

Como evitar armadilhas em contratos de fábricas de *software*

Claudia Hazan



Claudia Hazan é funcionária do Serviço Federal de Processamento de Dados (SERPRO). Graduada em Informática pela Universidade do Estado do Rio de Janeiro (UERJ), mestre em Engenharia de Sistemas e Computação/Qualidade de *Software* pelo Instituto Militar de Engenharia (IME), especialista em Análise de Pontos de Função com Certificação CFPS desde 2001.

1 INTRODUÇÃO

A Tecnologia da Informação tem sido utilizada em vários segmentos do mercado na automatização de processos, visando o aumento da eficiência e eficácia dos processos organizacionais. Pode-se observar o aumento da demanda de desenvolvimento de novos sistemas e de manutenção dos sistemas existentes para que estes se adaptem às novas necessidades dos seus usuários. Nesse contexto, as áreas de Tecnologia da Informação de muitas organizações governamentais precisam contratar serviços de desenvolvimento e manutenção de sistemas externamente para suprir a demanda das áreas de negócios do Órgão.

Os principais tipos de contratos de prestação de serviços de desenvolvimento ou manutenção de sistemas são os seguintes: Contrato por Preço Fechado, Contrato por homem-hora, Contrato baseado em Métricas de Tamanho Funcional, descritos nos parágrafos seguintes.

O Contrato por Preço Fechado consiste no contrato de um produto, por exemplo um Sistema de Recursos Humanos, com um preço definido para sua construção. Esse tipo de contrato pode ser prejudicial para a empresa contratada que precisa definir um preço para um sistema que está em fase inicial de especificação. Algumas vezes, as empresas que participam de uma licitação para este tipo contrato superestimam o do tamanho do projeto, fornecendo um preço elevado para o produto e gerando um ônus enorme para o contratante, prejudicando a economicidade do contrato.

O Contrato por homem-hora foi bastante utilizado pelos Órgãos Públicos. Nesse tipo de contrato, o pagamento é realizado pela hora de trabalho do profissional contratado. Assim, o Órgão Público estima uma quantidade de horas para o desenvolvimento ou manutenção de sistemas e as empresas que participam da licitação fornecem o

preço da hora. Na verdade, trata-se de um contrato de terceirização de mão de obra e não de Fábrica de *software*, trazendo muitos riscos gerenciais e trabalhistas da empresa contratada para o Órgão Público contratante. Observe que se a empresa contratada alocar ao projeto de desenvolvimento de um Sistema de Recursos Humanos programadores improdutivos, que estejam aprendendo a linguagem de programação do sistema em questão, o ônus do sistema não ficar pronto será do contratante. Deve-se ressaltar que a Instrução Normativa – IN04, associada ao processo de contratação de serviços de Tecnologia da Informação pela Administração Pública Federal direta, autárquica e fundacional, publicada pela Secretaria de Logística e Tecnologia da Informação (SLTI) do Ministério do Planejamento Orçamento e Gestão (MPOG) em 2008, preconiza a utilização de métricas em contratos de fábrica de *software*, e ainda destaca que a aferição de esforço por meio da métrica homem-hora apenas poderá ser utilizada mediante justificativa e sempre vinculada à entrega de produtos de acordo com prazos e qualidade previamente definidos.

O tipo de contrato de desenvolvimento e manutenção de sistemas baseado em métricas de tamanho funcional tem sido utilizado por organizações governamentais desde a década de 90. Nesse tipo de contratação, o pagamento da empresa contratada é realizado por meio do dimensionamento do produto entregue, utilizando a métrica estabelecida no contrato. As métricas de tamanho mais utilizadas nos contratos são: Pontos por Casos de Uso, Pontos por Casos de Uso Ajustados, Pontos de Função Ajustados e Pontos de Função Não Ajustados.

A métrica Pontos por Casos de Uso (PCU) foi proposta por Gustav Karner (1993) com o propósito de estimar recursos para projetos de *software* orientados a objeto, modelados por meio de especificação de Casos de Uso. A métrica é de fácil aplicação, não requer muito tempo de treinamento ou experiência prática. No entanto, o PCU somente pode ser aplicado em projetos de *software* cuja especificação tenha sido documentada em casos de uso. Além disso, como não existe um padrão único para a escrita de uma especificação de caso de uso, diferentes estilos na escrita do caso de uso ou na sua granularidade podem levar a resultados diferentes na

medição por PCU. Assim, a métrica se torna subjetiva, a empresa contratada pode modelar um sistema com muitos casos de uso para maximizar a contagem de PCU, onerando o custo do projeto para a organização contratante. O Fator de Ajuste dessa métrica também é bastante subjetivo, não existindo critérios objetivos para a pontuação das características do projeto que impactam no Fator de Ajuste. Assim, a métrica PCU não é recomendada como unidade de medida para o estabelecimento de contratos de fábrica de *software*.

