

# Estendendo a Contagem de Pontos de Caso de Uso para Aplicação na Terceirização do Desenvolvimento de Software

Gustavo Bestetti Ibarra<sup>1,2</sup>, Patrícia Vilain<sup>1</sup>

<sup>1</sup>Departamento de Informática e Estatística (INE) – Universidade Federal de Santa Catarina (UFSC)

<sup>2</sup>Tribunal Regional do Trabalho de Santa Catarina (TRT/SC)

`gustavo.ibarra@trt12.jus.br`, `vilain@inf.ufsc.br`

**Abstract.** *This paper points out some problems of the process of estimating the size of software projects by use case points and proposes a new metric for classifying use cases that can replace the original classification. This metric uses technical variables based on user interface prototypes, business rules and preliminary class diagrams. The modification proposed to the use case points technique is devoted to the negotiation process with companies that outsource software development. This work was elaborated based on evidences and experiments that took place during the development of software projects at the Labor Court of the State of Santa Catarina (TRT/SC).*

**Resumo.** *Este artigo aponta alguns problemas do processo de estimativas do tamanho do projeto de um software utilizando pontos de caso de uso e propõe uma nova métrica para a classificação dos casos de uso que pode substituir a classificação original. Esta métrica utiliza variáveis técnicas que podem ser medidas através de protótipos de tela, regras de negócio e diagramas de classes conceituais. A modificação proposta para a técnica de pontos de casos de uso é voltada ao processo de negociação com empresas que terceirizam o desenvolvimento de software. Este artigo foi escrito baseando-se em evidências e experiências realizadas durante o desenvolvimento de projetos de software no Tribunal Regional do Trabalho de Santa Catarina (TRT/SC).*

## 1. Introdução

A terceirização é hoje um desafio para as empresas públicas e privadas brasileiras que buscam reduzir seus custos de produção e aumentar sua produtividade e lucros. Sérgio Pinto Martins [MARTINS 2009] afirma: "No Brasil, a noção de terceirização foi trazida por multinacionais por volta de 1950, pelo interesse que tinham em se preocupar apenas com a essência do seu negócio. A indústria automobilística é exemplo de terceirização, ao contratar a prestação de serviços de terceiros para a produção de componentes do automóvel, reunindo peças fabricadas por aqueles e procedendo à montagem final do veículo". Este mesmo processo está começando a ocorrer tanto nas empresas públicas, quanto nas empresas privadas, quando se trata de desenvolvimento de software.

No caso das empresas privadas, esta medida justifica-se principalmente em função dos altos encargos incidentes sobre a folha de pagamento e na complexidade da legislação trabalhista. Já no caso das empresas públicas, a terceirização é quase que uma necessidade institucional, pois com as políticas de Estado Mínimo estas empresas estão sofrendo fortes pressões para reduzir gastos e pessoal [ALVES 2002], e na maioria das

vezes estas reduções afetam as áreas “meio” destas empresas, e conseqüentemente as áreas de TI. Ainda no cenário público, as terceirizações apresentam maior complexidade, pois só podem ser feitas através de processos licitatórios os quais são regidos por lei [BRASIL 1993] e são pouco flexíveis.

Outro fator complicador do processo de terceirização do desenvolvimento de software está na dificuldade de se estabelecer estimativas mais precisas, as quais são muito importantes para o estabelecimento dos contratos de desenvolvimento de software. O processo de estimativa torna-se ainda mais importante na esfera pública em função das recomendações do TCU (Tribunal de Contas da União) que condenam a contratação por horas trabalhadas [BRASIL 2006].

Diversos métodos de estimativas já foram propostos, como Linhas de Código, Pontos de Função (FP) [ALBRECHT 1979], Pontos de Caso de Uso (UCP) [KARNER 1993], Story Points e Ideal Days [COHN 2005]. Linhas de Código foi uma das primeiras técnicas porém possui forte dependência das características de cada desenvolvedor e das características das linguagens de programação. Já as técnicas utilizadas em métodos ágeis, como Planning Poker, Story Points e Ideal Days são muito subjetivas e dependem do “sentimento” da equipe de desenvolvimento do sistema [ØSTVOLD et al. 2008] em relação a funcionalidade a ser desenvolvida.

A técnica de Pontos de Caso de Uso foi proposta por Gustav Karner [KARNER 1993] [KARNER 1993b] como uma adaptação do método de Pontos de Função [ALBRECHT 1979] [VAZQUEZ et al. 2003] para aplicação em projetos orientado a objetos a partir de casos de uso levantados na fase de requisitos [VIEIRA 2007]. No entanto, para a análise da complexidade dos casos de uso, as definições propostas por Karner não são muito objetivas e dependem muito da forma como os casos de uso são escritos e detalhados [ANDA et al. 2001] [VIEIRA E WAZLAWICK 2006]. Outro ponto limitante na proposta de Karner é a necessidade do detalhamento dos casos de uso, fato que muitas vezes não ocorre nas fases iniciais dos projetos e também com empresas que optam por uma abordagem mais ágil no seu processo de desenvolvimento de software.

