



**ORLANDO DE ARAUJO**

**DEFINIÇÃO DE UMA METODOLOGIA PARA  
AUDITORIA DE PREÇOS EM CONTRATOS DE  
PROJETOS DE SOFTWARE BASEADA NA  
ANÁLISE DE PONTOS DE FUNÇÃO**

**Rio de Janeiro**

**Julho 2008**

Orlando de Araujo

Definição de uma Metodologia  
para Auditoria de Preços em Contratos de  
Projetos de Software baseada na Análise de  
Pontos de Função

Monografia apresentada à Universidade  
Estácio de Sá, como requisito final do curso  
de pós-graduação Lato Senso em Auditoria  
de Tecnologia da Informação.

Orientador: Prof<sup>ra</sup>. Claudia Hazan

Rio de Janeiro  
Julho 2008

## Sumário

	<b>Pág.</b>
<b>Introdução</b> .....	6
<b>Capítulo 1</b>	
Validação da estimativa de preços .....	10
<b>Capítulo 2</b>	
Análise da compatibilidade dos preços finais .....	25
<b>Capítulo 3</b>	
Uso da APF em procedimentos de auditoria.....	32
<b>Conclusão</b> .....	42
<b>Referências</b> .....	44
<b>Anexos</b> .....	46

## Resumo

Freqüentemente, o auditor governamental é chamado a emitir parecer sobre a compatibilidade dos preços praticados por entes públicos na contratação de serviços de *software*, o que ocorre em geral em dois momentos: no procedimento licitatório, com base apenas nas especificações do projeto básico, e na fase de execução, a partir da análise dos produtos efetivamente entregues. A métrica de tamanho funcional de projetos de *software* – Pontos de Função, criada por Allan J. Albrecht, tem sido mundialmente utilizada na indústria de *software*, especialmente por entes governamentais brasileiros, no estabelecimento de contratos de prestação de serviços de desenvolvimento de *software*. De fato, a Análise de Pontos de Função (APF) tem apoiado organizações e departamentos de desenvolvimento de *software* na análise de qualidade e produtividade e, especialmente, na geração da estimativa de tamanho dos projetos. É importante destacar que o modelo de qualidade de *software* *Capability Maturity Model* (CMMI) recomenda que a derivação das estimativas de prazo, esforço e custo sejam baseadas na estimativa de tamanho do projeto. Outro benefício a ser destacado é que a APF, sendo baseada na visão do usuário e independente da tecnologia de implementação utilizada, facilita estimativas de projetos de *software* nas fases iniciais do ciclo de vida do projeto, ou seja, considerando as especificações dos projetos básicos das licitações. E, ainda, a Contagem de Pontos de Função é baseada em uma metodologia sólida, publicada no manual de práticas de contagem (CPM) pelo *International Function Point Users Group* (IFPUG). Além disso, o Brasil é um dos países com maior número de profissionais com certificação de especialistas em Pontos de Função - CFPS. Assim, a APF constitui uma valiosa ferramenta para apoiar um auditor na elaboração de pareceres sobre os custos de projetos de *software* com bases teóricas sólidas e consistentes, permitindo-lhe validar as estimativas e medições realizadas pelos contratantes, além de facilitar também a comunicação, durante todas as etapas do contrato, entre os órgãos de controle, os gestores públicos e as empresas desenvolvedoras de *software*. Este trabalho tem como propósito mostrar como a técnica de APF pode ser utilizada nos procedimentos de auditoria de conformidade, de modo a se obter uma avaliação correta da adequação dos preços dos serviços de desenvolvimento de *software* estimados ou medidos, a partir de apenas duas variáveis básicas: o preço unitário do Ponto de Função, estimado por pesquisa de mercado ou estabelecido pela competição entre os licitantes, e o número de Pontos de Função correspondentes aos serviços, obtidos como resultado do processo de estimativa ou medição.

## **Abstract**

Often, the government auditor is required to issue an opinion on the compatibility of prices charged by public entities when contracting software services, which generally occurs in two stages: in the tender process, based only on basic specifications of the project, and the implementation phase, from the analysis of the product actually delivered. The functional size metric of software projects – Function Points, created by Allan J. Albrecht, has been used worldwide in the software industry, especially by Brazilian government entities, in the establishment of a service contract for software development. In fact, the Function Point Analysis (FPA) has supported organizations and departments of software development in the analysis of quality and productivity and, especially, the generation of the estimated size of the projects. It is important to emphasize that the model of quality of software Capability Maturity Model (CMMI) recommends that the derivation of the estimations of time, effort and cost are based on the estimated size of the project. Another benefit to be highlighted is that FPA is based on the user's view and independent of the technology used, ease the estimations of software projects in the early stages of its life cycle, i.e., considering the specifications of the projects bidding the contract. Moreover, the FPA is based on sound methodology, published in the Counting Practices Manual (CPM) by International Function Point Users Group (IFPUG). Brazil is one of the countries with highest number of professionals with certification of specialists in function points - CFPS. Thus, the FPA is a valuable tool to support the auditor on drafting opinions about the costs of software projects with solid theoretical foundations and consistent, allowing it to validate the estimates and measurements performed by contractors, and also facilitate the communication, during all stages of the contract, between the control entities, public managers and developing companies. This work intends to show how the of FPA technique can be used in the procedures of audit of compliance, in order to obtain a correct assessment of the adequacy of prices for software development estimated or measured, from only two basic variables: the unit price of the function point, estimated by market research or established by competition among the bidders, and the number of function points corresponding to the services, obtained as a result of the estimate or measurement.

## Lista de Figuras

	Pág.
Figura 1 – Visão Geral dos Tipos Funcionais da Análise de Pontos por Função	13
Figura 2 – Insumos típicos para estimativas de projetos	17
Figura 3 – Parâmetros de produtividade	23
Figura 4 – O processo de contagem da APF	27
Figura 5 – Procedimento de Contagem de Pontos de Função	A-1

## INTRODUÇÃO

A variedade de Sistemas de Informação passíveis de especificação para fins de desenvolvimento por empresas contratadas é virtualmente ilimitada, mesmo restringindo-se o universo apenas aos órgãos e entidades públicos. Tomando apenas alguns exemplos recentes na esfera federal, verifica-se que o Ministério do Esporte contratou o desenvolvimento de um sistema de controle de acesso de pessoas e veículos às instalações dos Jogos Pan-americanos de 2007, baseado no uso de crachás com tecnologia de rádio-frequência, ao preço de R\$ 9.000.000,00. Para o mesmo evento, o Ministério da Justiça realizou contratações sigilosas abrangendo sistemas para a área de segurança dos Jogos, com orçamento superior a R\$ 160.000.000,00<sup>1</sup>.

Não são raras contratações desse porte pela administração pública federal, o que exige do profissional de controle o conhecimento de técnicas que permitam aferir se os preços cobrados estão compatíveis com o produto efetivamente entregue.

Nesse contexto, o profissional de auditoria governamental se vê frequentemente diante da necessidade de analisar indícios de sobrepreço ou superfaturamento em compras relacionadas à Tecnologia da Informação (TI), algumas vezes a partir de denúncias encaminhadas aos órgãos de controle, ou em apoio a processos judiciais, que exigem pareceres exatos e tempestivos, atributos que nem sempre são conciliáveis. As demandas são variadas, retratando a crescente complexidade dessa área de conhecimento.

Deseja-se avaliar, por exemplo, se estão compatíveis com o mercado os salários atribuídos à mão-de-obra especializada em processos de terceirização, como analistas de suporte de redes Microsoft, administradores de banco de dados Oracle ou desenvolvedores em determinadas linguagens de programação, como Java, cuja remuneração pode variar em função de diversos fatores, como certificações profissionais exigidas, tempo de experiência, local de prestação dos serviços e jornada a ser cumprida.

Também se questiona se foram corretamente estimados pelo gestor público, previamente à compra, os custos com a aquisição de licenças de *software* e equipamentos importados, como grandes servidores de aplicações e componentes de rede ou segurança, muitas vezes com um único representante no mercado, exigindo análise minuciosa das necessidades do comprador e das especificações dos produtos e serviços a serem adquiridos.

---

<sup>1</sup> Contratações mencionadas no Acórdão TCU 876/2007-Plenário, disponível em [www.tcu.gov.br](http://www.tcu.gov.br)

Apesar das dificuldades inerentes a cada um desses problemas, entretanto, uma demanda específica vem assumindo proporções desafiadoras: a verificação da compatibilidade de preços nas licitações e contratos para desenvolvimento de *software*. Os órgãos de controle esperam que o profissional de TI seja capaz de, a partir das especificações do sistema, emitir, parecer conclusivo sobre o preço cobrado, com grau de precisão comparável aos laudos na área de obras civis, por exemplo<sup>2</sup>.

Embora alguns estudos baseados na técnica Análise de Pontos de Função (APF), publicados por instituições como *The International Software Benchmarking Standards Group* (ISBSG)<sup>3</sup> e *Software Productivity Research* (SPR)<sup>4</sup>, venham apresentando parâmetros para esse tipo de avaliação, eles ainda são desconhecidos para muitos auditores. Assim, recomenda-se maior disseminação da técnica de APF, e de suas aplicações na análise de qualidade, produtividade, prazo, esforço e custos, junto a esses profissionais, bem como para os gestores públicos, que nem sempre dispõem do conhecimento necessário para fazer uma estimativa aceitável dos valores a despender nas compras, obrigatória por determinação expressa da chamada Lei das Licitações – Lei 8.666/93.

A análise de preços em uma contratação de grande porte pode ser bastante trabalhosa, especialmente tendo em vista a singularidade dos requisitos funcionais e não funcionais das aplicações, como, entre outros, tecnologia, complexidade, segurança e criticidade em relação a um prazo de entrega imposto pelo cliente. Muitas vezes, tais requisitos são utilizados como pretexto para cobrança de preços bem acima do mercado.

Em qualquer situação, entretanto, mesmo nos casos mais complexos, é preciso que se busquem parâmetros para a elaboração de um eventual parecer sobre o preço do *software* a ser contratado, de modo a evitar o desperdício dos escassos recursos públicos. Segundo a literatura especializada, essa tarefa pode ser facilitada pelo uso da APF, o que levou à questão central deste trabalho: como a APF pode auxiliar o auditor governamental de licitações e contratos na validação das estimativas, medições e avaliações de preços em desenvolvimento de *software*?

Para responder a essa questão, foram delineados três objetivos, relacionados ao uso da APF em auditorias de licitações e contratos na área de TI.

---

<sup>2</sup> A Resolução 361/1991 do CONFEA considera de apenas 15% o erro máximo da estimativa de preços do projeto básico, o que é bem abaixo do erro das estimativas e avaliações de *software*, conforme verificado no trabalho.

<sup>3</sup> <http://www.isbsg.org>

<sup>4</sup> <http://www.spr.com/>

O primeiro deles visou descrever os benefícios e limitações do uso das técnicas de estimativa de PF para a validação das estimativas de tamanho, realizadas sobre as especificações de um sistema a ser desenvolvido, de modo a permitir uma análise sobre o projeto básico, ainda na fase pré-contratual, quando há, em geral, apenas registros iniciais e pouco precisos das demandas que o novo sistema deverá atender.

Em seguida, pretendeu-se descrever as modalidades de uso da APF para a análise da compatibilidade dos preços finais dos sistemas entregues, de modo a permitir uma análise sobre o sistema efetivamente gerado após a contratação, normalmente diferente das especificações iniciais.

Por fim, foi mostrado como a APF pode ser aplicada nos procedimentos de auditoria, facilitando a comunicação entre os órgãos de controle, os gestores públicos e as empresas desenvolvedoras de *software*, de modo a se obter uma avaliação correta da adequação dos preços praticados.

Em resumo, procurou-se responder às seguintes questões: quais são os benefícios e limitações do uso da APF para a validação das estimativas de tamanho do projeto básico? Quais são as modalidades de uso da APF para a análise da compatibilidade dos preços finais dos sistemas entregues? Como a APF pode ser aplicada nos procedimentos de auditoria, facilitando a comunicação entre os órgãos de controle, os gestores públicos e as empresas desenvolvedoras de *software*, de modo a se obter uma avaliação correta da adequação dos preços praticados?

O principal conceito teórico utilizado neste trabalho é a APF, técnica publicada em 1.979, por Allan J. Albrecht, com o objetivo de proporcionar a medição do tamanho dos projetos de *software*, bem como a análise de produtividade das equipes de desenvolvimento, constituindo uma ferramenta importante e bastante utilizada no mercado brasileiro, especialmente nas organizações governamentais.

É de se notar, entretanto, que o uso dessa técnica neste trabalho teve por objetivo não a determinação do tamanho do *software*, mas a validação do preço estimado ou final da contratação, usando como principal referência a obra *Estimating Software Costs - Bringing Realism to Estimating*, publicada por Capers Jones, que se encontra na segunda edição, de 2007, tratando de inúmeras técnicas para a finalidade a que se propõe, incluindo a APF.

A utilização de uma técnica consagrada, como a APF, tende a facilitar as tarefas de auditoria anteriormente mencionadas, que consistem em estimativa prévia à contratação, ou avaliação *ex post* sobre o produto gerado, resultando em pareceres com bases teóricas

sólidas e consistentes, aptos a servirem como instrumento para argumentação convincente em procedimentos administrativos ou judiciais.

Este trabalho consiste de pesquisa documental e bibliográfica aplicada, de natureza qualitativa, fenomenológica, com o uso predominante do método indutivo. A pesquisa documental abrange editais de licitação, contratos e relatórios de acompanhamento, relativos à contratação de serviços de desenvolvimento de sistemas pelos órgãos e entidades públicos.

A organização deste trabalho encontra-se conforme a seguir.

O Capítulo 1 descreve atividades relacionadas à validação das estimativas do projeto básico, abordando as principais métricas de tamanho de *software*, destacando a métrica Ponto de Função - PF. É apresentada uma visão geral da APF, considerando situações que podem limitar o uso da contagem detalhada, obrigando ao uso de técnicas de estimativa de tamanho, obtidas a partir de outras técnicas específicas, tais como Contagem Indicativa, Contagem Estimativa e *backfiring*.

O Capítulo 2 trata da avaliação *ex post* dos sistemas entregues, mencionando causas da variação dos sistemas quanto ao tamanho, esforço, prazo e preço, em relação ao projeto básico, e descrevendo a contagem de Pontos de Função propriamente dita e suas restrições, as modalidades de uso para avaliação do preço final, correspondentes aos tipos de contagem aplicação, desenvolvimento e manutenção evolutiva (*enhancement*), bem como a obtenção dos pontos de função ajustados.

Por fim, o Capítulo 3 aborda o uso da APF em procedimentos de auditoria, descrevendo a relação entre as atividades de controles interno e externo das licitações e os contratos realizados pelo setor público. São relacionadas algumas dificuldades de comunicação entre fornecedores, contratantes e órgãos de controle, questões legais e jurisprudenciais atinentes às contratações públicas de sistemas. Também, são apresentadas algumas cláusulas recomendadas nas contratações tendo como base a métrica PF e procedimentos de avaliação da conformidade dos preços praticados, sobrepreço e superfaturamento, a partir da contagem de pontos de função dos sistemas entregues.

