposta desse trabalho, os fatores de ambientes devem ser considerados apenas na estimativa de esforço.

A técnica proposta define a medida TUCP (*Technical Use Case Point*) para obtenção de um valor de tamanho de projeto baseado apenas nos requisitos funcionais e nos requisitos não-funcionais do sistema. Essa medida é obtida pelo ajuste do UUCP apenas pelos fatores técnicos - TCF.

$$\text{TUCP} = \text{UUCP} * \text{TCF} \quad \text{Eq. (6)}$$

##### 4.2 Estimativas de Tamanho por Produtos de Trabalho

Para a realização de estimativas de produtos de trabalho, deve-se inicialmente definir os tipos de produto. De acordo com os modelos SW-CMM [6][7] e CMMI-SW [2], um projeto de desenvolvimento de software trabalha com quatro tipos principais de produtos de engenharia: Requisitos, Análise e Projeto, Codificação, e Teste. Um tipo de produto de trabalho ainda pode ser categorizado em subtipos.

Por exemplo, para o tipo de produto de trabalho do tipo Análise e Projeto, podemos ter os subtipos: projeto gráfico, especificação de arquitetura, projeto detalhado, especificação de banco de dados, especificação de IHC (Interface Humano-Computador) para cada caso de uso. Desta forma, a granularidade torna-se mais fina para estimar os casos de uso, sendo mais fácil o acompanhamento do projeto de software e permitindo ações corretivas antes do final de toda a análise e projeto.

Com base nos tipos de produto, as estimativas de tamanho podem ser realizadas para cada produto de caso de uso. Por exemplo, um projeto com o caso de uso Autenticar Usuário e os tipos de produtos Requisitos, Análise e Projeto, Codificação e Teste poderia ter os seguintes produtos de trabalho:

- Especificação do Caso de Uso Autenticar Usuário: Descrição, Regras de negócio, fluxos, pré-condições, pós-condições, entre outras seções.
- Projeto do Caso de Uso Autenticar Usuário: Realização do caso de uso com diagramas de classe e sequência para implementação do caso de uso.
- Código do Caso de Uso Autenticar Usuário: Classes definidas no projeto do caso de uso e implementadas na linguagem definida.
- Teste do Caso de Uso Autenticar Usuário: Casos de teste para os cenários de execução.

De acordo com a técnica PCU, o esforço do projeto é diretamente proporcional ao seu tamanho em pontos de caso de uso. Assim sendo, o tamanho de um produto de caso de uso pode ser baseado no percentual de esforço empregado para seu desenvolvimento (ver exemplo de medições de percentual de esforço por tipo de produto na Tabela 3).

Medições realizadas para o projeto, ou por tipo de projeto podem ser utilizadas para determinar um percentual de distribuição adequado para a instituição. O tamanho de cada produto de caso de uso é então definido de forma proporcional a seu fator de complexidade e o seu percentual de distribuição de esforço por tipo de produto.

$$\text{TUCP}(\text{produto de caso de uso}) = \frac{\text{Peso do Caso de Uso}}{(\text{Soma de Todos os Pesos de Casos de Uso})} * \text{TUCP} * \text{PEP} \quad \text{Eq. (7)}$$

O tamanho do produto de trabalho por caso de uso definido em Eq.(7) é dado pelo produto do resultado da divisão entre o peso do caso de uso sobre a soma do peso de todos os casos de uso do sistema, com o tamanho de todo o sistema - TUCP na Eq.(6) e com o percentual de esforço por produto - PEP (Veja Tabela 3).

#### 4.3 Estimativa de Esforço com Fator de Produtividade Diferenciado

Para obter a estimativa de esforço, o método PCU utiliza um fator de multiplicação (ou produtividade) comum para todos os tipos de atividade de um projeto. No entanto, a produtividade pode ser variável, dependendo do tipo de projeto e da equipe que realiza cada tipo de atividade. Por exemplo, em uma organização que possui *frameworks* ou componentes reutilizáveis, a produtividade de atividades de codificação costuma ser alta.