Observando estas limitações, este artigo apresenta uma nova métrica para a classificação dos casos de uso segundo sua complexidade para ser aplicada em conjunto com a métrica definida por Karner, produzindo, assim, estimativas mais precisas para os projetos desenvolvidos. Esta métrica baseia-se em algumas variáveis técnicas que podem ser medidas através de protótipos de tela e diagramas de classes conceituais, ao invés de casos de uso expandidos.

A métrica proposta neste artigo é resultante da análise do desenvolvimento de projetos no Tribunal Regional do Trabalho de Santa Catarina (TRT/SC) realizados em 2008 e 2009, e pode ser utilizada na elaboração de contratos de desenvolvimento de software e editais de licitação, tornando assim mais objetivo o controle dos custos do projeto.

O restante do artigo está organizado como segue. A seção 2 apresenta o processo de desenvolvimento de software no TRT/SC. A seção 3 introduz a métrica de contagem de pontos de caso de uso. A seção 4 apresenta a extensão do processo de estimativa proposta neste artigo. A seção 5 apresenta um estudo de caso. A seção 6 conclui o artigo.

## 2. Processo de Desenvolvimento de Software no TRT/SC

O processo atual de desenvolvimento do TRT/SC é composto das fases de iniciação, concepção, elaboração, construção, testes, homologação, implantação e estabilização (pós-implantação). A fase de iniciação caracteriza-se pelo estudo de viabilidade do projeto. As fases de concepção, elaboração, construção, testes, homologação, implantação e estabilização, são realizadas de forma iterativa e incremental, onde em cada iteração um conjunto de casos de uso é desenvolvido. Como o foco deste artigo está no processo de estimativa e custo efetivo do projeto, serão comentadas somente as atividades e artefatos importantes para este processo.

Durante a concepção do projeto, as equipes realizam sessões de modelagem [AMBLER 2004] com os usuários para levantamento dos requisitos do sistema. O resultado destas sessões é concretizado com a produção dos artefatos listados na Tabela 1.

**Tabela 1. Artefatos mínimos para análise de complexidade dos casos de uso**

Artefato
- Diagrama ou Lista de Casos de Uso
- Protótipos de telas para cada caso de uso
- Descrição geral dos casos de uso (opcional)
- Relação de requisitos de interface e regras de negócio de cada caso de uso
- Diagrama de classes conceituais geral ou específico para cada caso de uso

A proposta definida neste artigo é utilizada ao longo do processo de desenvolvimento dos sistemas do TRT/SC onde podem ser terceirizadas as fases de elaboração e construção. Para isso, na medida em que os artefatos são produzidos (para cada caso de uso ou conjunto de casos de uso), são realizadas estimativas de custo e prazo segundo a métrica proposta. Caso estas estimativas estejam em conformidade com o planejado, os casos de uso são liberados para elaboração e construção. Caso não estejam, são negociadas com os usuários e patrocinadores do projeto reduções nos requisitos com o objetivo de reduzir a complexidade e, conseqüentemente, os custos e prazos. O mesmo processo de negociação é utilizado nas fases de elaboração e construção se ocorrerem mudanças nos requisitos que alterem as variáveis técnicas propostas neste artigo e utilizadas na medição da complexidade dos casos de uso.

A fase de construção é caracterizada como uma fase estritamente técnica, com a produção do código fonte do sistema. É seguida da fase de testes, homologação, implantação e estabilização. Em qualquer destas fases, podem surgir mudanças nos requisitos que serão tratadas da mesma forma, ou seja, pode ser feita uma nova análise de complexidade dos casos de uso modificados e caso as alterações modifiquem os custos do projeto, estas alterações são negociadas, tanto com o cliente quanto com as empresas contratadas.

## 3. Pontos de Caso de Uso

A métrica de contagem de pontos de caso de uso foi inicialmente proposta por Gustav Karner [KARNER 1993], como uma adaptação da métrica de contagem de pontos de função [VAZQUEZ et al. 2003] voltada para sistemas orientados a objetos. Simplificando, o método de Karner consiste em contar a quantidade de atores e de transações dos casos de uso. Esta contagem é realizada através de um método conduzido em 6 passos descritos a seguir.

**Passo 1:** Os atores dos modelos de casos de uso são categorizados em simples, médio e complexo. Para cada tipo de ator, Karner definiu um peso (Tabela 2). A quantidade de cada tipo de ator é multiplicado pelo respectivo peso, e ao final os valores são somados. O valor final é o peso total de atores (TAP – Total Actor Points).