## Capítulo 1

### VALIDAÇÃO DA ESTIMATIVA DE PREÇOS

Diante da identificação de necessidades relacionadas ao desenvolvimento de *software*, a primeira etapa para atendimento da demanda consiste usualmente em estimativas preliminares, de modo a se conhecer melhor os resultados que se deseja alcançar e a metodologia a ser utilizada para solução do problema. Em especial, é necessário conhecer os custos estimados para o projeto, o que precisa estar consignado de forma bastante clara, de modo a facilitar a aprovação dos gastos pela administração, bem como o controle de atributos do processo, como legalidade, economicidade e impessoalidade, o que é especialmente relevante na gestão de recursos públicos.

#### Métricas de Tamanho de Software

Embora o tamanho de um imóvel não permita, sem outros parâmetros, o cálculo do custo de sua construção, essa informação é uma das primeiras a ser obtida, quando se deseja avaliar quanto se despenderá nessa empreitada, o que costuma ser feito com bastante precisão pelos profissionais da área de engenharia civil.

A exatidão das estimativas para construção de um imóvel ainda está bem à frente do que se tem disponível em se tratando de custo de *software*, mas a qualidade das projeções relativas ao desenvolvimento tem melhorado nos últimos anos.

Como observa Jones (2007), os projetos de *software* adquiriram a reputação de ultrapassar custos e prazos, mais que qualquer outro item dentro das operações dos negócios, o que motivou inúmeros trabalhos acadêmicos e profissionais, na tentativa de prever, com razoável exatidão, quanto pagar por uma solução de *software*, tarefa que assume maior relevância à medida que os sistemas de informação crescem em tamanho e importância estratégica dentro das organizações.

Mesmo no momento atual, após significativos esforços para enquadramento da indústria do *software* dentro de padrões mais previsíveis, o cenário ainda parece mal resolvido. Conforme o ISBSG (2008), analisando projetos de sua base de dados, abrangendo experiências de vários países, o preço que o governo e as empresas pagam por projetos de *software* comparáveis entre si varia em mais de 10 vezes, havendo ainda, em média, estouro das estimativas de prazo em cerca de 37% dos projetos e, de custo, em 22%.

Percebe-se, portanto, a importância de melhoria no processo de estimativas envolvendo projetos de *software*. As estimativas inadequadas podem levar a prazos não cumpridos, custos excessivos com horas extras e má qualidade do *software*, conforme descrito por Hazan (2005).

Considera-se que a primeira informação a ser obtida para estimativa de um *software* é o seu tamanho, que será insumo para as estimativas de esforço, prazo e custo do projeto.

Antes, porém, põe-se uma questão metodológica: estabelecer e padronizar uma unidade para medir o tamanho de um *software*, problema amplamente comentado na literatura técnica, que apresenta várias soluções, com preponderância das chamadas métricas de tamanho funcional.

A métrica de tamanho funcional Pontos de Função (PF) é a mais utilizada na indústria de *software*, segundo o ISBSG (2008). A técnica que dimensiona um projeto de *software* em PF é denominada Análise de Pontos de Função (APF). Esta técnica consiste no dimensionamento do projeto de *software* levando em conta a funcionalidade requisitada e recebida pelo usuário, sem considerar como o projeto foi desenvolvido. A APF é uma técnica objetiva, gerando resultados consistentes, independentemente da pessoa que desempenha o papel de contador de Pontos de Função.

Há apenas quatro métodos para medição de tamanho funcional de *software* reconhecidos pela *International Organization for Standardization* (ISO): IFPUG CPM 4.1 (ISO/IEC 20926), NESMA CPM 2.1 (ISO/IEC 24570), Mark II CPM 1.3.1 (ISO/IEC 20968) e COSMIC-FFP *Measurement Manual* 2.2 (ISO/IEC 19761), como afirma Vazquez (2007).

Hazan (2005) destaca três unidades ou métricas principais para medição do tamanho de *software*: Pontos de Função (PF), Linhas de Código (LOC – *Lines of Code*) e Pontos por Casos de Uso (PCU), considerando vantajoso o primeiro sobre as demais alternativas, posição compatível com a de Vazquez (2007) que, reconhecendo a diversidade de métodos, afirma que nenhum deles conta com a aceitação de mercado conferida aos Pontos de Função.

Considerando as três alternativas apresentadas em Hazan (2005), parece consenso entre os autores que as medidas baseadas em LOC não são adequadas, cabendo a aplicação desse método somente em situações especiais. Até mesmo a facilidade para a contagem das linhas de código de um programa, tarefa que usualmente pode ser automatizada, é tratada com cautela, como por Vazquez (2007), que adverte que essa facilidade aparente é perigosa, havendo o risco de impossibilidade de comparação entre medidas, se não forem

tomados cuidados na definição das linhas que efetivamente serão contadas para esse fim. Um dos graves problemas da unidade LOC é que ela não é adequada para estimar um projeto de *software* nas etapas iniciais do ciclo de vida, onde existem apenas documentos iniciais de requisitos.

Esse problema não ocorre com a métrica PF, que permite, além de medir o tamanho do sistema em termos de funcionalidade fornecida ao usuário, estimar esse tamanho em qualquer fase do ciclo de vida, conforme Vazquez (2007).

Em relação à comparação entre PF e PCU, Hazan (2005) observa que esta última métrica, proposta por Gustav Karner em 1993, para estimar recursos em projetos modelados por casos de uso, não traz nenhum benefício adicional sobre a primeira, destacando também a maturidade da métrica PF, mais antiga, e, especialmente, a subjetividade da métrica PCU, sendo dependente da maneira pela qual os requisitos do projeto são documentados.

E, ainda, a PCU só pode ser aplicada em projetos cujos requisitos estejam documentados em casos de uso, ao passo que a métrica PF não é dependente da forma que o Engenheiro de Requisitos documenta os requisitos e pode ser utilizada em qualquer tipo de especificação. Assim, a métrica PCU não é recomendada para o estabelecimento de contratos de *software*, devido à sua limitação e, especialmente, subjetividade.

Uma vantagem da APF, apresentada por Hazan (2005), é que ela pode ser utilizada como insumo para o modelo COCOMO II, empregado para a derivação das estimativas de esforço, prazo e custo, objeto central deste trabalho, a partir de informações disponíveis sobre o tamanho do *software*.

Feitas essas considerações, adota-se neste estudo a técnica APF, que encontra-se descrita no Manual de Práticas de Contagem (CPM) versão 4.1.2, publicado pelo *International Function Point Users Group* (IFPUG), fundado em 1986.

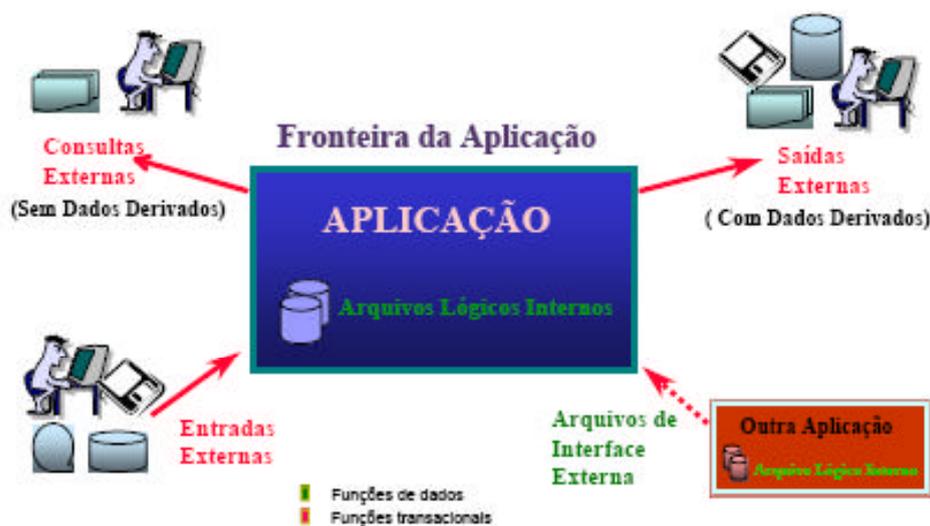
## Características, componentes e restrições da APF

Conforme mencionado, a APF está consagrada no mercado. Entretanto, embora exista desde 1979, seu uso no Brasil começou realmente na década de 90, com o apoio da empresa *Unisys*, consolidando-se mais tarde, quando grandes contratos públicos começaram a utilizar o PF como unidade, permitindo uma série de benefícios, entre os quais a redução dos riscos para a contratante e a possibilidade de estimar o tamanho e,

conseqüentemente, o custo do projeto de *software*, sendo o custo do PF predefinido no contrato, nas fases iniciais do ciclo de vida.

A contagem de PF, descrita com mais detalhes no próximo capítulo, permite obter o tamanho de um projeto de *software*, mapeando os requisitos funcionais nos cinco tipos de função da APF, ilustrados na Figura 1, a saber: Arquivos Lógicos Internos (ALI), Arquivos de Interface Externa (AIE), Entradas Externas (EE), Saídas Externas (SE) e Consultas Externas (CE), considerando a complexidade e contribuição funcional de cada um destes tipos de função. Existe ainda um Fator de Ajuste (FA), baseado nas características gerais da aplicação, utilizado nas contagens de Pontos de Função ajustados, conforme o IFPUG (2004).

Figura 1 – Visão Geral dos Tipos Funcionais da Análise de Pontos por Função



(HAZAN, 2005, p.13)

Apesar das inúmeras qualidades que possui, entretanto, é reconhecido que o uso da APF pode não ser adequado a algumas situações específicas. Segundo Vazquez (2007), em se tratando de manutenções, a APF propõe-se a medir apenas as evolutivas, isto é, aquelas que alteram os requisitos funcionais de um *software*.

Mesmo considerando a restrição da APF às manutenções evolutivas, há ainda algumas considerações restritivas, uma vez que, como o número de PF não considera a forma de implementação, quanto menor o escopo do projeto, mais será sentido o impacto dessa abstração dos detalhes na contagem final, o que não recomenda a medição de PF

para atividades não empreendidas como projetos, seja pela quantidade ou pelo pequeno tamanho, como aqueles até 50 PF, conforme Vazquez (2007).

O uso do PF para medir projetos pequenos também é considerado inadequado em Jones (2007), em função dos pesos atribuídos pelo método, que possuem um limite inferior relativamente alto. Em função disso, o autor considera que só faz sentido o uso de métricas como as do IFPUG para projetos de, no mínimo, 10 a 15 PF.

Outra limitação importante da métrica parece estar no Fator de Ajuste, aplicado após a contagem de PF não ajustados, resultando variação de até 35%.

Segundo Vazquez (2007), esse fator não reflete corretamente o impacto de características não funcionais do *software* sobre a produtividade, além de ter alto grau de subjetividade, o que justificaria a tendência atual do IFPUG de retirá-lo da próxima versão do manual de práticas de contagem. De fato, atualmente, o IFPUG também considera correta a contagem de Pontos de Função não ajustados. Além disso, o padrão ISO reconheceu como métrica de tamanho funcional somente os Pontos de Função não ajustados, ou seja, não reconheceu o FA. E, ainda, o insumo para o modelo COCOMO é a utilização de Pontos de Função não ajustados.

Atualmente, algumas organizações já estão começando a estabelecer seus contratos em Pontos de Função não Ajustados, como, por exemplo, o BNDES<sup>5</sup>.

Por fim, há também atividades não mensuráveis usando APF, como o suporte ao desenvolvimento ou outras atividades de apoio, que não são diretamente proporcionais à funcionalidade entregue pelos projetos, podendo ser melhor avaliadas, em muitos casos, pela contratação de horas de trabalho, como menciona Vazquez (2007).

## Fatores que podem limitar o uso da Contagem de Pontos de Função

A contagem de PF preconizada pelo IFPUG exige especificações razoavelmente completas e é bastante precisa. Entretanto, nem sempre haverá informações e tempo suficientes para a realização desse tipo de contagem, havendo no mercado diversos métodos de estimativa de PF, como as descritas em Vazquez (2007). Deve-se ressaltar que, por determinação legal ou para obter aprovação do projeto, a apresentação de estimativas é fundamental, já que os padrões atuais não admitem que uma organização inicie um projeto sem informações essenciais, como o objetivo e as estimativas de esforço necessário, prazo de entrega e preço.

---

<sup>5</sup> Edital da Concorrência AA 01/2007-BNDES, Anexo I, disponível em [www.bndes.gov.br](http://www.bndes.gov.br)

Os métodos de estimativa são classificados usualmente em diretos e derivados. Um exemplo do primeiro grupo é a analogia simples, em que se compara o projeto atual com outros projetos similares concluídos, obtidos em uma base histórica. Desde que o grau de similaridade seja suficiente e que os dados históricos sejam confiáveis e atualizados, é possível estimar-se o tamanho do projeto de *software*.

No entanto, os métodos mais utilizados na área das estimativas são os derivados ou paramétricos, que estimam o tamanho do *software* a partir de variáveis relacionadas a alguns de seus atributos, como a contagem dedutiva, baseada apenas no número de Arquivos Lógicos Internos, conforme descrito em Vazquez (2007).

Apesar de menos precisas que a contagem de PF, duas situações comuns em que há necessidade do uso de técnicas de estimativa são as seguintes: dimensionamento de sistemas legados, usualmente sem documentação adequada e contendo milhões de linhas de código, e obtenção do tamanho de um novo sistema nas etapas iniciais do ciclo de vida. Para o primeiro caso, é freqüente a aplicação da técnica conhecida como *backfiring*, ao passo que, para a segunda situação, são amplamente utilizadas técnicas de estimativa de PF, dentre as quais se destaca a Contagem Indicativa e a Contagem Estimativa, publicadas pela *Netherlands Software Metrics Association* (NESMA) (2008).

O *backfiring*, como método de estimativa, é visto com reservas pelos especialistas. Ele é considerado arriscado, já que apresenta erros consideráveis em relação à contagem de PF. Essencialmente, o método consiste em calcular o tamanho em PF a partir do número de linhas de código, utilizando um fator de conversão fixo para cada linguagem de programação. Entretanto, há divergência entre os autores quanto ao valor desse fator em mais de 100%, como na linguagem Cobol, para a qual autores diferentes atribuem 77 LOC por PF, 107 ou mesmo 177, como menciona Vazquez (2007).

Outro problema importante é descrito em Jones (2007), afirmando que a precisão desse método não é grande, podendo ter erros superiores a 50% para mais ou para menos, já que estilos de programação individuais podem causar grandes variações na contagem de linhas de código.

Apesar das críticas, a prática é comum no mercado, já que é uma maneira rápida e de baixo custo para se obter o tamanho do acervo de aplicações de uma organização. Jones (2007) considera esta prática quando se trabalha com estimativas de tamanho de sistemas legados, nos quais somente o código fonte é conhecido.

Além das raras situações em que se faz uso do *backfiring*, há situações de uso dos métodos de Contagem Indicativa e Contagem Estimativa, bastante empregados nas etapas

iniciais do ciclo de desenvolvimento, quando o projeto não é conhecido em sua totalidade, mas alguns dos requisitos já foram consolidados.

Na Contagem Indicativa, admite-se que já foram identificadas as funções de dados do projeto: Arquivos de Interface Externa (AIE) e Arquivos Lógicos Internos (ALI). O método então atribui 35 PF para cada ALI e 15 PF para cada AIE identificado, sendo esses números obtidos conforme o detalhamento a seguir, obtido em Vazquez (2007) e Hazan (2005).