No entanto, as atividades de “especificação de requisitos” e “análise e projeto” não serão agilizadas devido ao reuso de código. Portanto, a organização deverá, a partir das medições de esforço realizado para cada tipo de atividade e os TUCP’s calculados para os projeto, manter uma base de dados com fator de produtividade para cada tipo de atividade e característica de projeto.

O Esforço para gerar um produto de caso de uso é calculado por:

$$\text{Esforço} = \text{TUCP}(\text{produto de caso de uso}) * \text{EF} * \text{FPP} \quad \text{Eq. (8)}$$

O Esforço definido em Eq.(8) gasto em um produto de trabalho por caso de uso é dado pelo produto do tamanho do caso de uso - TUCP apresentado em Eq.(7) com o fator de ambiente do sistema – EF apresentado em Eq.(3) e com o fator de produtividade do tipo de produto de trabalho - FPP (Veja Tabela 4).

#### 4.4 Estimativas de Prazo e Custo

As estimativas de prazo e custos devem ser realizadas a partir do esforço estimado com a técnica proposta, disponibilidade de recursos, restrições do projeto e custos homens-horas. Note que essa estimativa pode ser feita para o projeto como um todo ou para cada produto de trabalho através do esforço estimado para o produto.

#### 4.5 Exemplo de TUCP

Para exemplificar a técnica, vamos considerar o caso de uso Autenticar Usuário como intermediário, com uma interface gráfica (ator Complexo) em uma empresa que possui medições e características de projeto de acordo com a Tabela 3 e Tabela 4, uma equipe produtiva e um fator técnico de complexidade mediano. Vamos considerar também que existe o caso de uso Cadastrar Usuário de característica Simples.

**Tabela 3 - Percentual de esforço por produto - PEP**

| Tipo de produto   | Percentual de Esforço |
|-------------------|-----------------------|
| Requisitos        | 20%                   |
| Análise e Projeto | 30%                   |
| Codificação       | 35%                   |
| Testes            | 15%                   |

**Tabela 4 - Fatores**

| Fator                                                                                 | Valor |
|---------------------------------------------------------------------------------------|-------|
| EF                                                                                    | 1,00  |
| Soma de Pesos:<br><i>Peso(Autenticar Usuário) +</i><br><i>Peso(Cadastrar Usuário)</i> | 15,00 |
| TUCP de todo o sistema                                                                | 18,36 |
| Fator de produtividade - FPP<br>(Análise e Projeto)                                   | 15,00 |

Com estas características, o tamanho total do projeto seria de 18,36 TUCP's e para o caso de uso teríamos:

**TUCP(Projeto do Caso de Uso Autenticar Usuário)** =  $(\text{Peso}(\text{Autenticar Usuário}) / (\text{Peso}(\text{Autenticar Usuário}) + \text{Peso}(\text{Cadastrar Usuário}))) \times (\text{TUCP de todo o sistema}) \times (\text{Percentual de Esforço na Análise e Projeto}) = (10/(10 + 5)) \times 18,36 \times 0,30 = \mathbf{3,62 \text{ TUCPs}}$

Considerando o fator de produtividade da atividade de análise e projeto como 15 teríamos um esforço de:

**Esforço** =  $(\text{Autenticar Usuário}) = \text{TUCP}(\text{Autenticar Usuário}) * \text{EF} * \text{FPP}(\text{Análise e Projeto}) = 3,62 * 1 * 15 = \mathbf{54,03 \text{ homens-horas}}$  para a análise e projeto do caso de uso *Autenticar Usuário*.

## 5. ESTUDO DE CASO

Esse estudo de caso foi realizado em uma organização de pesquisa e desenvolvimento certificada como SW-CMM nível 2 que utiliza processo RUP (Rational Unified Process) [5] [10] em projetos de pesquisa e desenvolvimento em diversas áreas de tecnologia da informação e telecomunicações, em diferentes plataformas (J2EE, J2ME e .NET).