**Tabela 2. Pesos dos atores**

Tipo	Descrição	Peso
Simple	Um ator é considerado simples se ele representa outro sistema com uma API definida	1
Médio	Um ator é considerado médio se é uma interação com outro sistema através de um protocolo ou se é uma interação humana através de linha de comando.	2
Complexo	Um ator é considerado complexo se ele interage com o sistema através de uma interface gráfica.	3

**Passo 2:** Cada caso de uso é categorizado em simples, médio e complexo de acordo com seu número de transações (Tabela 3), incluindo as transações contidas nos fluxos alternativos dos casos de uso. Segundo Karner, uma transação é um evento que ocorre entre a requisição do usuário e a resposta do sistema. Para cada tipo é atribuído um peso e a quantidade de cada tipo de caso de uso é multiplicada pelo respectivo peso, e ao final os valores são somados. O valor final é o peso total dos casos de uso (TUCP – Total Use Case Points).

**Tabela 3. Peso dos casos de uso**

Tipo	Descrição	Peso
Simple	Até 3 transações	5
Médio	De 4 a 7 transações	10
Complexo	Mais que 7 transações	15

**Passo 3:** É calculado a quantidade de pontos de caso de uso não ajustados (UUCP – Unadjusted Use Case Points) através da soma:  $UUCP = TAP + TUCP$ .

**Tabela 4. Fatores técnicos que influenciam na complexidade**

Fator	Descrição	Peso
T1	Sistemas Distribuídos	2,0
T2	Tempo de resposta/performance	1,0
T3	Eficiência (on-line)	1,0
T4	Processamento interno complexo	1,0
T5	Código deve ser reutilizável	1,0
T6	Facilidade de instalação	0,5
T7	Usabilidade	0,5
T8	Portabilidade	2,0
T9	Facilidade de manutenção	1,0
T10	Acessos simultâneos (concorrência)	1,0
T11	Aspectos especiais de segurança	1,0
T12	Acesso direto para terceiros	1,0
T13	Facilidades especiais de treinamento	1,0

**Passo 4:** A quantidade de pontos de caso de uso obtida no passo 3 é ajustada pelos fatores técnicos e ambientais, os quais estão definidos nas Tabelas 4 e 5. Para cada fator é atribuído um valor entre 0 e 5 de acordo com sua influência no projeto. Atribuindo 0 para um fator, significa que ele é irrelevante para o projeto, enquanto 5 indica que é um fator essencial. Escalas intermediárias devem ser usadas para indicar a real relevância do fator para o projeto.

O peso dos fatores técnicos (TFactor – Technical Factor) é obtido através da multiplicação da pontuação atribuída para cada fator pelo seu respectivo peso (Tabela

4), somando-se então todos os valores ao final. Uma vez obtido o TFactor, calcula-se TCF (Technical Complexity Factor) através da fórmula:  $TCF = 0,6 + (0,01 * TFactor)$ .

**Tabela 5. Fatores ambientais que contribuem na eficiência**

Fator	Descrição	Peso
F1	Familiaridade com a Metodologia de Gestão e Desenvolvimento	1,5
F2	Experiência na Aplicação	0,5
F3	Expert na Técnica de Desenvolvimento	1,0
F4	Experiência do Gerente de Projeto	0,5
F5	Motivação	1,0
F6	Requisitos estáveis	2,0
F7	Trabalhadores part-time	-1,0
F8	Dificuldade da Linguagem de Programação	-1,0

O peso dos fatores ambientais (EFactor) é obtido através da multiplicação da pontuação atribuída para cada fator pelo seu respectivo peso (Tabela 5), somando-se então todos os valores ao final. Uma vez obtido o EFactor, calcula-se EF (Environmental Factor) através da fórmula:  $EF = 1,4 + (-0,03 * EFactor)$ .

**Passo 5:** O cálculo da quantidade de pontos de caso de uso ajustados é feito através da multiplicação dos pontos de casos de uso não ajustados pelo fator técnico e pelo fator ambiental, ou seja,  $UCP = UUCP * TCF * EF$ .

**Passo 6:** O esforço é calculado através da multiplicação da quantidade de pontos de caso de uso ajustados (UCP) pelo valor específico de Homem-Hora (HH) para cada ponto por caso de uso. Karner apresenta resultados que mostram que para cada UCP são necessários 20 horas-homem.

#### **4. Extensão do Processo de Estimativa**

A proposta deste artigo consiste em definir uma nova métrica para a classificação dos casos de uso segundo sua complexidade para ser aplicado em conjunto com o método proposto por Karner [KARNER 1993], substituindo a sua análise da complexidade dos casos de uso pela análise proposta neste artigo. Para isso foram definidas 5 variáveis técnicas que utilizam as seguintes informações: tipo do caso de uso, quantidade de requisitos de interface com o usuário, quantidade de regras de negócio, quantidade de conceitos associados, e quantidade de campos na interface com o usuário ou relatório. Para aplicar esta nova métrica da complexidade dos casos de uso é necessário que estas informações estejam definidas no processo de desenvolvimento de software.