O método considera como premissa a complexidade média para todos os tipos de função da APF. E, ainda, cada ALI representa 10 PF e tem associadas 3 entradas externas para inclusão, alteração e exclusão dos dados do arquivo (12 PF), 2 consultas externas (8 PF) e 1 saída externa, correspondente a relatório com totalizações (5 PF), perfazendo o total de 35 PF. Além disso, cada AIE (7 PF) tem associadas 2 consultas externas, correspondentes a uma consulta detalhada e uma lista dos dados da tabela (8 PF), alcançando os 15 PF utilizados pelo método.

Apesar da simplicidade desse método em relação à contagem de PF, ele não é imune a críticas. Hazan (2005) constata que a prática tem demonstrado que superestima o tamanho da maioria dos projetos, sugerindo como alternativa a abordagem denominada contagem indicativa inteligente, em que os fatores 35 e 15 são alterados em função dos dados históricos da organização.

Essa crítica é compatível com a observação em Vazquez (2007), salientando que a contagem indicativa apresenta maior dispersão de resultados em relação à contagem final de PF, se comparada à Contagem Estimativa, descrita a seguir.

Em uma fase geralmente posterior no ciclo de desenvolvimento, quando já são conhecidas, em linhas gerais, todas as funcionalidades do sistema, Vazquez (2007) reconhece a simplicidade da Contagem Estimativa, publicada pela NESMA, que preconiza a identificação de todos os tipos de função da APF, considerando baixa a complexidade para as funções de dados ALI e AIE, e média para as funções transacionais EE, SE e CE.

Os autores pesquisados, embora reconhecendo a existência dos métodos de estimativa e suas variações, são unânimes em alertar para a necessidade de cautela na utilização dos parâmetros, que devem ser ajustados à realidade dos projetos e da organização em que são desenvolvidos.

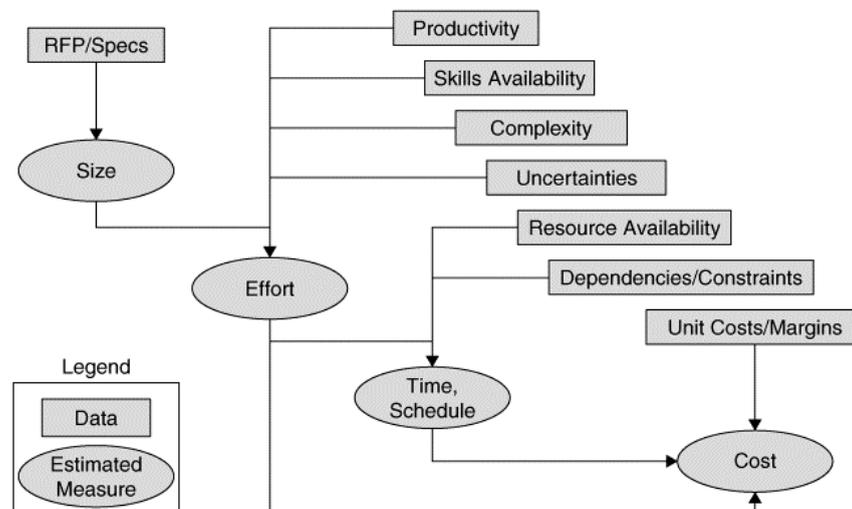
## Utilização do tamanho do *software* para determinação do custo

Após obter o tamanho do sistema a ser desenvolvido em Pontos de Função, a partir das técnicas anteriormente mencionadas ou similares, a organização contratante já possui um importante parâmetro para estimar o custo a ser despendido no projeto, mas essa tarefa não é trivial.

De fato, embora os contratos baseados em PF tenham usualmente o preço calculado como o produto do preço unitário do PF pelo número de PF a serem entregues, o tamanho do *software* só permite determinar o segundo fator. O primeiro fator, o preço unitário do PF, pode variar amplamente no mercado, assim como outros parâmetros relevantes, como esforço e prazo.

Como adverte Vazquez (2007), o número de PF não mede diretamente esforço, produtividade ou custo. É apenas medida de tamanho funcional do *software* que, em conjunto com outras variáveis, poderá ser usado para derivar produtividade e, assim, estimar esforço e custo, como ilustra a Figura 2.

**Figura 2 – Insumos típicos para estimativas de projetos**



(PARTHASARATHY, 2007, p. 266)

Quais dessas variáveis são relevantes em cada caso concreto é a grande pergunta que se procura fazer no momento da estimativa dos limites aceitáveis de custo para um dado projeto.

À semelhança do que se observou anteriormente para determinação do tamanho do *software*, há várias técnicas para estimativas de custos, como modelagem algorítmica,

julgamento de especialista, analogia, Lei de *Parkinson* e a atribuição de preço para ganhar, conforme descrito em Sommerville (2007).

Estas duas últimas técnicas consistem em categorias curiosas, uma vez que partem de premissas radicais, embora existentes na prática: a primeira determina o custo a partir dos recursos disponíveis, assumindo que o projeto se expandirá até alocar para si todos os recursos; a segunda, que o custo é determinado pelo orçamento disponível, que funciona como limite para as demandas que podem ser transformadas em requisitos. Assim, se uma organização dispõe de 5 pessoas e orçamento de R\$ 200.000,00, essas seriam duas estimativas realistas para recursos alocados e custo total do projeto.

Mas é na modelagem algorítmica de custos que reside o maior interesse para os fins a que se propõe este estudo. Ela também pode ser combinada com outras técnicas, para comparação de resultados, o que é especialmente útil em projetos de grande porte, segundo Sommerville (2007).

Esse tipo de modelagem trabalha a partir de informações históricas de custos, usualmente relacionando o tamanho do *software* aos custos do projeto, ao esforço para produção e ao prazo de entrega.

O esforço desempenha um papel crucial nesses modelos, uma vez que a indústria do *software* é intensiva em trabalho, consistindo a maior parcela dos custos envolvidos na remuneração dos gerentes de projeto, analistas e programadores. Assim, um PF será mais caro se demandar mais tempo para ser produzido, o que pode ocorrer em função de uma série de fatores, como ambiente de desenvolvimento e linguagem de programação, que afetam a produtividade, como analisa Sommerville (2007).

Assim, pode-se esperar que a produtividade média para desenvolvimento de sistemas *Web* com o uso da linguagem ASP seja maior do que se for utilizada a linguagem JAVA, já que são necessárias 10 horas para implementação de um PF em ASP e 15 horas para que se obtenha a mesma funcionalidade em Java, mas ambas apresentam produtividade menor que *Visual Basic* e *Delphi*, no ambiente cliente servidor, sendo necessárias apenas 6,8 horas para implementar um PF nessas linguagens, conforme dados de Hazan (2005).

A mediana da produtividade dos projetos de governo na base do ISBSG (2008) é de 9,7 horas por PF, mas ela varia em função do ambiente de desenvolvimento, obtendo-se 13,3 horas em projetos usando linguagens de terceira geração, ou 5,2 horas para linguagens de quarta geração.

Outros efeitos destacados na relação do ISBSG são decorrentes da estabilidade da equipe durante o projeto, aumentando a produtividade, e do aumento do número de usuários, locais ou unidades de negócio, reduzindo a produtividade.

O primeiro dos aspectos mencionados no parágrafo anterior é particularmente preocupante na realidade brasileira, em que a área de tecnologia da informação possui alta rotatividade de mão-de-obra, agravada algumas vezes por baixa remuneração oferecida pelas empresas desenvolvedoras de *software*, como forma de se tornarem mais competitivas nos processos de licitação pública, reduzindo o valor dos contratos.

Diversos dos trabalhos pesquisados advertem para a dificuldade de encontrar fórmulas gerais para conversão do tamanho de um *software* (S) em Esforço (E), Prazo (P) e Custo (C), que podem conduzir a erros grosseiros em alguns casos, admitindo-se, quando muito, que elas são fáceis de usar e podem permitir uma verificação inicial da razoabilidade para outras estimativas produzidas, como observa Jones (2007).

A realidade brasileira atual, entretanto, não permite muito mais que isso ao gestor público, quando deparado com a obrigação legal de estimar preços de *software*, além do recurso, sempre recomendado, a consultas ao mercado.

Em Vazquez (2007), encontra-se crítica às relações simplificadas entre essas variáveis, em que as fórmulas são lineares, como a do esforço, definido como a taxa de entrega multiplicada pelo tamanho funcional. Por exemplo, admitindo-se que a taxa de entrega de uma empresa seja de 50 Homem-hora por PF e um sistema tenha 1.000 PF, o esforço será de  $50 \times 1.000 = 50.000$  Homem-hora (Hh). O cálculo é bastante simples, mas essas relações repousam em premissas que dependem de fatores externos, não sendo possível afirmar, por exemplo, que a taxa de entrega é constante para um projeto.

Em Hazan (2005), verifica-se que o fator de conversão de PF para Hh depende de diversos outros atributos do projeto, como plataforma de desenvolvimento, complexidade da aplicação e experiência da equipe, devendo ser estabelecido segundo as características de cada projeto, o que recomenda a criação de bases de dados estratificadas, com informações sobre atributos dos projetos mais relevantes para a organização.

Antes de passar à questão do custo do *software* propriamente dito, é conveniente observar o exemplo tratado em Hazan (2005), de modo a sedimentar, a partir de um caso real de análise de uma ferramenta de *e-learning*, os conceitos até agora apresentados, partindo-se da premissa de que o tamanho do *software* já foi estimado em 671 PF ajustados.

O prazo ótimo para desenvolvimento ( $T_d$ ) é calculado a partir de uma das fórmulas de bolso de Jones (2007), onde o prazo é obtido a partir do tamanho da aplicação, estimado em PF, elevado a um expoente  $t$ , que é igual a 0,40 para aplicações comerciais. Tem-se então  $T_d = 671^{0,40} = 13,5$  meses.

Sendo o sistema desenvolvido em Java, utiliza-se a produtividade média tabelada para essa linguagem, de 15 h / PF, resultando em esforço de  $671 \text{ PF} \times 15 \text{ h} / \text{PF} = 10.065$  horas, ou, admitindo-se um mês com 22 dias úteis e um dia com 6 horas produtivas, 10.065 horas / 22 dias por mês / 6 horas por dia, ou 76,3 Homem-meses, aproximadamente.

Para um esforço de 76,3 Homem-meses e um prazo ótimo de 13,5 meses, o número de pessoas a serem alocadas ao projeto é de  $76,3 \text{ Hm} / 13,5 \text{ m}$ , ou 6 pessoas, aproximadamente.

Mas quanto esse sistema vai custar efetivamente ao demandante no mercado brasileiro? Nenhuma das referências pesquisadas estabeleceu parâmetros exaustivos ou métodos de estimativa que dessem uma resposta imediata, mas a determinação do esforço e o cronograma de atividades podem ser em um bom ponto de partida.

Para Sommerville (2007), as estimativas de custo, esforço e prazo do projeto são normalmente realizadas em conjunto, já que os custos de desenvolvimento são, principalmente, os custos de esforço envolvido, sendo este usado para ambas as estimativas, de custo e de cronograma.

Em Jones (2007), há uma metodologia de estimativa abrangente e detalhada, dividindo o processo de estimativa de custo em um padrão com 10 atividades: 1) analisar os requisitos; 2) estimar o tamanho; 3) identificar as atividades a serem incluídas, como análise e codificação; 4) estimar defeitos potenciais e métodos para sua remoção; 5) estimar equipe; 6) ajustar premissas baseado na capacitação e experiência; 7) estimar esforço e cronograma; 8) estimar custos de desenvolvimento; 9) estimar custos de manutenção e melhoria; 10) apresentar a estimativa ao cliente e defender sua aprovação.

Ao descrever a oitava atividade, entretanto, o autor deixa claro que os custos de desenvolvimento são muito complexos e, apesar de dependentes do esforço e do cronograma, são afetados por outros fatores, como os salários dos diversos componentes da equipe; encargos; taxas de inflação e de câmbio, dependendo da duração e natureza do projeto; licenças; despesas de capital; viagens e transporte; administração de contratos e subcontratação; impostos; elaboração de tutoriais e treinamento.

Apesar de mais complexa, a abordagem de Jones em 10 itens trata de questões bastante comuns na contratação de projetos de *software*, como a necessidade de orçar os

custos com manutenção corretiva do produto a ser gerado, usualmente sem ônus para o cliente durante um período de garantia mínimo, bem como a necessidade de levar em conta, na determinação do custo, a existência de defeitos no *software*.

Há inúmeras variáveis a se levar em conta no momento do estabelecimento do custo, devendo ser verificado, em cada caso, quais aquelas que são relevantes em relação ao valor total.

Em geral, a abordagem utilizada, conforme Jones (2007), é estimar o *software* em termos de horas, dias ou meses de esforço, aplicando-se então os demais custos.

Para fins de controle, as referências pesquisadas parecem apontar para a necessidade de, em cada requisição de serviços, haver a definição de uma tabela básica com as características relevantes do *software*, bem como o percentual de Lucros e Despesas Indiretas - LDI, usada largamente entre os profissionais da engenharia, correspondentes a custos tomados em consideração na elaboração da proposta, mas não relacionados diretamente ao desenvolvimento.

Em Sommerville (2007), há três parâmetros envolvidos no cálculo do custo total de um projeto de desenvolvimento de *software*: os de *hardware* e *software*, incluindo manutenção; de viagens e treinamentos; e de esforço, que incluem os custos indiretos, como subsistência, iluminação, pessoal de apoio, como contadores e faxineiros, operação de redes, instalações centrais, seguridade social e benefícios, como seguro-desemprego. Em geral, o preço cobrado pelos serviços é simplesmente o custo mais o lucro.

Há enumeração de alguns custos associados também em Hazan (2005), como o custo por perfil do profissional, recursos computacionais, treinamento, consultoria, viagens e custos indiretos.

Portanto, para que se possa derivar o custo final de um projeto, conhecidos os parâmetros já levantados, faz-se necessário que existam levantamentos específicos a respeito desses componentes de custos, incluindo os salários e encargos sociais correspondentes aos diversos profissionais envolvidos no projeto e outras despesas associadas.

Até o momento, todavia, não constam ainda resultados expressivos para a consolidação desses levantamentos no mercado, o que leva a solução desse item para a competição entre os potenciais fornecedores, contra-indicando a aceitação de realização desses serviços sem o devido procedimento licitatório.

A exigência de que esses custos estejam expressos nas propostas, entretanto, é um importante ponto de referência, de modo a criar na área de *software* a cultura da

composição unitária dos preços, discriminando-se os diversos itens de custo, de modo que seja facilitada eventual renegociação de preços.

Mesmo na literatura internacional, não foram encontradas tabelas confiáveis com esses custos. Conforme Jones (2007), por exemplo, as informações sobre custo são extremamente variáveis devido a grandes diferenças entre a composição de equipes e o uso de especialistas, variações no salário médio e horas extras não pagas. Feitas essas e várias outras ressalvas, o autor informa que o custo médio do PF nos Estados Unidos é de 1.200 dólares.

Estabelecido o preço do *software*, chegar-se-ia, em princípio, ao preço unitário de um PF, bastando dividir o preço total pelo tamanho em PF. Por exemplo, se o valor do contrato é de R\$ 100.000,00 e há 100 PF, cada PF será orçado em R\$ 1.000,00.

Não é raro, entretanto, que seja feita contratação parcial do ciclo de desenvolvimento. Por exemplo, somente a codificação em uma determinada linguagem, ou somente a análise de requisitos.