A técnica de estimativa proposta neste artigo é aplicada em todos os projetos de software da instituição. Para esse estudo de caso, serão apresentados dois projetos com características e tamanhos diferentes. Os projetos serão denominados A e B por questões de confidencialidade.

O esforço estimado do projeto A foi de 28.000 homens-horas e tem como característica o desenvolvimento de um sistema para a automação de processos. O esforço estimado do projeto B foi de 2.000 homens-horas e é um sistema para cadastro e consulta de materiais que utilizou em seu desenvolvimento um framework com alto reuso e geração de código.

Para calcular o percentual de erro estimado para cada atividade utilizando a técnica proposta, foi utilizada a métrica *Symmetric Relative Error* (SER) proposta por M. Jorgensen e D. Sjobeg [11]. A fórmula é dada por:

$$\begin{aligned} \text{SER} &= \text{Real} - \text{Estimado} / \text{Real}, \text{ se } \text{Real} \leq \text{Estimado} \\ \text{SER} &= \text{Real} - \text{Estimado} / \text{Estimado}, \text{ se } \text{Real} > \text{Estimado}, \end{aligned}$$

onde “Real” é o esforço real do projeto e “Estimado” é o esforço estimado utilizando a técnica proposta neste artigo.

Os resultados apresentados nas Tabela 5 e Tabela 6 mostram o percentual de erro relativo entre valores estimados e realizados. Para efeitos de comparação, a Tabela 5 apresenta valores estimados com o fator de produtividade único para todos os tipos de produto, enquanto a Tabela 6 apresenta o fator de produtividade calibrado por tipo de produto de trabalho.

Na Tabela 6 os resultados apresentados mostram que o percentual de erro total em relação ao real é muito menor se comparado com a Tabela 5, onde não existiu a distribuição do fator de produtividade.

**Tabela 5 - Percentual de erro sem distribuição de produtividade por tipo de produto de trabalho**

| Projeto | REQ    | A&P    | COD    | TESTE  | Total  |
|---------|--------|--------|--------|--------|--------|
| A       | -2,00  | 24,88  | -49,03 | 29,08  | -11,05 |
| B       | -47,81 | -86,26 | -32,96 | -52,59 | -46,61 |

**Tabela 6 - Percentual de erro com distribuição de produtividade por tipo de produto de trabalho**

| Projeto | REQ    | A&P   | COD    | TESTE  | Total |
|---------|--------|-------|--------|--------|-------|
| A       | -1,93  | 24,81 | -11,76 | 29,08  | 3,24  |
| B       | -10,85 | 7,37  | 0,29   | -34,61 | -3,59 |

A calibração no fator de produtividade utilizada na Tabela 6 considera que o fator de produtividade nos projetos A e B foram

- Fator de produtividade de requisitos: entre 15 e 20 homens-horas.
- Fator de produtividade de análise e projeto: entre 20 e 25 homens-horas.
- Fator de produtividade de codificação: entre 15 e 20 homens-horas.
- Fator de produtividade de testes: entre 10 e 15 homens-horas.

Estes valores podem ser redistribuídos caso as características do projeto e da equipe influenciem em alguma atividade durante o desenvolvimento de software. Segundo a Tabela 6, os valores apresentados no projeto A para os produtos análise e projeto e teste tiveram um percentual de erro maior do que o real.

Apesar dos desvios nos percentuais de erros em alguns produtos de trabalho serem maior do que 20%, o desvio total no esforço é muito pequeno se comparado com os valores individuais para cada produto de trabalho.

Para o projeto B (ver Tabela 6), os percentuais dos erros relativos para “análise e projeto” e “codificação” foram baixos. Isso ocorreu devido à distribuição realizada nos principais produtos de trabalho, que levou em consideração que a “análise e projeto” de um sistema de consulta e cadastro exige um esforço menor. Na codificação, o esforço também foi baixo devido à geração de código através de um *framework*, que permitiu um desenvolvimento mais rápido.