Para exemplificar o processo de análise da complexidade dos casos de uso, utilizaremos o caso de uso “Petição Complementar” (UC17 da Tabela 10) do sistema de Gerenciamento de Processos Administrativos Eletrônicos (PROAD-TRT/SC) apresentado nas figuras 1, 2 e 3.

##### **4.1. Análise de Complexidade de Casos de Uso**

A seguir são apresentadas as variáveis técnicas utilizadas no processo de análise de complexidade dos casos de uso proposto neste artigo.

###### **4.1.1. Pontos por Tipo de Caso de Uso (UCTP – Use Case Type Point)**

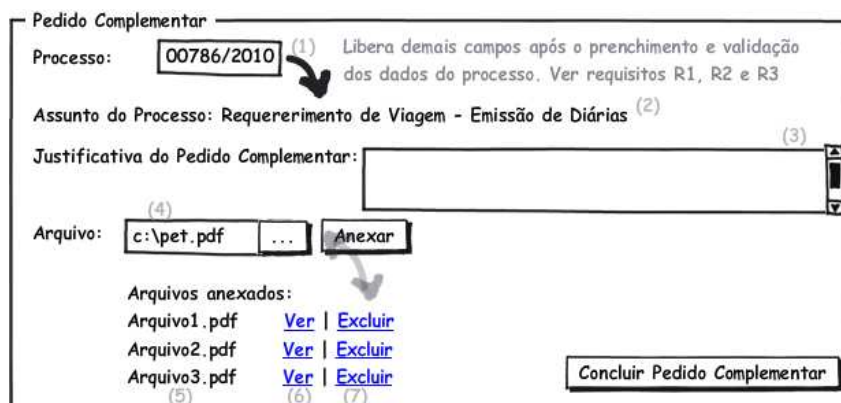
Com a utilização de frameworks e padronização das técnicas de desenvolvimento, os casos de uso podem ser classificados de acordo com seu tipo de funcionalidade, e para

cada tipo podemos atribuir um peso de complexidade de desenvolvimento. A Tabela 6 mostra alguns tipos de casos de uso utilizados pelo TRT/SC. É importante destacar que esta classificação e seus pesos são muito dependentes dos padrões de cada empresa e, portanto, devem ser adaptados para cada realidade, levando em consideração que o esforço para desenvolvimento de cada tipo é diretamente proporcional ao valor dado ao UCTP, ou seja, quanto menor o valor de UCTP, menor o esforço necessário para o desenvolvimento do caso de uso. Para a ponderação destes valores, deve-se levar em consideração se a empresa possui padrões de interface, ferramentas e frameworks que auxiliam no desenvolvimento destes tipos de casos de uso.

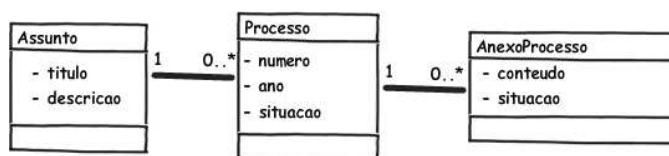
**Caso de Uso:** Petição Complementar  
**Nível de Detalhamento:** Baixo  
**Atores:** Usuário comum  
**Objetivo do caso de uso:** Permitir ao usuário entrar com Petição Complementar em processos que já estão em andamento.  
**Descrição:** O usuário informa o número e ano do processo em que deseja entrar com a petição complementar. Informa uma justificativa para ingresso da petição complementar e anexa os arquivos que deseja juntar no processo.  
**Requisitos do caso de uso:**

- R1. Controle dos campos da tela: O sistema deverá manter habilitado somente o campo número do processo, até que este seja validado (R3).
- R2. Formato e máscara do número do processo: O número e ano do processo devem ser informados no mesmo campo com a máscara NNNNN/AAAA, onde NNNNN representa o número do processo com 5 dígitos e AAAA representa o ano do processo com 4 dígitos. A máscara deve ser automática pelo sistema.
- R3. Carregar dados do processo: Ao informar o número do processo o sistema deve buscar as informações do processo e exibi-las para o usuário, liberando demais campos da tela. Se o processo não existir ou estiver na situação JUNTADO, o sistema deverá exibir mensagem informativa ao usuário, não liberando acesso aos demais campos da tela.
- R4. Deve-se permitir anexar um ou mais arquivos no pedido complementar, sendo que pelo menos 1 é obrigatório.
- R5. A justificativa do pedido complementar é obrigatória e deve ter no mínimo 30 caracteres.
- R6. Ao concluir o pedido complementar, o sistema deve gerar um arquivo com o conteúdo descrito na justificativa e anexá-lo automaticamente. O pedido e seus arquivos devem ficar na situação PENDENTE DE RECEBIMENTO, e o processo que recebeu o pedido complementar deve ficar na situação PEDIDO COMPLEMENTAR PENDENTE.