Coloca-se, assim, mais uma questão importante: deve ser paga fração do valor do PF, dependendo do trabalho realizado, ou o fornecedor deve calcular preços diferenciados para o PF, dependendo do serviço?

A regra que vem evoluindo no mercado é que o PF do contrato seja o cheio, correspondente a todas as etapas do ciclo de desenvolvimento, de modo a dar mais flexibilidade à contratação. Não sendo necessário todo o desenvolvimento, em uma ordem de serviço específica, aplica-se então um percentual fixo, correspondente aos serviços efetivamente prestados. Por exemplo, se o preço do contrato para um PF for de R\$ 800,00 e o contrato fizer somente análise e projeto, paga-se 35% desse valor, ou outro percentual estipulado desde o edital.

Como variam as etapas em função da metodologia de desenvolvimento utilizada e há variações em função de outros fatores, não há ainda uma resposta geral, mas há regras conhecidas no mercado.

Há referências em ISBSG (2008), por exemplo, com uma média de 35% do PF para análise e projeto, para um ciclo simplificado. Em Jones (2007), com um ciclo bem mais complexo, prevendo até 25 atividades a serem consideradas no desenvolvimento, os maiores percentuais são atribuídos a codificação (23%), elaboração de documentação do usuário (12%) e gerenciamento do projeto (11%).

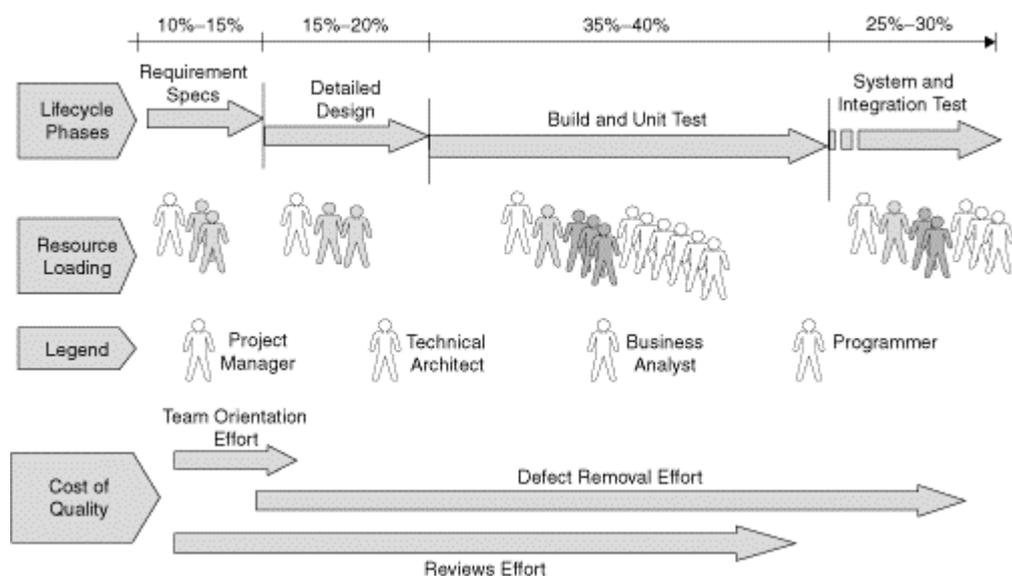
Em contratação do Serpro mencionada em Vazquez (2007), houve essa estratificação do ponto de função, a partir da divisão média do esforço empregado em cada

fase, constando 20% para análise de requisitos, 30% para análise e projeto, 40% para integração e testes e 10% para implantação.

Sugestões de faixas percentuais para cada fase também são encontradas em Parthasarathy (2007), constando 10 a 15% para especificação de requisitos, 15 a 20% para projeto detalhado, 35 a 40% para construção e testes unitários e 25 a 30% para testes de sistema e integração, como se vê na Figura 3.

Em geral, os autores advertem para o perigo dessa estratificação com base em tabelas prontas, uma vez que ela pode variar, como Jones (2007), que considera 5 conjuntos de variáveis independentes afetando essa distribuição: a classe da aplicação, o tamanho, a linguagem de programação, a presença ou ausência de reutilização e a metodologia de desenvolvimento utilizada.

**Figura 3 – Parâmetros de produtividade**



(PARTHASARATHY, 2007, p. 161)

Diante de todos os conceitos e fórmulas apresentados, um ponto que merece ser comentado trata da possibilidade de utilização de meios automatizados para realização desses procedimentos. De fato, há várias ferramentas no mercado para esse fim. Em Jones (2007), distinguem-se 6 métodos para estimativa de custos de *software*, 3 manuais e 3 automatizados, sendo os últimos considerados mais rigorosos e sofisticados.

Uma opção gratuita de *software* para auxiliar nessas estimativas está disponível no *site* [http://sunset.usc.edu/research/COCOMOII/expert\\_cocomo/expert\\_cocomo2000.html](http://sunset.usc.edu/research/COCOMOII/expert_cocomo/expert_cocomo2000.html), baseada no modelo denominado COCOMO, desenvolvido por Boehm na década de 80.

Conforme Jones (2007), esse permanece o único modelo cujos algoritmos não são tratados como segredos, tendo sido desenvolvido um novo modelo, o COCOMO II, que suporta APF, entre outras melhorias.

Indicações de outras alternativas podem ser facilmente obtidas na Internet, como no endereço eletrônico [http://www.laatuk.com/tools/effort\\_estimation\\_tools.html](http://www.laatuk.com/tools/effort_estimation_tools.html), que lista inclusive ferramentas proprietárias, como o SLIM-Estimate, da Quantitative Software Management– QSM, o Checkpoint, da Software Productivity Research – SPR e o CostXpert, da Marotz, Inc. No mercado nacional, uma proposta interessante encontra-se no WinFPA, da Fernando Ferreira Consultoria Ltda, disponível para avaliação em <http://www.winfpa.com.br>.

## Capítulo 2

### ANÁLISE DA COMPATIBILIDADE DOS PREÇOS FINAIS

No capítulo anterior, foram tratadas algumas questões recorrentes na área de contratação de serviços de *software*, como as estimativas do projeto, algumas das quais ainda carecendo de pesquisas adicionais para aplicação segura no mercado brasileiro.

Aprovado o orçamento e realizada a contratação, entretanto, o problema da aplicação dos recursos do contratante ainda não está resolvido. Frequentemente, ocorrem estouros de prazo e de orçamento, decorrentes de especificações erradas ou incompletas, além de deficiência ou inexistência de estimativas. Como nem sempre o prazo pode ser dilatado nem o orçamento é flexível para acomodar grandes variações, esse comportamento pode levar à cessação dos investimentos e abandono do projeto, com prejuízo para a organização.

#### Causas de variação nos projetos

Conforme mencionado no capítulo anterior, na base do ISBSG (2008) há disparidade de preços em contratações da ordem de até 10 vezes, havendo ainda, em média, estouro de prazo em 37% dos projetos, e de custo em 22%, aproximadamente.

A realidade brasileira contém exemplos ilustrativos desses problemas, relacionados à deficiência na condução do processo de especificação, contratação e fiscalização da execução dos serviços: sistemas que não atendem a especificações de desempenho e segurança, ou violam regras do negócio; fornecedores que pleiteiam aditivos aos contratos celebrados, por entenderem que houve aumento no objeto; cronogramas que não são cumpridos, com acusações mútuas entre contratante e contratado sobre a responsabilidade pelos atrasos.

Por exemplo, o fornecedor estabelece um preço para o projeto, imaginando que apenas implementará regras de negócio bem definidas e consolidadas, ao passo que o contratante pretende que ele estude os processos de negócio e proponha melhoria, assumindo todas as etapas do ciclo de desenvolvimento, incluindo extensa documentação do sistema e do usuário final. Essa questão ganha vulto porque muitas das aspirações do contratante não são expressas nas requisições de serviços, editais ou contratos, gerando conflitos que representam riscos consideráveis para a continuidade do projeto.

Outro problema comum é a ausência ou inadequação de estimativas por parte do contratante e/ou contratado. Na fase de estimativas, por exemplo, pode ter sido usada a já mencionada técnica da analogia e estimados tamanho, esforço, prazo e custos mas, durante a implementação do projeto, verificuem-se diferenças substanciais entre o sistema em desenvolvimento e aquele tomado como paradigma, resultando em maior prazo para implementação.

Um último exemplo desses problemas, talvez o mais grave, reside na validação dos artefatos produzidos. É comum haver problemas relacionados não só à produtividade, mas também à qualidade dos serviços, gerando devolução do fiscal do contrato para retrabalho, onerando o desenvolvedor e provocando atraso no cronograma do contratante. Se ambos não estiverem em perfeita sintonia nesse processo, haverá insatisfação recíproca entre as partes, com prejuízo para o projeto e para os usuários finais.

Esse contexto evidencia a necessidade da existência de estimativas realistas, bem como de base histórica que as fundamente, para que sejam mais consistentes e comprometidas com a realidade de cada organização. Se uma equipe de programadores de *mainframe* acaba de ser treinada em Java e vai começar a desenvolver um novo sistema, por exemplo, não se deve esperar que eles atinjam a produtividade anunciada pelas tabelas publicadas, como 15 horas por PF.

Também deixa claro que, por mais que os modelos simplificados, como a fórmula de prazo de Jones, sejam úteis para o início das estimativas, eles não prescindem do trabalho de análise do caso específico e de um processo de melhoria contínua das informações das bases históricas.

Um último ponto que merece ser mencionado é que o preço do Ponto de Função no mercado brasileiro ainda está em fase de acomodação, havendo variações bastante acentuadas entre contratos distintos, não apenas em função de fatores endógenos, como tecnologia e capacitação dos profissionais, mas também da própria inexperiência das partes sobre as variáveis relevantes para o estabelecimento do preço unitário.

Nesse aspecto, há necessidade de elaboração de especificações mais detalhadas em relação às variáveis que afetam os custos do fornecedor, evitando surpresas, bem como o estímulo à concorrência entre os potenciais fornecedores e a exigência de que seja expressa a composição do preço praticado em planilhas, o que contribuirá para a redução desse nível de variação a patamares mais aceitáveis.

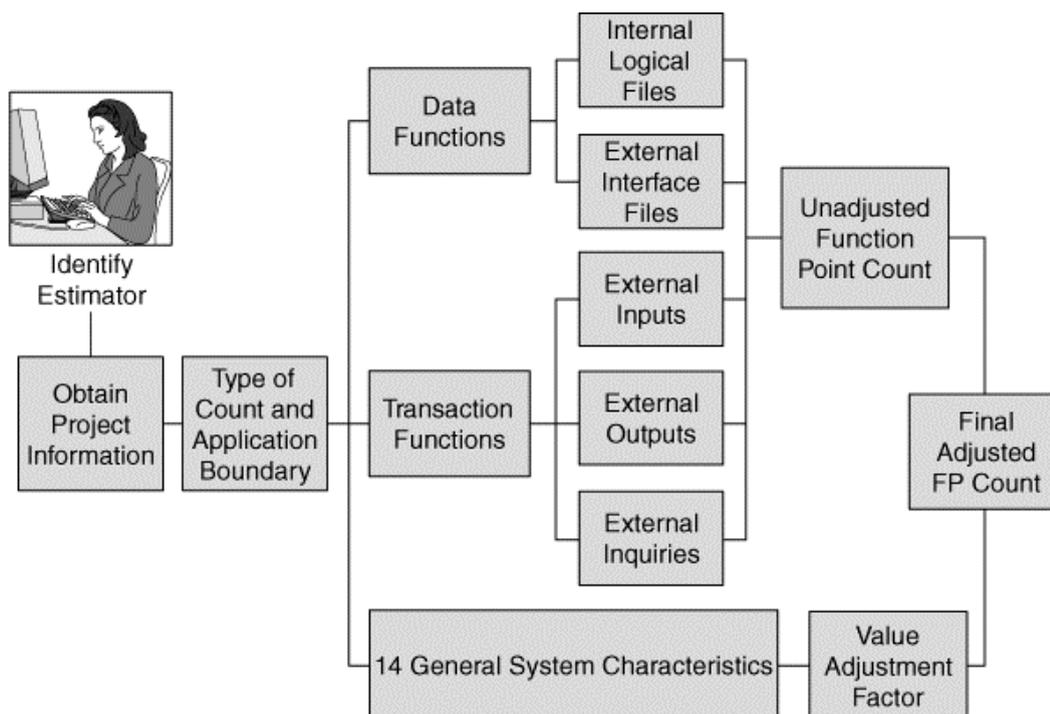
## A Contagem de Pontos de Função

Concluído o projeto de *software*, podem ser encerrados os ciclos de estimativas, passando-se à sua medição, já que o produto final existe e seus componentes estão disponíveis para mensuração, que pode ser feita pela APF.

Nesse momento, a avaliação *ex post*, medindo o produto entregue, pode indicar que as estimativas foram confirmadas ou mesmo superadas, mas também pode ter havido insucesso, demandando análise mais detida para apurar as responsabilidades pelos prejuízos, bem como para evitar a repetição de problemas semelhantes no futuro.

A versão atual das práticas de contagem do IFPUG é a 4.1.2, sendo a técnica descrita em mais detalhes no Anexo 1 deste trabalho. Neste ponto, é feita apenas uma explanação geral do método, conforme ilustrado na Figura 4, a seguir.

**Figura 4 – O processo de contagem da APF**



(PARTHASARATHY, 2007, p.58)

São três os tipos de contagem do IFPUG (2004): desenvolvimento, melhoria (manutenção evolutiva) e aplicação. Segundo Vazquez (2007), para o primeiro tipo mede-se a funcionalidade fornecida aos usuários finais do *software* quando da sua primeira instalação, incluindo as funções de conversão de dados necessárias à implantação. No segundo tipo, medem-se as funções adicionadas, modificadas ou excluídas do sistema pelo

projeto, além das eventuais funções de conversão de dados. No terceiro e último tipo, também conhecido como *baseline*, mede-se a funcionalidade total fornecida aos usuários por uma aplicação instalada em um determinado momento, não sendo consideradas as funções de conversão de dados.

Então, identifica-se o escopo da contagem e a fronteira da aplicação. O primeiro “define as funcionalidades que serão incluídas em uma contagem de pontos de função específica” e o segundo “indica o limite entre o *software* que está sendo medido e o usuário” (IFPUG, 2004, pp. G-1 e G-2).

Dessa forma, o escopo define o que será medido, podendo inclusive abranger mais de uma aplicação, e a fronteira permite distinguir os elementos internos e externos a essa aplicação.

A seguir, contam-se as funções de dados e transacionais, definidas as primeiras como aquelas que representam as funcionalidades fornecidas ao usuário para atender às necessidades de armazenamento de dados, e as segundas como as que representam os requisitos de processamento fornecidos pelo sistema ao usuário, segundo Vazquez (2007). As funções de dados são os Arquivos Lógicos Internos (ALI) e os Arquivos de Interface Externa (AIE) e as transacionais são as Entradas Externas (EE), Saídas Externas (SE) e Consultas Externas (CE).

Essas funções têm uma definição formal, transcrita abaixo, que é complementada por observações e exemplos práticos ao longo do manual do IFPUG, de modo a evitar dúvidas em algum caso específico, o que é bastante comum, tendo em vista a diversidade de tipos de projetos de *software* sobre a proposta original do método, de 1979.