Analisando a Tabela 5 e a Tabela 6, podemos verificar que a distribuição da produtividade por tipo de produto de trabalho auxiliou na precisão da estimativa de esforço. No entanto, uma base histórica para uso na calibração da produtividade é importante, pois fatores como características de projeto, plataforma utilizada e desempenho da equipe podem influenciar neste fator de produtividade.

## 6. CONCLUSÃO

Esse trabalho apresentou uma extensão da técnica PCU, para atender recomendações do CMMI-SW nível 2, permitindo uma visão mais detalhada das estimativas por tipo de produto de trabalho, possibilitando realizar refinamentos dessas estimativas ao longo do processo de desenvolvimento.

Essa técnica, TUCP, pode ser utilizada em projetos que utilizam casos de uso para especificação de requisitos de software.

As principais contribuições deste trabalho foram:

- A extensão do método PCU através da TUCP permitiu uma visão mais detalhada das estimativas por tipo de produto de trabalho;
- O fator de produtividade por tipo de produto de trabalho gerou menor erro na estimativa total do projeto e na estimativa de cada produto de trabalho, possibilitando assim um planejamento e acompanhamento mais efetivos;
- A abordagem proposta possibilita fazer refinamentos nas estimativas nas várias fases do processo de desenvolvimento;
- Um fornecimento de uma técnica de estimativa aderente às práticas dos modelos CMMI-SW, como também do SW-CMM [6][7].

Como conclusões mais importantes deste trabalho, pode-se citar:

- Uma base histórica com as estimativas da organização deve ser implementada (como recomenda a literatura de métricas de software), afim de que sirvam de fundamentação nas calibrações, e nos fatores de produtividade para projetos futuros;
- A elaboração do caso de uso deve ser descrita em um nível de detalhamento adequado, para que a estimativa baseada em PCU seja eficaz;

- A inexistência de padrões aceitos universalmente para a construção de casos de uso dificulta a comparação entre projetos de diferentes organizações. Não há como garantir que os TUCP's estarão medindo a mesma coisa, se os critérios utilizados para construir os casos de uso forem muito diversificados.

### Referências Bibliográficas

- [1] Anda, B. *Comparing Effort estimates Based on Use Case Points with Expert Estimates*, 4th International Conference, Toronto, Canada, October 1-5, 2001, LNCS 218, 2001.
- [2] *CMMI<sup>SM</sup> for Systems Engineering/Software Engineering*, Version 1.1, Staged Representation (CMMI-SE/SW, V1.1, Staged). Disponível em: <http://www.sei.cmu.edu/publications/documents/02.reports/02tr002.html>. Acesso em 27/04/2004.
- [3] *ISO/IEC 15504: Information Technology – Process Assessment*, Part 1 to Part 5, 2003.
- [4] Karner, G. *Metrics for Objectory*. Diploma thesis, University of Linköping, Sweden. No. LiTH-IDA-Ex-9344:21. December 1993.
- [5] Kruchten, Philippe. *The Rational Unified Process - An Introduction*. 2nd Ed. New Jersey: Addison-Wesley, 2000.
- [6] Paulk, M. C. et al. *Capability Maturity Model for Software*, Version 1.1, SEI-CMU-93-TR-24, Software Engineering Institute, 1993.
- [7] Paulk, M., Weber, C. *The Capability Maturity Model: Guidelines for Improving the Software Process*, 17th ed. Addison-Wesley. ISBN 0-201-54664-7, 2003.
- [8] Schneider, G., Winters, J. *Applying Use Case: A Practical Guide*. 2nd ed. Addison-Wesley, 2001. ISBN 0-201-70853-1.
- [9] Symons, C.R. *Software Sizing and Estimating*, MKII FPA. John Wiley and Sons, 1991.
- [10] Rational Software Corporation, *Rational Unified Process*, version 2001.3, CD-ROM, Rational Software, Cupertino, Calif.:2001.
- [11] Ribu, K. *Estimating Object-Oriented Software Projects with Use Cases*. Masters Thesis, University of Oslo. November 2001.