**Figura 1. Descrição do Caso de Uso Pedido Complementar**



**Figura 2. Protótipo de Tela do Caso de Uso Pedido Complementar**



**Figura 3. Diagrama de Classes de Domínio correspondentes ao Caso de Uso Petição Complementar**

A análise de um protótipo de tela pode auxiliar no enquadramento em um dos tipos de caso de uso. No entanto esta tarefa depende da experiência do analista. A classificação do caso de uso em um tipo permite a obtenção da variável técnica UCTP.

**Tabela 6. Tipos de Caso de Uso utilizados no TRT/SC**

Tipo	Descrição	UCTP
Consulta de Entidade	Casos de uso de consulta a entidades, onde podem ser informados parâmetros de pesquisa e o sistema efetua a pesquisa e exibe os resultados.	1
CRUD Tabular Entidade	Casos de uso que possibilitam as operações de CRUD (Create, Retrieve, Update, Delete) de vários registros ao mesmo tempo utilizando formato "Tabela", ou seja, cada linha corresponde a um registro. Este tipo de caso de uso não fornece opções de pesquisa e sempre mostra todos os dados da tabela. Normalmente são utilizados para manter tabelas de suporte do sistema, como por exemplo Tipos, Municípios, etc.	1
CRUD Entidade Mestre	Ao contrário do CRUD Tabular, este tipo de casos de uso permite opções de pesquisa a objetos e as operações de CRUD são realizadas com base em uma única entidade (mestre).	2
CRUD Entidade Mestre/Detalhe	Casos de uso que possibilitam em uma única interface manter dados de uma entidade principal (mestre) e suas associações (detalhes), podendo os detalhes ter cardinalidade superior a 1. Este tipo de interface pode suportar a manutenção de mais de uma associação (detalhe).	3
Relatório	Casos de uso de geração de relatórios com base na seleção de objetos.	3
Manipulação de Documentos	Casos de uso que manipulam arquivos/documentos em formatos DOC, RTF, ODT, PDF, XML dentre outros tipos.	5
Componentes de Serviços	Componentes de serviços utilizados para integração entre sistemas ou reaproveitamento por outros sistemas. Exemplos destes tipos de componentes são: Webservices, EJBs, serviços REST, etc.	5

A partir da análise do protótipo da figura 2, pode-se classificar o caso de uso no tipo "CRUD Entidade Mestre/Detalhe", pois temos uma tela com informações de dois conceitos: Processo e AnexoProcesso. Assim, é atribuído o valor 3 à variável UCTP do exemplo.

#### **4.1.2. Pontos por Requisito de Interface com o Usuário (UIRP – User Interface Requirement Point)**

Os requisitos de interface são todos os requisitos que facilitam a usabilidade. A pontuação em função dos requisitos de interface com o usuário é realizada através da multiplicação do número de requisitos de interface com o usuário (NUIR) pelo peso deste tipo de requisito (UIRW – peso 2). Este fator também pode variar para cada empresa em virtude da experiência das equipes de desenvolvimento e dos frameworks utilizados.

Considerando o exemplo, podemos categorizar como requisitos de interface com o usuário os requisitos de números 1, 2, 3, 4 e 5, pois tratam de facilidades da interface e validações estendidas de campos. Como  $UIRP = UIRW * NUIR$ , para o exemplo em questão temos  $UIRP = 2 * 5$ , logo, UIRP é igual a 10.

#### **4.1.3. Pontos por Regras de Negócio (BRP – Business Rules Point)**

Uma regra de negócio é um tipo de requisito não funcional constante em um caso de uso, como por exemplo: fórmulas de cálculo, regras de atualização e inclusão de dados, etc. A pontuação em função das regras de negócio é realizada através da multiplicação do número de regras de negócio (NBR) pelo peso deste tipo de requisito (BRRW - peso 3). Este fator pode variar para cada empresa em virtude da arquitetura dos frameworks utilizados.

Considerando nosso exemplo, pode-se categorizar como regra de negócio o requisito de número 6, logo temos NBR igual a 1 e BRRW igual a 3. Como  $BRP = BRRW * NBR$ , temos que  $BRP = 3 * 1 = 3$ .

#### 4.1.4. Pontos por Classes de Domínio (DCP – Domain Classes Point)

Devem ser contados quantos conceitos estão envolvidos no caso de uso (NDC – Number of Domain Classes), excluindo-se desta contagem as agregações e composições cujos conceitos são fracos em relação ao domínio do caso de uso. Conceitos são as classes relacionadas ao negócio e utilizadas no caso de uso. A obtenção deste valor é realizada através da análise do diagrama de classes de domínio que o caso de uso trata. A sugestão é que seja feito um diagrama de classes de domínio para cada caso de uso, facilitando assim esta avaliação. Com base no número de conceitos obtidos (NDC), pode-se obter o valor de DCP consultando a Tabela 7.