Um arquivo lógico interno (ALI) é um grupo de dados ou informações de controle logicamente relacionados, identificado pelo usuário e mantido dentro da fronteira da aplicação. A intenção primária de um ALI é armazenar dados mantidos através de um ou mais processos elementares da aplicação que está sendo contada.

Um arquivo de interface externa (AIE) é um grupo de dados ou informações de controle logicamente relacionados, identificado pelo usuário e referenciado pela aplicação, porém mantido dentro da fronteira de outra aplicação. A intenção primária de um AIE é armazenar dados referenciados por um ou mais processos elementares dentro da fronteira que está sendo contada. Isto significa que um AIE contado para uma aplicação deve ser um ALI em outra aplicação.

(IFPUG, 2004, p. 2-7)

Uma entrada externa (EE) é um processo elementar que processa dados ou informações de controle que vêm de fora da fronteira da aplicação. A intenção primária de uma EE é manter um ou mais ALIs e/ou alterar o comportamento do sistema.

Uma saída externa (SE) é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação. A intenção primária de uma SE é apresentar informações ao usuário através de lógica de processamento adicional à recuperação de dados ou informações de controle. A lógica de processamento deve conter pelo menos uma fórmula matemática ou cálculo, ou criar dado derivado. Uma saída externa também pode manter um ou mais ALIs e/ou alterar o comportamento do sistema.

Uma consulta externa (CE) é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação. A intenção primária de uma consulta externa é apresentar informações ao usuário através da recuperação de dados ou das informações de controle. A lógica de processamento não contém fórmula matemática ou cálculo, nem cria dado derivado. Nenhum ALI é mantido durante o processamento, nem é alterado o comportamento do sistema.

(IFPUG, 2004, p. 2-8)

A cada uma das funções analisadas no sistema em medição, o método associa uma complexidade, que pode ser baixa, média, ou alta, atribuindo-se um número de PF a partir dessa associação, dependendo do tipo de função. Assim, um ALI de complexidade alta tem associados 15 PF, ao passo que uma CE de complexidade alta tem 6 PF, conforme o manual do IFPUG (2004).

De posse das planilhas de cálculo preenchidas com as funções de dados e as transacionais, a próxima etapa do método consiste em calcular o número de PF não ajustados, obtido pelo somatório dos Pontos de Função de todas as funções identificadas.

Obtido o número de PF não ajustados, o resultado final da contagem, chamado de PF ajustados, é obtido pela multiplicação dos PF não ajustados pelo Fator de Ajuste, que varia entre 0,65 e 1,35, isto é, representando um impacto de até 35%, para cima ou para baixo sobre os Pontos de Função não ajustados. Esse fator é obtido a partir da pontuação de 14 Características Gerais do Sistema (CGS), descritas no Anexo 1 deste trabalho.

Para adequar-se ao padrão ISO/IEC de medição funcional, o IFPUG tornou o FA opcional na aplicação da técnica APF. Além disso, uma crítica importante é que a variação de  $\pm 35\%$  não é suficiente para representar adequadamente as funcionalidades gerais da aplicação, como observa Vazquez (2007).

Outro problema relacionado ao FA é apresentado por Hazan (2005), comentando que seu cálculo é difícil e demorado, especialmente para os profissionais sem experiência em contagem de PF e, por isso, muitas organizações fixam um único fator para todas as aplicações na mesma plataforma. Apesar das limitações, ele é previsto no manual de práticas de contagem e pode ser usado para considerar requisitos não funcionais no tamanho *software*, como vem sendo feito pelo Serpro.

É recomendável, porém, como parâmetro único para toda a aplicação, que o valor do FA seja devidamente comunicado aos potenciais fornecedores, desde o edital ou do momento em que venha a ser determinado.

Tratando da experiência obtida em mais de 500 sistemas, Hazan (2005) observa os fatores de ajuste entre 0,8 e 0,9 para sistemas *batch*, em torno de 1 para sistemas *mainframe on-line* ou *micro standalone*, entre 1,1 e 1,2 para sistemas cliente/servidor e *web*, além de fatores maiores que 1,25 para alguns sistemas cliente/servidor mais complexos.

Calculado o FA, o último passo para contagem consiste na aplicação de fórmulas associadas a cada tipo de contagem, conforme Vazquez (2007), sendo:

*Para projetos de desenvolvimento:*

$$DFP = (UFP + CFP) \times VAF,$$

sendo *UFP* e *CFP*, respectivamente, o número de PF não ajustados correspondentes às funções disponíveis após a instalação e às funções de conversão de dados, e *VAF* o fator de ajuste

*Para projetos de melhoria (enhancement):*

$$EFP = [(ADD + CHGA + CFP) \times VAFA] + (DEL \times VAFB),$$

sendo *ADD* o número de PF não ajustados das funções incluídas pelo projeto de melhoria; *CHGA*, o número de PF não ajustados, após os serviços, das funções modificadas; *DEL*, o número de PF não ajustados das funções excluídas; *VAFA* é o fator de ajuste depois do projeto; *VAFB* é o fator de ajuste antes do projeto

*Para projetos de aplicação, há 2 fórmulas disponíveis:*

$$AFP = ADD \times VAF,$$

para a contagem inicial, sendo *ADD* o número de PF das funções instaladas

$$AFP = [(UFPB + ADD + CHGA) - (CHGB + DEL)] \times VAFA,$$

*para a contagem após projetos de melhoria, sendo UFPB o número de PF não ajustados antes do projeto; ADD, o número de PF das funções incluídas; CHGA, o número de PF, após o projeto, das funções alteradas; CHGB, o número de PF, antes do projeto, das funções alteradas; DEL, o número de PF das funções excluídas; VAFA é o fator de ajuste após o projeto*

Concluída a avaliação *ex post* sobre o tamanho funcional do sistema e levando-se em consideração os modelos de estimativas descritos no capítulo precedente, uma série de perguntas relevantes sobre a conformidade da contratação e execução do contrato poderão ser respondidas, tais como:

- ✍ O sistema atende às especificações originais do projeto?
- ✍ A contagem de Pontos de Função final indicou um crescimento anormal do sistema em relação à estimativa?
- ✍ Há justificativa para a alteração?
- ✍ Em caso de contratação baseada em Homem-hora ou equivalente, o tamanho do *software* é compatível com o esforço alegado pelo fornecedor ou atestado pelo contratante?
- ✍ Os recursos mobilizados pelo fornecedor foram compatíveis com o esforço estimado e com o prazo estabelecido no contrato?
- ✍ O prazo de desenvolvimento foi compatível com o tamanho do produto final gerado e com as estimativas contratuais?

As respostas a essas questões poderão sugerir a ocorrência de irregularidades na execução do contrato, ou a confirmação das estimativas iniciais. Além disso, a comparação dos custos reais do desenvolvimento com as estimativas constantes do projeto básico poderão sugerir alterações no modelo utilizado pelo gestor para elaboração das estimativas e constituir uma base para futuras negociações.

## Capítulo 3

### USO DA APF EM PROCEDIMENTOS DE AUDITORIA

Os atos dos gestores de órgãos e entidades governamentais estão sujeitos a controles específicos de diversas naturezas, o que torna a administração de recursos públicos uma atividade com objetivos às vezes conflitantes. Assim, ao lado de requisitos de eficiência e eficácia, comuns às organizações privadas, há questões envolvendo equidade, como a preferência por micro-empresas em algumas situações, e princípios constitucionais e disposições legais específicos, como a publicidade e a regra das licitações para aquisição de produtos e serviços, incluindo *software*.

#### O controle das licitações e contratos no setor público brasileiro

O gestor público está submetido a uma série de controles dos seus atos, como os provenientes do Ministério Público e do Poder Judiciário. Na área administrativa federal, os atos, inclusive as contratações de bens e serviços, estão sujeitas aos controles interno, no âmbito dos poderes Executivo, Judiciário e Legislativo, sobressaindo-se a Controladoria Geral da União – CGU, e externo, a cargo do Congresso Nacional, com o auxílio do Tribunal de Contas da União - TCU, conforme estipulado pela Constituição da República, nos artigos 71 e 74.

Algumas entidades públicas, como estatais, estão sujeitas ainda a outros controles adicionais, como o das agências reguladoras responsáveis pela área econômica em que atuam, ou de organismos internacionais que fiscalizam sua atividade, sem mencionar o crescente controle social, resultado do aumento da consciência cidadã.

Diante de todos esses controles, todos legítimos, cabe ao gestor demonstrar, de modo inequívoco, que suas contratações atenderam a todas as normas e princípios aplicáveis, entre eles a economicidade, suprimindo suas demandas ao menor custo possível, e a transparência, deixando claras todas as atividades envolvidas na escolha da proposta, sem privilegiar fornecedor ou solução específica.

Na área de *software*, ainda há dificuldades dos gestores nos processos de aquisição, sobretudo pela dificuldade de saber, com antecedência, quais são as necessidades de informação da entidade e como organizar os processos de trabalho, de modo que essa informação seja tratada adequadamente, no tocante à especificação dos sistemas, bem como a questões relacionadas à tecnologia, como ambiente e linguagem em que as

aplicações serão desenvolvidas, além da organização do processo de desenvolvimento propriamente dito, com seus custos associados.

No lado dos fornecedores, a situação não é muito melhor. Conforme Hazan (2005), a maioria das empresas de *software* ainda estimam projetos com base na opinião e no sentimento do líder, principalmente pela dificuldade em produzir as estimativas de tamanho.

Entretanto, seja por força de lei ou outro ato normativo, o gestor público é obrigado a preceder as contratações de um Projeto Básico, que deve caracterizar de forma adequada o seu objeto, isto é, os serviços a serem realizados, e identificar com clareza os seus elementos constitutivos, como determina o art. 6º, IX da Lei nº 8.666/93. No projeto, o gestor precisa ter uma estimativa do custo dos serviços previstos, para que possa decidir sobre a aceitabilidade das propostas que vier a receber, como determinam os artigos 40, X e 48 da Lei das Licitações.

Além disso, todo o processo de licitação e contratação, incluindo os relatórios de execução, deve ser devidamente alimentado, para comprovar a regularidade perante os organismos de controle e subsidiar a prestação periódica de contas.

São dois os momentos principais em que os controles interno e externo atuam, embora não haja restrições temporais rígidas para essa atividade.

O primeiro deles vai da preparação da licitação até a assinatura do contrato, verificando se o objeto está bem definido e atende ao interesse da organização, se não há cláusulas que restrinjam desnecessariamente a competitividade e se as estimativas estão consistentes, sobretudo o preço e o prazo, atividades que, na área de *software*, demandam instrumentos que permitam ao auditor avaliar se esses parâmetros são aceitáveis.

O segundo momento vai da assinatura do contrato até o término de execução contratual, para verificar se o objeto foi executado ou está em execução de forma satisfatória, atendendo às especificações do projeto básico, incluindo a conformidade com as estimativas de tamanho, esforço, prazo, recursos alocados e custo.

Essas ações podem ser associadas com os três tipos de controle no tempo: prévio, antes da ocorrência do ato; concomitante, em paralelo à execução; e *a posteriori*, após o término do ato ou processo auditados. Em geral, o mais efetivo desses controles é o prévio, já que procura impedir que o dano ocorra, corrigindo os rumos da licitação ou as cláusulas contratuais, de modo a evitar prejuízos futuros, nem sempre possíveis ou convenientes, como se vê na dificuldade de paralisação de obras públicas, mesmo após constatadas irregularidades graves.

No caso da ocorrência de ações de controle após o sistema pronto, ou mesmo após a conclusão de algumas de suas fases, com a entrega dos artefatos correspondentes, o auditor estará fazendo, nessa ótica, uma avaliação *ex post* ou exercendo o controle *a posteriori*.

Apesar da quantidade de organismos de controle existentes, pouco se tem produzido a respeito da economicidade das contratações na área de *software* e dos fatores que lhes são correlatos, como produtividade e cronogramas.

Além disso, não se tem notícia, até o momento, de bases de dados que subsidiem análises das estimativas de tamanho, prazo e custos dos projetos das contratações, para fins de *benchmark*, o que seria altamente desejável.

Gradativamente, porém, à medida que mais contratações vão sendo firmadas a partir da unidade PF, análises envolvendo essa técnica vão se fazendo presentes em relatórios de auditoria em licitações e contratos.

No Acórdão 669/2008-Plenário, o TCU determinou a uma unidade do Ministério da Educação que não exigisse ou pontuasse licitantes por qualidades não diretamente relacionadas ao objeto contratado, como a pontuação pelo desenvolvimento de sistemas acima de 8.000 PF, contrariamente ao art. 3º, § 1º, I da Lei 8.666/93.

Também recentemente, no Acórdão 443/2008, do Plenário do TCU, há informações sobre o andamento do processo de desenvolvimento de novos sistemas da previdência social. Um dos sistemas enumerados tinha estimativa de 446 PF, com a previsão de esforço de 11.150 horas, o que resulta em produtividade de 25 horas por PF, a mesma dos demais sistemas relacionados, e que é bastante elevada, especialmente levando-se em consideração que a linguagem utilizada para programação é Java, com média esperada em torno de 15 horas por PF, como visto anteriormente.

Em outro julgado do TCU, Acórdão 1.782/2007-Plenário, foi criticado o critério adotado por uma agência reguladora, que fixou o preço máximo de uma hora de desenvolvimento em R\$ 67,80, vinculado a uma tabela de produtividade por linguagem, em que figurava, por exemplo, Java, com 14,92 horas por PF. Para o desenvolvimento de 10 PF em Java, por exemplo, a remuneração seria 10 PF x 14,92 h/PF x R\$ 67,80/h, resultando em R\$ 10.115,76, o que foi considerado desvantajoso, já que poderia haver contratadas que tivessem produtividade maior que a tabelada e, portanto, consumissem menos horas por PF.

No Acórdão citado, uma das recomendações do TCU em relação à forma de remuneração consistia na utilização de tabelas com preço unitário do PF por plataforma, eliminando-se o Homem-hora e a produtividade média, estimulando a competitividade.

Essa recomendação parece compatível com o preconizado pelo ISBSG (2008), afirmando que hoje em dia não há pretexto para o gestor ficar em vôo cego: uma vez estabelecidas as características do projeto, pode ser estimada a produtividade, a partir dos dados históricos, e negociar o preço do projeto usando o PF como unidade.

Se o gestor considerar que cabe contratação direta para algum projeto de desenvolvimento específico, as estimativas precisam ser ainda mais precisas, ou é preciso haver elementos confiáveis que mostrem a compatibilidade dos preços. Pelo Acórdão 1.558/2003-Plenário, por exemplo, o TCU determinou audiência dos responsáveis e cessação do contrato de uma unidade do Ministério do Desenvolvimento, por infração ao art. 26, II da Lei 8.666/93, já que contratou por dispensa sem apresentar outras propostas que demonstrassem a economicidade da proposta.

Um outro cuidado que se tem tomado é com relação à definição clara do objeto no projeto básico. A simples menção a um dado número de PF, sem os sistemas específicos a serem desenvolvidos ou mantidos, pode indicar a terceirização disfarçada de cargos da área-fim do órgão, para fugir ao concurso público, obrigatória em regra pelo art. 37, II da Constituição, como observado no Acórdão 1.672/2006-Plenário, em que os serviços a serem desenvolvidos não estavam previstos no projeto básico.