**Tabela 7. Pontos em Função da Quantidade de Conceitos do Caso de Uso**

Quantidade de Conceitos Associados	DCP
$NDC \leq 5$	1
$5 < NDC < 10$	2
$NDC \geq 10$	3

Considerando o exemplo, o diagrama de classes relacionado com o caso de uso em questão apresenta os conceitos Processo e AnexoProcesso. De acordo com a Tabela 7, como  $NDC=2$ , o DCP é igual a 1. É importante observar que neste exemplo a classe Assunto não é contabilizada no NDC pois trata-se de uma agregação e portanto pode ser considerada apenas como atributo da classe Processo.

#### 4.1.5. Pontos por Quantidade de Campos na Interface com o Usuário ou Relatório (UIFP – User Interface Fields Point)

Deve ser contado o número de campos e parâmetros que constam na tela ou relatório (NUIF – Number of User Interface Fields), incluindo-se as listas e suas colunas. Botões não devem ser contabilizados pois já são considerados nas variáveis UIRP e/ou BRP, quando associados a alguma regra de negócio ou requisito de interface. Com base no número de campos contabilizados (NUIF), pode-se obter o valor de UIFP consultado a Tabela 8. Analisando o protótipo do exemplo (figura 2), podemos observar a presença de 7 campos ( $NUIF=7$ ), os quais estão destacados no protótipo através de numeração entre parênteses, como por exemplo o campo processo demarcado com (1). Com este número, podemos consultar a Tabela 8 e obter o número de pontos correspondente. Para nosso exemplo obtemos UIFP igual a 1.

**Tabela 8. Pontos em Função da Quantidade de Campos na Interface com o Usuário**

Qtde de Campos na Interface ou Relatório	UIFP
$NUIF \leq 10$	1
$10 < NUIF \leq 20$	2
$NUIF > 20$	3

## 4.2. O Cálculo da Complexidade do Caso de Uso

A complexidade do caso de uso é obtida através da análise do peso final do caso de uso. O peso final (FW – Final Weigth) é obtido através do somatório de todas as variáveis técnicas:  $FW = UCTP + UIRP + BRP + DCP + UIFP$ . Se FW for menor ou igual a 7, a



complexidade do caso de uso é classificada como Simples. Caso FW fique entre 7 e 20, a complexidade do caso de uso é classificada como Média. Para valores de FW maiores ou iguais a 20, a complexidade do caso de uso é classificada como Complexa.

**Tabela 9. Classificação da Complexidade em Função do Peso Final (FW)**

Peso Final	Complexidade
$FW \leq 7$	Simples
$7 < FW < 20$	Médio
$FW \geq 20$	Complexo

No exemplo, com base no valor obtido para FW, consultamos a Tabela 9 e finalmente obtemos a classificação do caso de uso. Como  $FW = UCTP + UIRP + BRP + DCP + UIFP$ , no exemplo temos  $FW = 3 + 10 + 3 + 1 + 1 = 18$ . Logo, o caso de uso do exemplo é classificado como Médio.

A definição da fórmula e a configuração das Tabelas 6, 7, 8 e 9 são resultantes da análise do histórico de desenvolvimento dos casos de uso no TRT/SC e foram validadas através de análise de regressão múltipla [LARSON E FARBER 2004] baseada em dados coletados de 8 sistemas desenvolvidos no TRT/SC em 2008 e 2009. Para cada caso de uso foram analisadas as variáveis técnicas propostas, aplicado o cálculo do peso final e registrado a quantidade de esforço (horas trabalhadas) para cada caso de uso.

### 4.3. Estimativa do Projeto

Uma vez obtida a complexidade dos casos de uso, procede-se o cálculo para contagem total de pontos de caso de uso, conforme a técnica de Gustav Karner [KARNER 1993], substituindo a forma de classificação de complexidade dos casos de uso pela proposta neste artigo.

## 5. Estudo de Caso

Na Tabela 10, é mostrado um exemplo de estimativa do sistema de Gerenciamento de Processos Administrativos Eletrônicos (PROAD-TRT/SC) desenvolvido pelo TRT/SC com terceirização das fases de elaboração e construção. Este foi um dos projetos, dentre outros no TRT/SC, em que a métrica aqui proposta foi aplicada com sucesso. Nesta tabela podemos observar a classificação de complexidade dos casos de uso em simples, médio e complexo em função da análise das variáveis técnicas UCTP, UIRP, BRP, DCP e UIFP, conforme proposto neste artigo.

Após a classificação de complexidade dos casos de uso, o analista aplica o cálculo proposto por Karner para obter a estimativa de esforço em UCP (Use Case Points).

Levando-se em consideração a existência de 3 atores complexos, 17 casos de uso simples, 7 casos de uso médios, 5 casos de uso complexos e aplicando a técnica proposta por Karner, temos UUCP igual a 239. Considerando-se TCF igual a 1,12 e EF igual a 0,79, temos 200,76 pontos de caso de uso ajustados (UCP), arredondados para 201 UCP.