## Dificuldades de comunicação entre os atores envolvidos

Até o presente, foram apresentadas características da APF que podem ajudar nos processos de estimativa e de medição do tamanho de um *software*, parâmetro que, juntamente com outras variáveis, pode ser utilizado para se chegar ao esforço, prazo, recursos necessários e ao custo do sistema, tarefa que pode ter diferentes propósitos. A contagem não é um fim em si mesmo, tendo sempre um propósito, como medir o serviço entregue por um fornecedor para sua posterior remuneração, conforme Vazquez (2007).

No serviço público, esse propósito pode ser, por exemplo, a elaboração de um parecer de auditoria, ou a liquidação da despesa, momento em que o fiscal do contrato atesta quanto dos serviços foram executados, para fins de pagamento ao fornecedor.

Para exercerem essas atividades reguladas na legislação, os responsáveis precisam estar devidamente capacitados, já que não cabe alegação de ignorância sobre o que é um

PF, por exemplo, para justificar pagamentos indevidos pela realização dos serviços. Em outras palavras, se pessoas atestarem recebimentos por certo número de unidades de serviços que não sabiam medir, isso constitui infração à norma legal, passível de punição.

O exemplo ilustra a necessidade de treinamento e certificação dos gestores, fiscais de contratos, auditores e fornecedores em APF, uma vez que o processo de contagem não é trivial.

Apesar disso, segundo Vazquez (2007), a APF, por usar objetos de contagem que independem da tecnologia utilizada, elimina tecnicidades e facilita a compreensão entre as partes, sendo um importante fator para melhoria da comunicação, facilitando também a comunicação efetiva com todos que possam ser afetados pelo sistema, o que é excelente em termos de melhoria da governança.

Nessa comunicação, incluem-se os organismos de controle, que podem inferir diversas características a partir da especificação inicial do tamanho de um sistema em PF.

No controle financeiro do projeto, área de intenso conflito de interesses entre fornecedor, gestor e controle, também ocorrem melhorias. Conforme o ISBSG (2008), o uso do valor unitário do PF permite integrar o ciclo de vida do desenvolvimento com os benefícios de todas as partes.

De fato, nesse caso, o fornecedor será remunerado pelos PF efetivamente entregues, ao passo que o contratante não pagará pelo tempo do serviço, mas pelo produto efetivamente gerado. Por sua vez, os auditores precisam tão-somente medir os produtos gerados e o preço cobrado pela unidade de fornecimento, o que é mais objetivo e rápido.

## A possibilidade de nova forma de contratação

A APF parece ter vindo preencher uma importante lacuna no processo de contratação de desenvolvimento de *software*, especialmente no setor público. Em Vazquez (2007), citam-se as organizações públicas regidas pela Lei 8.666/93 como exemplos em que os processos de aquisição para projetos individuais são custosos demais e, por isso, procuram contratar um grande pacote de recursos que pode ser usado conforme a demanda.

É preciso, entretanto, usar a recomendação do autor com cautela, para que não ocorra a terceirização disfarçada, já mencionada neste capítulo, criando-se contratos guarda-chuva que abrangem qualquer serviço que se queira contratar, o que é ilegal. Nada obsta, porém, o uso da APF para contratação de serviços em picos de demanda, por exemplo, ou serviços extraordinários, desde que devidamente justificados.

Na elaboração deste trabalho, foram identificados diversos editais de procedimentos licitatórios utilizando essa abordagem, como os disponíveis nas páginas do Banco Nacional de Desenvolvimento Econômico e Social – BNDES, da Companhia Nacional de Abastecimento – CONAB e da Agência Nacional de Saúde Suplementar – ANS no mês de maio de 2008<sup>6</sup>.

Antes da disseminação do PF, havia duas modalidades básicas para esse tipo de contratação: 1) por Homem-hora, ou *body shopping* ou ainda *time and material*, de grande flexibilidade, mas exigindo o gerenciamento de todo o serviço pela contratante, incluindo a produtividade dos contratados; 2) preço global fixo, privilegiando a abordagem de projeto, que não admite alterações facilmente, mas com tendência a gerar estimativas sub ou superdimensionadas, pela falta de conhecimento do fornecedor sobre o domínio do problema, além de falhas na execução por inabilidade do contratado, que pode ser percebida tarde demais, conforme Vazquez (2007).

Com a APF, porém, há uma terceira via, conforme o ISBSG (2008): a compra de *software* com base no custo por unidade funcional entregue, que pode trazer benefícios substanciais à abordagem tradicional.

Em Vazquez (2007) também são reconhecidas as vantagens da contratação por valor unitário, modelo que procuraria equilibrar as deficiências da contratação por Homem-hora e preço global fixo, mas cujo grande desafio era encontrar um elemento que pudesse ser reconhecido de maneira inequívoca, uniforme e consistente pelas partes contratantes, e que também possuísse uma natureza não excessivamente técnica.

A superação desse desafio parece possível com a criação e o aprimoramento das unidades funcionais, como o PF.

## Cláusulas recomendadas nas contratações com base na APF

Apesar de a adoção dos contratos baseados no valor unitário do PF representar um importante avanço para resguardar os interesses de ambas as partes, há necessidade de cuidados na formulação das cláusulas que vão regular a prestação dos serviços, já que os processos de estimativa e medição não são simples. Segundo Vazquez (2007), é recomendável o estabelecimento de um anexo ao contrato, esclarecendo a perspectiva do

---

<sup>6</sup> Como são documentos extensos para integrarem os anexos deste trabalho e, sendo temporários, podem ser retirados das respectivas páginas no futuro, solicita-se o encaminhamento de mensagem a [orlandoa@bol.com.br](mailto:orlandoa@bol.com.br), caso se deseje cópia desse material.

contratante quanto a alguns aspectos da técnica que podem causar diferenças nas interpretações, evitando problemas durante a execução.

Vazquez (2007) também sugere um adendo com o padrão de interfaces com o usuário, como telas e relatórios, simplificando-se a contagem futura das transações e reduzindo-se as divergências de contagem, além de exemplos para a identificação e contagem dos grupos de dados e informações de controle mantidos ou referenciados pelo sistema, já que bons exemplos valem mais do que muitas especificações.

A qualificação do futuro contratado é bastante importante, tendo em vista que projetos baseados, em PF, em geral demandam maiores esforços na área de gestão de projetos. Para Vazquez (2007), é necessária capacidade compatível com a estimativa da demanda em termos de PF, exigência que deve constar como um dos requisitos para a habilitação técnica. Além disso, recomenda-se o estabelecimento de metas de qualidade, como a quantidade de defeitos por unidade de produto entregue.

Um ponto que vem causando problemas na prática é a falta de especificação clara de quais etapas do ciclo de desenvolvimento serão realizadas, já que, por vezes, pode ter sido realizada a análise de requisitos previamente à contratação, ou será necessário apenas a codificação e os testes.

O usual nesses casos é estabelecer o valor do PF cheio, como mencionado no Capítulo 1, cobrando-se percentuais em função das etapas realizadas, com base nas ordens de serviço. Mas se isso não ficar claro desde o edital da licitação, a proposta do fornecedor pode trazer um preço não adequado aos serviços prestados, excessivamente alto ou baixo, gerando insatisfação para uma ou ambas as partes.

Em Vazquez (2007), consta ainda que o preço irá variar quando houver necessidade de muitos documentos acompanhando o *software*, como modelo UML, manual de usuário e ajuda *on-line*, preocupação que também consta em Jones (2007). É importante, portanto, que a documentação necessária, mesmo quando não extrapolar os limites usuais das exigências de mercado, conste expressamente do edital e do contrato.

Outra questão ocorre quanto ao uso do valor cheio do PF para operações que demandam menos esforço que o usual, o que não é recomendado. Vazquez (2007) observa que a produtividade na alteração e exclusão de funcionalidades é superior à observada na inclusão, devendo haver distinção no preço, mencionando exemplo de contrato em que a quantidade de horas associadas a um PF incluído foi 67% maior que a de um PF alterado ou excluído.

Quanto ao fator de ajuste da APF, quando a organização decidir utilizá-lo, a recomendação de Vazquez (2007) é que essa informação esteja expressa no edital, caso já se disponha de informações suficientes para a sua determinação, de modo a ajudar o fornecedor a definir melhor o preço, além de eliminar essa possível fonte de divergência futura.

Por fim, é conveniente lembrar que nenhuma técnica elimina a necessidade de cuidado na definição do objeto. Como observa o ISBSG (2008), nenhuma cláusula com penalidades será capaz de cuidar de especificação de requisitos incompleta, problemas de escopo e arquitetura errada, já que será, em geral, impossível estabelecer quem é o responsável pelas falhas.

### Avaliação da conformidade dos preços, sobrepreço e superfaturamento

Auditorias de conformidade são destinadas, em geral, a determinar a aderência de atos administrativos a determinações constitucionais, legais, judiciais ou regulamentares. Quando se realizam auditorias em licitações e contratos, os preços praticados podem estar de acordo com as determinações legais, o que inclui a compatibilidade com os preços de mercado, ou podem violar alguma norma.

Quando os preços forem superiores aos aceitáveis, pode-se estar diante de hipóteses de sobrepreço ou superfaturamento. Basicamente, ocorre sobrepreço quando existe licitação ou contrato com preço acima do esperado, porém não houve ainda desembolso financeiro indevido por parte da contratante. Caso já tenham sido realizados pagamentos a esse título, superando o preço correto, o excesso será considerado superfaturamento.

Assim, sendo o preço aceitável para um serviço R\$ 1.000.000,00, mas o preço contratado R\$ 1.200.000,00, há sobrepreço de R\$ 200.000,00, ou 20%. Se, nesse serviço, os pagamentos efetuados até a avaliação forem de R\$ 900.000,00, ainda não houve superfaturamento, o que ainda poderá ocorrer no curso da execução, caso o serviço ainda não tenha sido completado.

Como já mencionado, não se tem ainda notícia de mecanismo apto a determinar o preço unitário de um PF, por falta de dados históricos do mercado nacional.

Espera-se, contudo, mudança nessa realidade do setor público, a partir do Acórdão 371/2008 do Plenário do TCU, recomendando à Secretaria de Logística e Tecnologia da Informação do Ministério do Planejamento - SLTI/MP que:

priorize a construção de um repositório de editais e contratos em formato texto completo e padronizado, com ferramenta de pesquisa textual eficaz e eficiente, que facilite a recuperação de especificações semelhantes e propicie publicidade e transparência ao processo, constituindo uma base de referência aos gestores da Administração Pública que contribua para a eficiência e economicidade na gestão pública e favoreça a fiscalização da aplicação dos recursos

Essa ação, voltada para as licitações em geral, está alinhada com outro marco importante: a publicação, em 20/5/2008, da Instrução Normativa SLTI n.º 4, em atendimento a determinações e recomendações do Plenário do TCU, constantes dos Acórdãos 786/2006 e 1480/2007.

Entre outras mudanças, o art. 14 dessa Instrução Normativa preconiza que a aferição de esforço por meio da métrica homens-hora só poderá ser utilizada mediante justificativa e sempre vinculada à entrega de produtos, de acordo com prazos e qualidade previamente definidos, privilegiando assim as métricas voltadas para o produto efetivamente entregue ao destinatário, como o PF.

Conforme foi visto, levando-se em consideração as variáveis que contribuem para a formação do preço do ponto de função, é importante que as bases contenham informações relevantes a esse respeito, como o tamanho do *software* em unidades comparáveis, o prazo, o ambiente, os produtos e subprodutos a serem gerados e o escopo de desenvolvimento, de forma que possam servir como referência para as contratações futuras.

Até o momento, porém, a melhor estratégia para chegar a pareceres confiáveis nos relatórios de auditoria é coletar informações suficientes para determinar o método de precificação do licitante, observando-se também se os processos licitatórios tiveram alta competitividade, não se admitindo condições restritivas à participação do maior número possível de licitantes idôneos a realizar os serviços.

Como orientação para cumprimento do disposto no art. 40, § 2º, II, da Lei das Licitações, determinando que o edital tenha como anexo o orçamento estimado em planilhas de quantitativos e preços unitários, recomenda-se a adoção de modelos tão analíticos quanto possíveis, não se limitando a pesquisas de mercado que contenham apenas o preço final do PF.

A partir do tamanho estimado do sistema a ser desenvolvido e suas especificações, podem ser estimados duração, esforço e cronograma de alocação dos membros da equipe ao projeto, item que constitui normalmente o maior dos custos, já que a indústria do *software* é intensiva em trabalho.

O cronograma de alocação da equipe permitirá ao licitante orçar o custo direto, mas outros itens serão normalmente considerados para a formação do preço da proposta, como material aplicado no projeto, despesas indiretas fixas, como administração, aluguel e telefone e variáveis, e despesas indiretas variáveis, como tributos e comissões.

Além desses itens, o proponente em geral deveria indicar as margens reservadas para os riscos associados ao projeto e o lucro.

Em linhas gerais, acredita-se que os aspectos abordados neste trabalho, além de delinearem as razões da complexidade do processo de avaliação de preços em auditorias de licitações e contratos, permitiram também definir uma metodologia para a realização de trabalhos com essa finalidade, distinguindo-se os dois momentos principais, associados aos processos de estimativa e medição do *software*.

Usualmente, trabalhos de auditoria são realizados a partir de um instrumento metodológico básico, denominado “Matriz de Planejamento”, que baliza as ações a serem utilizadas durante a fase de execução.

No Anexo 2, apresentam-se dois modelos de matriz de planejamento para as auditorias de licitações e contratos, destinadas a avaliar sobrepreço ou superfaturamento em projetos de *software*. O primeiro deles destina-se a subsidiar a auditoria a partir dos parâmetros estimados, como foi discutido no Capítulo 1 desta pesquisa. O segundo pretende fornecer elementos para que se determine se os serviços efetivamente prestados corresponderam aos requisitos de conformidade com padrões esperados, especialmente em relação a prazo e custo.

No Anexo 3, sugere-se um modelo de planilha de projeto, de modo a facilitar a comparação entre as propostas e evitar problemas futuros relacionados ao entendimento dos termos do edital, além de insumo para a análise a ser empreendida na auditoria.

## CONCLUSÃO

Este trabalho teve como objetivo principal descrever como a APF pode auxiliar o auditor governamental de licitações e contratos na validação das estimativas, medições e avaliações de preços em desenvolvimento de *software*, desenvolvendo uma metodologia para realização desses trabalhos, sendo o resultado desse esforço consolidado nas duas Matrizes de Planejamento apresentadas no Anexo 2, destinadas a subsidiar as auditorias realizadas sobre as estimativas ou os produtos entregues.

Foram também delimitados 3 objetivos específicos.

O primeiro deles foi desenvolvido no Capítulo 1 e visava descrever os benefícios e limitações do uso da técnica de APF para a validação das estimativas de tamanho, realizadas sobre as especificações de um sistema a ser desenvolvido. A APF, associada a outras técnicas que usam o tamanho do *software* como insumo básico, revelou-se bastante útil nesse processo, permitindo a detecção de inconsistências desde as fases iniciais.

O próximo objetivo visava descrever as modalidades de uso da APF para a análise da compatibilidade dos preços finais dos sistemas entregues, de modo a permitir uma análise sobre o sistema efetivamente gerado após a contratação, tarefa empreendida no Capítulo 2. Foi evidenciado que, embora não se trate de tarefa trivial, a contagem dos pontos de função de um dado sistema permite avaliar, de modo sistemático, o objeto efetivamente entregue ao contratante, constituindo importante instrumento para fiscalização da execução dos contratos, por parte do gestor.

O terceiro e último objetivo específico pretendeu mostrar como a APF pode ser aplicada nos procedimentos de auditoria, facilitando a comunicação entre os órgãos de controle, os gestores públicos e as empresas desenvolvedoras de *software*, de modo a se obter uma avaliação correta da adequação dos preços praticados. Esse objetivo mostrou-se bastante difícil na prática, pela falta de cultura, especialmente no setor público, de formação de bases de dados que permitam a comparação dos parâmetros relevantes de cada projeto, em nível de detalhe suficiente para permitir a avaliação dos preços, contendo informações como as sugeridas nas planilhas no Anexo 3.

Nesse sentido, espera-se que a publicação do Acórdão 371/2008, do Plenário do Tribunal de Contas da União, recomendando a formação de bases dessa natureza, assim como a Instrução Normativa SLTI n.º 4, regulamentando as contratações de TI e privilegiando métricas baseados em produto, contribuam de modo decisivo para a alteração desse quadro.

Esta breve pesquisa indicou também que a área de métricas de *software* ainda é um campo bastante aberto a trabalhos acadêmicos úteis à área de auditoria.

Pode-se destacar uma importante contribuição para avaliação de preços, que consiste na pesquisa sobre os métodos empregados pelas empresas para a definição do preço final dos serviços relativos ao *software*. Como se viu, há uma grande variedade de componentes de custo a serem considerados, mas não se tem ainda noção do peso relativo de cada um deles na realização de atividades de uma fábrica de *software*.

Outra pesquisa também importante seria a validação dos diversos modelos paramétricos de estimativa à realidade brasileira. As muitas alternativas existentes contêm valores de coeficientes e expoentes que precisam ser ajustados a casos específicos, mas é preciso determiná-los para as contratações de segmentos específicos, como o setor público, tarefa que provavelmente será facilitada à medida que as bases de dados sejam alimentadas.

## Referências

BRASIL. Banco Nacional de Desenvolvimento Econômico e Social. *Licitações e Editais*. Disponível em <<http://www.bndes.gov.br>>. Acesso 23/6/2008.

\_\_\_\_\_. *Constituição da República Federativa do Brasil*. Brasília: Senado Federal, 1988.

\_\_\_\_\_. Instrução Normativa SLTI nº 4, de 19 de maio de 2008. Dispõe sobre o processo de contratação de serviços de Tecnologia da Informação pela Administração Pública Federal direta, autárquica e fundacional. *Diário Oficial da República Federativa do Brasil*. Brasília, 20/5/2008. Disponível em <<http://silabconsulta.planejamento.gov.br>>. Acesso 10/6/2008.

\_\_\_\_\_. Lei nº 8.666, 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. *Diário Oficial da República Federativa do Brasil*. Brasília, 22/6/1993. Disponível em <<http://www.planalto.gov.br>>. Acesso 10/5/2008.

\_\_\_\_\_. Tribunal de Contas da União. *Pesquisa de Jurisprudência*. Disponível em <<http://www.tcu.gov.br>>. Acesso 10/4/2008.

COCOMO. *Constructive Cost Models*. Disponível em <[http://sunset.usc.edu/research/COCOMOII/expert\\_cocomo/expert\\_cocomo2000.html](http://sunset.usc.edu/research/COCOMOII/expert_cocomo/expert_cocomo2000.html)>. Acesso 10/4/2008.

CONFEA. Resolução 361, de 10 de dezembro de 1991. Dispõe sobre a conceituação de Projeto Básico em Consultoria de Engenharia, Arquitetura e Agronomia. *Legislação do Sistema Confea-Crea*. Disponível em <<http://normativos.confex.org.br>>. Acesso 10/5/2008.

FERREIRA, Fernando. *WinFPA*. Disponível em <<http://www.winfpa.com.br>>. Acesso 10/5/2008.

HAZAN, Claudia; FUKS, Hugo; LUCENA, Carlos J. P. *Avaliação do Tamanho Funcional de Ferramentas de E-learning*. Monografias em Ciência da Computação n.º 16/05, Departamento de Informática. Rio de Janeiro: PUC, 2005.

HAZAN, Claudia; STAA, Arndt v. *Análise e Melhoria de um Processo de Estimativas de Tamanho de Projetos de Software*. Monografias em Ciência da Computação n.º 04/05, Departamento de Informática. Rio de Janeiro: PUC, 2005.

IFPUG. *Function Point Counting Practices Manual*. Versão 4.2.1. New Jersey: International Function Point Users Group, 2004.

ISBSG. *Software Development Projects in Government - performance, practices and predictions*. Disponível em <<http://www.isbsg.org>>. Acesso em 25/5/2008.

LAATUK. *Effort Estimation Tools*. Disponível em <[http://www.laatuk.com/tools/effort\\_estimation\\_tools.html](http://www.laatuk.com/tools/effort_estimation_tools.html)>. Acesso 14/5/2008.

JONES, Capers. *Estimating Software Costs - Bringing Realism to Estimating*. 2. ed. Nova Iorque: Mc Graw Hill, 2007.

NESMA. *Early Function Point Counting*. Disponível em <<http://www.nesma.nl/english/earlyfpa.htm>>. Acesso 16/3/2008.

PARTHASARATHY, M. A. *Practical Software Estimation: function point methods for insourced and outsourced projects*. Nova Iorque: Addison Wesley, 2007.

SOMMERVILLE, Ian. *Engenharia de Software*. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

VAZQUEZ, Carlos E.; SIMÕES, Guilherme S.; ALBERT, Renato M. *Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software*. 7. ed. Revisada. São Paulo: Érica, 2007.

### **Obras Consultadas**

ANDRADE, Edméia. *Pontos de Casos de Uso e Pontos de Função na gestão de estimativa de tamanho de projetos de software orientados a objetos*. 2004. Dissertação (Mestrado em Gestão do Conhecimento e Tecnologia da Informação) - UCB, Brasília.

LACERDA, Isac M., ALBUQUERQUE, Jones O. *Uma experiência na Medição Automática de Pontos de Função*. ProQuality - Qualidade na Produção de Software - vol. 1, n. 1 - Lavras: Universidade Federal de Lavras, 2005.

PRESSMAN, Roger S. *Software engineering: a practitioner's approach*. 5. ed. Nova Iorque: Mc Graw Hill, 2001.

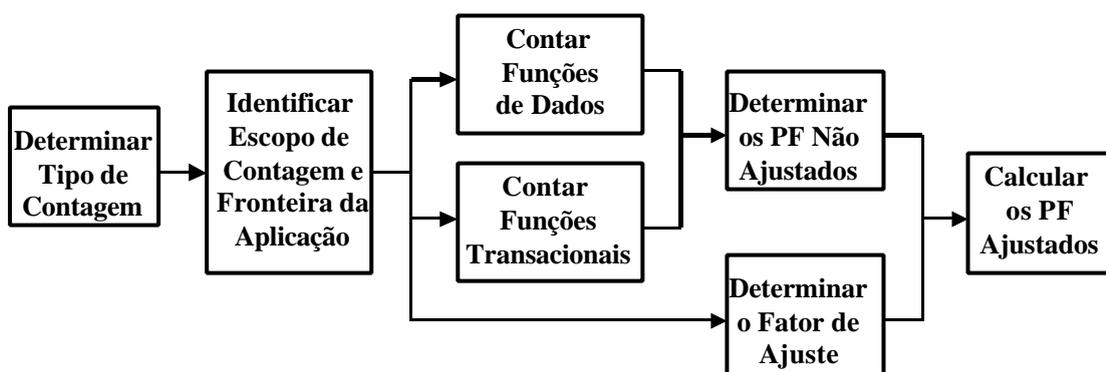
VIEIRA, Everton Luiz. *Uso do Conceito de Passos Obrigatórios para Aprimorar o Processo de Contagem do Método "Pontos de Caso de Uso"*. 2007. Dissertação (Mestrado em Ciência da Computação) - UFSC, Florianópolis.

## **ANEXOS**

## Anexo 1 – Descrição das Práticas de Contagem do IFPUG

Como se verificou nos capítulos deste trabalho, chega-se ao número de Pontos de Função ajustados de um dado sistema ao final de um processo de sete etapas. Este processo é o procedimento de contagem de Pontos de Função descrito no CPM (Figura 5). As etapas do procedimento de contagem são descritas em linhas gerais a seguir. Maiores detalhes podem ser obtidos no CPM 4.1.2 .

Figura 5 – Procedimento de Contagem de Pontos de Função



(IFPUG, 2004, p. 2-3)

### Determinar o Tipo de Contagem

O primeiro passo na contagem de Pontos de Função é determinar o tipo de contagem a ser realizada, já que ela pode ser associada a projetos ou aplicações. Existem três tipos de contagens de Pontos de Função:

- ? Contagem de Pontos de Função de Projetos de Desenvolvimento
- ? Contagem de Pontos de Função de Aplicações Instaladas
- ? Contagem de Pontos de Função de Projetos de Manutenção

### Identificar o escopo da contagem e a fronteira da aplicação

O escopo de contagem define a funcionalidade que será incluída em uma contagem de Pontos de Função específica. A fronteira indica o limite lógico entre a aplicação que está sendo medida, o usuário e outras aplicações que interagem com a aplicação em

medição. O escopo da contagem pode abranger mais de uma aplicação. A contagem de Pontos de Função sempre é realizada levando-se em consideração uma fronteira de aplicação. Assim, cada aplicação deve ser contada separadamente.

### Contar Funções de Dados (ALI e AIE)

As funções de dados correspondem aos Arquivos Lógicos Internos e Arquivos de Interface Externa.

Um arquivo lógico interno (ALI) é um grupo de dados ou de informações de controle logicamente relacionados, reconhecido pelo usuário, mantido dentro da fronteira da aplicação. A intenção primária de um ALI é armazenar dados mantidos através de um ou mais processos elementares da aplicação que está sendo contada.

Um arquivo de interface externa (AIE) é um grupo de dados ou de informações de controle logicamente relacionados, reconhecido pelo usuário, referenciado pela aplicação, porém mantido dentro da fronteira de uma outra aplicação. A intenção primária de um AIE é armazenar dados referenciados através de um ou mais processos elementares dentro da fronteira da aplicação que está sendo contada. Isto significa que um AIE contado para uma aplicação deve ser um ALI em outra aplicação.

Após, reconhecida uma função de dados como ALI ou AIE, é necessário atribuir-lhe uma complexidade, que determinará o número de pontos de função não ajustados correspondentes. Essa complexidade é determinada pelo número de Tipos de Dados Elementares (DER) e de Tipos de Registros Elementares (RLR) associados.

Um DER é um campo único, reconhecido pelo usuário e não repetido, ao qual aplicam-se as seguintes regras:

1 - Contar um DER para cada campo único, reconhecido pelo usuário e não repetido, mantido ou recuperado de um ALI ou AIE através da execução de um processo elementar.

2 - Quando duas aplicações mantêm e/ou referenciam o mesmo ALI/AIE, mas cada uma mantêm/referencia diferentes DER, contar apenas os DER que estão sendo usados por cada aplicação para medir o ALI/AIE.

3 - Contar um DER para cada parte de dado requisitada pelo usuário para estabelecer um relacionamento com outro ALI ou AIE.

Por sua vez, um RLR é um subgrupo de dados reconhecido pelo usuário dentro de um ALI ou um AIE, distinguindo-se os subgrupos obrigatórios dos opcionais.

Subgrupos opcionais são aqueles que o usuário tem a opção de usar um ou nenhum dos subgrupos durante o processo elementar que inclui ou cria uma instância do dado, diferentemente dos subgrupos obrigatórios, em que há necessidade de uso de pelo menos um.

Uma das seguintes regras deve ser aplicada quando da contagem de RLR:

- 1 - Contar um RLR para cada subgrupo opcional ou obrigatório do ALI ou AIE
- 2 - Se não existirem subgrupos, contar o ALI ou AIE como apenas 1 RLR.

De posse dessas informações para cada ALI ou AIE, eles devem ser classificados, segundo as tabelas a seguir, atribuindo-se-lhes a complexidade baixa, média ou alta, conforme o número de DER e RLR.

RLR	DER		
	1 a 19	20 a 50	51 ou mais
1	Baixa	Baixa	Média
2 a 5	Baixa	Média	Alta
6 ou mais	Média	Alta	Alta

Por fim, chega-se ao número de pontos de função não ajustados a serem atribuídos a cada uma dessas funções, conforme a tabela a seguir:

COMPLEXIDADE	PONTOS DE FUNÇÃO NÃO AJUSTADOS	
	ALI	AIE
<b>Baixa</b>	7	5
<b>Média</b>	10	7
<b>Alta</b>	15	10

### Contar Funções Transacionais (EE, SE e CE)

As funções transacionais são três: Entradas Externas (EE), Saídas Externas (SE) e Consultas Externas (CE). A noção dessas funções depende do conceito de processo elementar: a menor unidade de atividade que tem significado para o usuário, que é autocontida e deixa o negócio da aplicação em um estado consistente.

Assim, entrada externa (EE) é um processo elementar que processa dados ou informações de controle que vêm de fora da fronteira da aplicação, tendo como objetivo principal manter um ou mais ALI e/ou alterar o comportamento do sistema.

Saída externa (SE) é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação, tendo como principal objetivo apresentar informações ao usuário por meio de lógica de processamento que pode incluir, ou não, a recuperação de dados ou informações de controle. O processamento lógico deve conter pelo menos uma fórmula matemática ou cálculo, criar dados derivados, manter um ou mais ALI ou alterar o comportamento do sistema.

Por fim, consulta externa (CE) é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação, para apresentar informações ao usuário pela recuperação de dados ou informações de controle de um ALI ou AIE. O processamento lógico não deve conter fórmulas matemáticas ou cálculos, nem criar dados derivados. Além disso, nenhum ALI é mantido durante o processamento e nem o comportamento do sistema é alterado.

A contagem das funções transacionais é realizada em 5 etapas, a saber: identificar os processos elementares; determinar o objetivo principal desses processos, classificando-os como EE, SE ou CE; validar, segundo as regras de identificação de EE, SE e CE; determinar a complexidade das transações; determinar a contribuição das transações para a contagem de pontos de função não ajustados.

Para a classificação das funções transacionais em EE, SE ou CE, várias regras são aplicadas.

Uma EE deve atender a várias condições. Os dados ou informações de controle são recebidos de fora da fronteira da aplicação. Pelo menos um ALI deve ser mantido, se os dados que entram pela fronteira não forem informações de controle que alterem o comportamento do sistema. E para a identificação da unicidade da EE, uma das três afirmações seguintes deve ser aplicável: a) a lógica de processamento é diferente das lógicas executadas em outras entradas externas da aplicação; b) o grupo de dados identificado é diferente dos grupos identificados em outras entradas externas da aplicação; c) os ALI ou AIE referenciados são diferentes dos arquivos referenciados em outras entradas externas da aplicação.

Há também regras comuns para determinar o enquadramento de processos elementares como SE e CE, sendo ambas funções que enviam dados ou informação de controle para fora da fronteira da aplicação. Para a identificação da unicidade da função,

uma das três afirmações seguintes deve ser aplicável: a) a lógica de processamento é diferente das executadas em outras SE ou CE da aplicação; b) o grupo de dados identificado é diferente dos grupos identificados em outras SE ou CE da aplicação; c) os ALI ou AIE referenciados são diferentes dos arquivos referenciados em outras SE ou CE da aplicação.