Com este resultado podemos obter as estimativas de custo e esforço. O esforço é obtido através da multiplicação do UCP obtido pela quantidade de horas-homem (HH) necessárias para desenvolvimento de 1 UCP. Como dito anteriormente, Karner produziu estatísticas que levam ao valor de 20 HH por ponto de caso de uso. No entanto, com o

uso de frameworks e padrões institucionais, este valor tende a diminuir significativamente. No TRT/SC, este valor aproxima-se de 15HH por cada ponto de caso de uso. Se forem consideradas somente as fases de elaboração e construção, as quais são terceirizadas no TRT/SC, este valor reduz para aproximadamente 10HH. Com a obtenção do esforço do projeto, pode-se obter o custo através da multiplicação da quantidade total de horas-homem pelo valor de cada hora trabalhada. Para a formação do custo total ainda deve-se levar em consideração outros custos, como treinamentos, viagens, etc.

**Tabela 10. Complexidade dos Casos de Uso do Estudo de Caso**

Caso de Uso	UCTP	UIRP	BRP	DCP	UIFP	FW	Complexidade
UC01	3	3	4	2	6	23	Médio
UC03	5	2	1	2	11	15	Médio
UC04	1	4	4	6	6	24	Complexo
UC05	1	2	1	2	12	11	Médio
UC17	3	5	1	2	7	18	Médio
UC19a	1			1	5	3	Simples
UC19b	1			3	13	4	Simples
UC21a	1	2		1	1	7	Simples
UC21b	3	3	4	3	19	24	Complexo
UC22a	1			1	6	3	Simples
UC22b	3		3	5	16	15	Médio
UC24a	1			3	6	3	Simples
UC24b	5	4	3	3	14	25	Complexo
UC35	1			3	11	4	Simples
UC36	3			2	5	5	Simples
UC37	1			2	13	4	Simples
UC38a	1		1	2	6	6	Simples
UC38b	3	4	4	2	9	25	Complexo
UC40	2	3	3		10	19	Médio
UC41	3			2	3	5	Simples
UC44a	1			3	6	3	Simples
UC44b	5	5	3	3	14	27	Complexo
UC45a	1		1	3	7	6	Simples
UC45b	3	1	2	2	9	13	Médio
UC47	3			2	5	5	Simples
UC48a	1			1	6	3	Simples
UC48b	3			2	13	6	Simples
UC49a	1	1		1	6	5	Simples
UC49b	3			2	11	6	Simples

## 6. Conclusão

Este artigo apresentou uma nova métrica de estimativas que define uma forma objetiva para classificação de complexidade dos casos de uso para ser utilizada em conjunto com a aplicação da métrica de Karner para cálculo da quantidade de pontos de caso de uso. A análise de complexidade dos casos de uso é realizada com base em alguns artefatos (Tabela 1), os quais são analisados e através deles são ponderadas 5 variáveis técnicas que definem a complexidade de cada caso de uso.

O processo de estimativa de software é fundamental para o planejamento de projetos de software e também pode ser aplicado como ferramenta de negociação de escopo do projeto e mudanças de requisitos.

Na proposta original de Karner [KARNER 1993], a complexidade de um caso de uso é medida considerando o número de transações do caso de uso. No entanto, para a aplicação desta métrica é necessário que os casos de uso estejam expandidos. Outro fator limitante é a dependência da forma com que os casos de uso são escritos. Pelo fato da análise de complexidade ser baseada no texto dos casos de uso, este processo torna-se subjetivo e sujeito a entendimentos divergentes entre pessoas diferentes, principalmente entre equipes técnicas, clientes e fornecedores de serviços de TI.

Alguns trabalhos apresentam extensões propondo melhorias no método original de contagem de pontos de caso de uso. Entretanto, estas extensões consideram a utilização de casos de uso expandidos. Em [BRAZ E VERGILIO 2006] é definida uma métrica para análise de complexidade e esforço considerando as estruturas e seções dos casos de uso através da contagem e do peso dos cenários, atores, pré-condições e pós-condições. Em [VIEIRA 2007] é definida uma métrica para análise de complexidade dos casos de uso baseada na contagem do número de passos denominados obrigatórios. Nesta métrica o autor defende que a quantidade de passos obrigatórios influencia diretamente na complexidade do caso de uso e que os passos complementares não têm influência sobre o esforço de desenvolvimento.

A diferença da métrica proposta neste artigo para outras métricas, como as apresentadas em [VIEIRA 2007] e [BRAZ E VERGILIO 2006], está no fato da análise de complexidade dos casos de uso ser realizada de forma objetiva e sem a necessidade de ter os casos de uso expandidos. Esta abordagem minimiza conflitos e divergências nas classificações de casos de uso por pessoas diferentes.