Para ser considerado uma SE, além das regras mencionadas no parágrafo anterior, o processo deve atender às seguintes: a) A lógica de processamento do processo elementar contém pelo menos uma fórmula matemática ou cálculo; b) a lógica de processamento do processo elementar cria dados derivados; c) a lógica de processamento do processo elementar mantém pelo menos um ALI; d) a lógica de processamento do processo elementar altera o comportamento do sistema.

Por sua vez, para ser considerado uma CE, as seguintes regras devem ser validadas: a) a lógica de processamento do processo elementar recupera dados ou informações de controle de um ALI ou AIE; b) a lógica de processamento do processo elementar não contém fórmulas matemáticas ou cálculos; c) a lógica de processamento do processo elementar não cria dados derivados; d) a lógica de processamento do processo elementar não mantém nenhum ALI; e) a lógica de processamento do processo elementar não altera o comportamento do sistema.

Identificada uma função transacional, aplicam-se as regras para a sua contagem, que dependem do conceito de DER, já mencionado, e no de Arquivo Lógico Referenciado (ALR).

No CPM, o conceito de ALR é definido de forma alternativa, como um ALI lido ou mantido por uma função de transação ou um AIE lido por uma função de transação.

São contados inicialmente os ALR e DER, segundo regras variáveis para cada tipo de função transacional.

As seguintes regras de contagem são aplicadas para as EE.

Conta-se um ALR para cada ALI mantido, ou para cada ALI ou AIE lido durante o processamento da EE. Se um ALR for lido e mantido, contar apenas uma vez.

Em relação aos DER, conta-se um para cada campo não repetido, reconhecido pelo usuário que entre ou saia da fronteira da aplicação e que seja necessário para concluir a EE. Também deve ser contado para indicar a capacidade de enviar uma mensagem de resposta do sistema para fora da fronteira, indicando um erro ocorrido durante o processamento, confirmar que o processamento está concluído ou que deverá prosseguir. Por fim, o manual também menciona que deve ser contado um DER para a habilidade de especificar uma

ação a ser executada, mesmo existindo vários métodos para chamar o mesmo processo lógico.

Conta-se um ALR para cada ALI ou AIE lido durante o processamento. No caso de SE, conta-se também um ALR para cada ALI mantido, contando-se apenas um no caso de leitura e manutenção.

Para os DER, são aplicadas as seguintes regras: a) conta-se um DER para cada campo não repetido, reconhecido pelo usuário que entre na fronteira da aplicação e seja necessário para especificar quando, o que e/ou como os dados deverão ser recuperados ou gerados pelo processo elementar; b) conta-se um DER para cada campo não repetido, reconhecido pelo usuário que sai da fronteira; c) se um DER entrar e sair da fronteira, deve ser contado apenas uma vez no processo elementar; d) conta-se um DER para a capacidade de enviar uma mensagem de resposta do sistema para fora da fronteira da aplicação, para indicar um erro ocorrido durante o processamento, confirmação de que o processamento está concluído, ou confirmação de que o processamento deverá continuar; e) conta-se um DER para a habilidade de especificar uma ação a ser executada, mesmo que existam vários métodos para chamar o mesmo processo lógico.

O manual é bastante detalhado, ilustrando cada situação com exemplos e contra-exemplos práticos.

Contados os ALR e DER de cada processo elementar, aplica-se procedimento semelhante ao descrito para a contagem de ALI e AIE, a partir dos dados das tabelas seguintes para determinar o número de pontos de função não ajustados a cada EE, SE e CE identificada.

#### EE

RLR	DER		
	1 a 4	5 a 15	16 ou mais
0 a 1	Baixa	Baixa	Média
2	Baixa	Média	Alta
3 ou mais	Média	Alta	Alta

## SE ou CE

RLR	DER		
	1 a 5	6 a 19	20 ou mais
0 a 1	Baixa	Baixa	Média
2 a 3	Baixa	Média	Alta
4 ou mais	Média	Alta	Alta

De posse da complexidade, a leitura da tabela seguinte permite determinar o número de pontos de função não ajustados a serem atribuídos a cada uma dessas funções.

COMPLEXIDADE	PONTOS DE FUNÇÃO NÃO AJUSTADOS	
	EE ou CE	SE
<b>Baixa</b>	3	4
<b>Média</b>	4	5
<b>Alta</b>	6	7

## Determinar os Pontos de Função Não Ajustados

Esta etapa consiste no somatório das contribuições funcionais (número de PFs não ajustados) de todas as funções de dados e funções transacionais da contagem em questão. Muitas empresas automatizam este passo com planilhas de cálculo ou ferramentas de contagem de Pontos de Função.

## Determinar o Fator de Ajuste

O Fator de Ajuste indica as funcionalidades gerais fornecidas ao usuário da aplicação, levando em consideração as 14 Características Gerais do Sistema (CGS), que avaliam as funcionalidades gerais da aplicação.

Cada característica tem descrições associadas que ajudam a determinar o nível de influência da mesma. Os níveis de influência variam numa escala de 0 a 5, correspondendo 0 a nenhuma influência e 5 a influência máxima, conforme ilustra a tabela seguinte.

Pontuação	Nível de influência
0	Não presente ou sem influência
1	Influência Mínima
2	Influência Moderada
3	Influência Média
4	Influência Significativa
5	Forte influência

Para determinação do fator de ajuste, são somados os níveis de influência de cada uma das 14 CGS, sendo o resultado multiplicado por 0,01 e, a seguir, somado ao coeficiente 0,65.

Assim, uma aplicação que tenha obtido nível de influência igual a 2 para todas as CGS teria fator de ajuste igual a  $0,65 + 0,01 \times 14 \times 2 = 0,65 + 0,28 = 0,93$  ou 93%. Como esse fator é aplicado ao número de pontos de função não ajustados da aplicação, o resultado 93% indica que o tamanho ajustado é menor do que o não ajustado.

Ao contrário, se todos os níveis de influência de uma aplicação fossem pontuados com 4, indicando influência significativa, aos pontos não ajustados seria aplicado um fator igual a  $0,65 + 0,01 \times 14 \times 4 = 1,21 = 121\%$ , indicando que a contagem de Pontos de Função Ajustados será 21% maior que a contagem de Pontos de Função não Ajustados,.

Por fim, os valores extremos para os pontos de função ajustados correspondem aumento ou redução de 35% em relação à contagem padrão de 100% dos pontos de uma aplicação. Sendo o nível de influência zero para todas as CGS, o fator será  $0,65 + 0,01 \times 14 \times 0 = 65\%$  e, sendo o máximo de 5, o fator será  $0,65 + 0,01 \times 14 \times 5 = 135\%$ .

A tabela seguinte ilustra a contribuição de cada uma das CGS para o resultado final da contagem.

Descrição da CGS	Pontuação	Mínimo	Máximo
Comunicação de Dados	0-5	0	5
Processamento Distribuído	0-5	0	5
Desempenho	0-5	0	5
Configuração de Uso Intenso	0-5	0	5
Volume de Transações	0-5	0	5
Entrada de Dados On-Line	0-5	0	5
Eficiência do Usuário Final	0-5	0	5
Atualização <i>On-Line</i>	0-5	0	5
Processamento Complexo	0-5	0	5
Reusabilidade	0-5	0	5

Facilidade de Instalação	0-5	0	5
Facilidade de Operação	0-5	0	5
Múltiplos Locais	0-5	0	5
Facilidade de Mudança	0-5	0	5
Grau Total de Influência (TDI)		0	70
Fator = (TDI x 0,01) +0,65		0,65	1,35

### Calcular os Pontos de Função Ajustados

O resultado final do processo de medição dependerá do tipo de contagem que for realizada, como foi observado no Capítulo 2. A seguir são apresentadas as formulas para cada tipo de contagem.

✍ Para projetos de desenvolvimento

$$\mathbf{DFP = (UFP + CFP) \times VAF,}$$

sendo UFP e CFP, respectivamente, o número de PF não ajustados correspondentes às funções disponíveis após a instalação e às funções de conversão de dados, e VAF o fator de ajuste

✍ Para projetos de melhoria (enhancement)

$$\mathbf{EFP = [(ADD + CHGA + CFP) \times VAFA] + (DEL \times VAFB),}$$

sendo ADD o número de PF não ajustados das funções incluídas pelo projeto de melhoria; CHGA, o número de PF não ajustados, após os serviços, das funções modificadas; DEL, o número de PF não ajustados das funções excluídas; VAFA é o fator de ajuste depois do projeto; VAFB é o fator de ajuste antes do projeto

✍ Para projetos de aplicação, há 2 fórmulas disponíveis

$$\mathbf{AFP = ADD \times VAF,}$$

para a contagem inicial, sendo ADD o número de PF das funções instaladas

$$\mathbf{AFP = [(UFPB + ADD + CHGA) - (CHGB + DEL)] \times VAFA,}$$

para a contagem após projetos de melhoria, sendo UFPB o número de PF não ajustados antes do projeto; ADD, o número de PF das funções incluídas; CHGA, o número de PF, após o projeto, das funções alteradas; CHGB, o número de PF, antes do projeto, das funções alteradas; DEL, o número de PF das funções excluídas; VAFA é o fator de ajuste após o projeto

Como se viu, o processo de medição é trabalhoso, apesar de suficientemente detalhado, o que leva muitos usuários a criarem planilhas eletrônicas para automatização dos cálculos envolvidos.

## Anexo 2

### MATRIZ DE PLANEJAMENTO PARA AUDITORIA DE LICITAÇÕES

**Objetivo:** avaliar preços estimados, constantes do projeto básico, para desenvolvimento e manutenção evolutiva de software

Qn	QUESTÃO DE AUDITORIA	INFORMAÇÕES REQUERIDAS	FONTES DE INFORMAÇÃO	TÉCNICAS DE AUDITORIA	LIMITAÇÕES	O QUE A ANÁLISE VAI PERMITIR DIZER
01	O objeto da licitação e o método de estimativa de preços estão definidos de forma suficientemente clara no projeto básico, para não gerar dúvidas a gestores, potenciais licitantes e órgãos de controle?	Projeto básico, com descrição detalhada do objeto da licitação  Preço estimado para a contratação do objeto da licitação	Documentos relativos às fases interna e externa da licitação, incluindo pesquisas de mercado, memórias de cálculo, editais e anexos	Exame documental  Entrevistas  Pesquisa de mercado		Se ocorreu: - ausência de projeto básico; - definição de objeto genérico ou impreciso; - ausência ou insuficiência de estimativa de quantitativo e preços unitários em planilhas
02	Qual o tamanho funcional estimado do sistema referente ao projeto de desenvolvimento ou manutenção previsto na licitação?	Projeto básico	CPM IFPUG  Modelo de Desenvolvimento de Sistemas	Exame documental  Entrevistas	Insuficiência de especificações  Ausência de estimativa de tamanho	Se o tamanho estimado pelo gestor está compatível com as especificações do objeto contidas no edital
03	O prazo previsto no edital para a entrega dos serviços e vigência do contrato é factível?	Tamanho funcional estimado Esforço estimado Cronograma de atividades previsto	Estimativas do gestor e da equipe de auditoria Documentos da licitação	Exame documental	Ausência do cronograma de atividades previsto	Se ocorreu: - super ou subestimação do esforço e/ou do prazo, com repercussões no dimensionamento dos preços; - ausência de planejamento para a execução das atividades relativas ao contrato
04	O preço estimado para o contrato é compatível com o objeto da licitação?	Projeto básico Tamanho, esforço e prazo estimados	Documentos da licitação Estimativas da equipe de auditoria	Exame documental Pesquisa de mercado	Dificuldade de levantamento no mercado	Se o preço unitário do ponto de função, encontrado pelo quociente entre o valor estimado do contrato e o tamanho estimado do <i>software</i> é compatível com o mercado

## MATRIZ DE PLANEJAMENTO PARA AUDITORIA DE EXECUÇÃO DE CONTRATOS

*Objetivo: avaliar preços praticados em contratações, para desenvolvimento e manutenção evolutiva de software*

<b>Qn</b>	<b>QUESTÃO DE AUDITORIA</b>	<b>INFORMAÇÕES REQUERIDAS</b>	<b>FONTES DE INFORMAÇÃO</b>	<b>TÉCNICAS DE AUDITORIA</b>	<b>LIMITAÇÕES</b>	<b>O QUE A ANÁLISE VAI PERMITIR DIZER</b>
01	O sistema objeto do desenvolvimento ou da manutenção evolutiva corresponde às condições previstas no edital da licitação?	Projeto básico original Especificações finais do produto implementado	Documentos relativos às fases interna e externa da licitação Operação do sistema	Exame documental Exame direto sobre o sistema Entrevistas	Condições do edital insuficientes ou ambíguas	Se ocorreu: - alteração não justificada entre a solução proposta e a implementada; - realização de serviços não cobertos contratualmente
02	Qual o tamanho funcional medido dos serviços realizados?	Especificações finais do produto implementado	CPM IFPUG Modelo de Desenvolvimento de Sistemas	Exame documental Exame direto sobre o sistema Entrevistas	Insuficiência de documentação dos serviços executados	Se o tamanho funcional apresentou variação não justificada em relação ao estimado no projeto básico.
03	O prazo final para a entrega dos serviços foi compatível com o compromisso assumido no contrato?	Tamanho funcional medido Esforço realizado Cronograma de atividades	Documentos comprobatórios das solicitações de serviço e relatórios de execução	Exame documental	Problemas não documentados no fluxo de trabalho	Se ocorreu: - no caso de contratação por homem-hora, realização de esforço não compatível com o produto gerado; - descumprimento de prazo por falta de alocação tempestiva dos recursos necessários
04	O valor final do contrato é compatível com o objeto da licitação?	Projeto básico Tamanho, esforço e prazo e preço medidos	Documentos da licitação Cálculos da equipe de auditoria	Exame documental Pesquisa de mercado	Dificuldade de levantamento no mercado	Se o preço unitário do ponto de função é compatível com o mercado Se o número de pontos de função faturados correspondem ao medido

**Anexo 3 – Planilhas de detalhamento dos serviços**  
(uma planilha para cada grupo de serviços semelhantes)

1) Caracterização dos serviços a serem fornecidos em pontos de função

Número de PF não ajustados	Fator de Ajuste (1,00 se não utilizado)	Tipo de Serviço (desenvolvimento, manutenção evolutiva)
Metodologia de Desenvolvimento	Linguagem de programação	Taxa de entrega (horas por PF)
Produtividade (PF / mês)	Jornada de trabalho da equipe (horas por mês)	Esforço estimado (pessoas-hora)
Prazo para conclusão (meses)	Tamanho da equipe alocada (pessoas)	Custo da mão-de-obra (R\$)
Distribuição estimada percentual de esforço entre as fases do projeto		
Fase 1	% do esforço total da fase 1...	
Fase 2...	% do esforço total da fase 2...	

## 2) Cronograma de alocação da força de trabalho ao projeto

Mês em relação ao início	Horas de profissionais tipo 1	Horas de profissionais tipo 2	...
01			
02			
...			

## 3) Cálculo do preço final

Componente	Valor (R\$)	Percentual sobre o Custo Direto (CD)
Custo da mão-de-obra		
Material aplicado		
Custo Direto		100%
Despesas fixas		
Despesas variáveis		
Risco do projeto		
Lucro Líquido		
Preço final		

## 4) Preço unitário do ponto de função

Preço final (R\$)	Número de PF	Preço unitário (R\$/PF)