A aplicação no TRT/SC da proposta apresentada neste artigo mostrou que ela tem sido bastante precisa tanto na estimativa de projetos quanto na negociação de mudanças nos requisitos entre equipes de desenvolvimento e usuários e entre equipes de desenvolvimento e fábricas de software. As tabelas utilizadas para o cálculo do peso final de cada caso de uso e a métrica para contagem dos pontos de caso de uso, fazem parte da metodologia do TRT/SC e são definidas nos contratos com as empresas terceirizadas. As eventuais divergências entre equipes de desenvolvimento e empresas terceirizadas são resolvidas através de reuniões onde cada uma das partes apresenta o que foi considerado na sua contagem e a justificativa de suas considerações.

A proposta apresentada neste trabalho também pode ser aplicada em contratos de terceirização de desenvolvimento de software, estabelecendo assim uma ferramenta de negociação de requisitos e custos entre clientes e fornecedores. Para tanto, é necessário fazer a adaptação dos pesos dados nas Tabelas 6, 7, 8 e 9, em função das características de cada processo e da empresa onde será aplicada. Os pesos contidos nestas tabelas são muito dependentes do processo de desenvolvimento de cada empresa, principalmente em função dos padrões estabelecidos e frameworks utilizados. Outro ponto possível de adaptação é a substituição de protótipos de tela por técnicas que representam a interação entre o usuário e o sistema, como os UIs (Diagramas de Interação do Usuário) [VILAIN et al. 2000].

Como trabalhos futuros, pretende-se generalizar a classificação do tipo de caso de uso para que ele fique mais independente do framework utilizado e do projeto de interface definido com o usuário. Outro ponto a ser melhorado é possibilitar a ponderação do reaproveitamento de código na estimativa de cada caso de uso.

## 7. References

- ALBRECHT, A. J.. (1979) Measuring Application Development Productivity. Proceedings of the IBM SHARE/GUIDE Applications Development Symposium. Monterey, CA. October.
- ALVES, Angela Maria. (2002) Tese de Mestrado “Contratação de Produtos e Serviços de Software”. Unicamp. Novembro.
- AMBLER, S.. (2004) Modelagem Ágil. Porto Alegre. Bookman.
- ANDA, B.; DREIEM D.; SJØBERG, D.I.K.; JØRGENSEN, M.. (2001) Estimating software development effort based on use cases – Experiences from industry. In M. Gogolla, C. Kobryn (Eds.): UML 2001 - The Unified Modeling Language. Modeling Languages, Concepts, and Tools, 4th Int’l Conference. Springer-Verlag LNCS 2185, (2001), 487-502.
- BRASIL. (1993) Lei 8666/1993, de 21 de junho de 1993. Regulamenta o art. 37, inciso XXI, da Constituição Federal, institui normas para licitações e contratos da Administração Pública e dá outras providências. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/Leis/L8666cons.htm](http://www.planalto.gov.br/ccivil_03/Leis/L8666cons.htm). Acesso em: 26/02/2010.
- BRASIL. (2006) Tribunal de Contas da União. Acórdão 786/2006. Dispõe sobre a terceirização de serviços de TI nas empresas públicas. Publicado no Diário Oficial da União de 26/05/2006.
- COHN, M.. (2005) Agile Estimating and Planning. Prentice Hall. November.
- KARNER, G.. (1993) Metrics for Objectory. thesis at the university of Linkopmg, Sweden. LiTH-IDA-Ex-9344; 21. December.
- KARNER, G.. (1993b) Resource Estimation for Objectory Projects. Objectory Systems. September.
- LARSON, R; FARBER E.. (2004) Estatística Aplicada. 2ª Edição. Prentice Hall.
- MARTINS, S. P.. (2009) A terceirização e o Direito do Trabalho. 9º edição. Atlas.
- ØSTVOLD, K.M.; HAUGEN, N.C.; BENESTAD, H.C.. (2008) Using planning poker for combining expert estimates in software projects. Journal of Systems and Software. Volume 1. December.
- VAZQUEZ, C.; SIMÕES, G.S. e ALBERT, R.M.. (2003) Análise de Pontos de Função: medição, estimativas e gerenciamento de projetos de software. 1ª Edição. Érica.
- VIEIRA, E. L.. (2007) Uso do Conceito de Passos Obrigatórios Para Aprimorar o Processo de Contagem do Método de Pontos de Caso de Uso. Tese da Universidade Federal de Santa Catarina. Junho.
- VIEIRA, E. L.; WAZLAWICK, R. S.. (2006) Teoria Explanatória para Estimativa Baseada em Casos de Uso no Desenvolvimento Orientado a Objetos. In: XXXII CLEI - Conferência Latinoamericana de Informática, Santiago, Chile.
- VILAIN, P., AND SCHWABE, D., AND SOUZA, C.S. de.. (2000) A Diagrammatic Tool for Representing User Interaction in UML. In Proceedings of the UML2000 Conference, York, England, October 2000, Lecture Notes in Computer Science, Springer, Berlin, 133-147.