A métrica Pontos de Função (PF) é uma medida de tamanho funcional de projetos de *software*, considerando as funcionalidades implementadas, sob o ponto de vista do usuário. Tamanho funcional é definido como “tamanho do *software* derivado pela quantificação dos requisitos funcionais do usuário” (Dekkers, 2003). A métrica é independente da metodologia e tecnologia utilizadas, levando em consideração a visão do usuário. O Fator de Ajuste da métrica PF possui critérios objetivos de pontuação. No entanto, as características do projeto utilizadas na determinação do Fator de Ajuste, constituem requisitos não funcionais do projeto de *software*, além disso a ISO/IEC 20926 reconheceu Pontos de Função Não Ajustados como métrica de tamanho funcional (Dekkers, 2003). Em 2009, foi publicada a versão 4.3 do Manual de Práticas de Contagem (CPM) considerando os Pontos de Função Não Ajustados como a métrica padrão do IFPUG (International Function Point Users Group) (2009). Portanto, a autora recomenda o uso da métrica Pontos de Função Não Ajustados como unidade de medida nos contratos de fábrica de *software*. É importante destacar que o Tribunal de Contas da União (TCU) também tem recomendado, por meio da publicação de Acórdãos, a utilização de Pontos de Função Não Ajustados em contratos.

Devido ao cenário apresentado acima, atualmente, a maioria dos Órgãos Públicos está utilizando a métrica de Pontos de Função Não Ajustados em seus Editais para contratação de Fábrica de *software*. Sem dúvida, é uma grande evolução em relação à década de 90, onde a maioria dos contratos era de terceirização de mão de obra por meio de homem-hora, e muitas vezes os custos dos sistemas eram muito altos em relação aos resultados obtidos. No entanto, mesmo estabelecendo contratos utilizando a métrica Pontos de Função Não Ajustados,

ainda existem muitos problemas associados aos contratos de fábrica de *software*. Esse artigo tem como propósito recomendar o uso da métrica de Pontos de Função Não Ajustados em contratos de *software* e apresentar dicas para as organizações que desejam estabelecer contratos com Fábricas de *Software* externas evitar armadilhas no processo de contratação.

Este trabalho encontra-se organizado da seguinte maneira: a Seção 1 descreve o cenário de contratação de serviços de fábrica de *software* e o objetivo deste artigo; a Seção 2 apresenta uma visão geral da Análise de Pontos de Função; a Seção 3 identifica os principais problemas e soluções de contratos de *software* baseados na métrica PF; Finalmente, a Seção 4 conclui o artigo.

2 UMA VISÃO GERAL DA ANÁLISE DE PONTOS DE FUNÇÃO

A métrica Pontos de Função foi criada por Allan Albrecht (1979) visando minimizar as dificuldades associadas ao uso da métrica Linhas de Código (LOC) como unidade de medida de tamanho de *software* e suportar a previsão do esforço de desenvolvimento do projeto de *software* (Albrecht, 2003). Em 1986, uma Pesquisa do *Quality Assurance Institute* mostrou que PF é a melhor métrica para o estabelecimento de medições de qualidade e produtividade de projetos de *software* (Perry, 1986). Em 1993, PF se tornou a métrica mais utilizada e estudada na Engenharia de *Software* (Jones, 1994). Atualmente, a métrica de PF continua sendo a mais utilizada na indústria de *software*, como métrica padrão na definição de indicadores, como insumo para derivação de estimativas de prazo, custo e esforço e no estabelecimento de contratos de *software*. Esta seção tem como propósito apresentar uma visão geral da contagem de Pontos de Função, baseando-se nas regras de contagem do *Counting Practices Manual* (CPM) 4.3 (*INTERNATIONAL FUNCTION POINT USERS GROUP*, 2009).

Os principais objetivos da Análise de Pontos de Função são os seguintes:

- Medir a funcionalidade requisitada e recebida pelo usuário;
- Medir projetos de desenvolvimento e de manutenção evolutiva de *software* independentemente da tecnologia utilizada na implementação.

A Contagem de Pontos de Função Não Ajustados consiste no mapeamento dos requisitos funcionais do projeto de *software* nos cinco tipos funcionais da Análise de Pontos de Função: Arquivo Lógico Interno (ALI), Arquivo de Interface Externa (AIE), Entrada Externa (EE), Consulta Externa (CE), Saída Externa (SE), conforme ilustrado na Figura 1.

O Arquivo Lógico Interno é um grupo lógico de dados, mantido dentro da fronteira da aplicação, por meio de um ou mais processos elementares. O Arquivo de Interface Externa é um grupo lógico de

dados, referenciado por um ou mais processos elementares da aplicação. Contudo, eles são mantidos dentro da fronteira de outra aplicação. Uma Entrada Externa é um processo elementar que processa dados ou informação de controle que vem de fora da fronteira da aplicação. O objetivo principal de uma EE é manter um ou mais ALIs ou alterar o comportamento da aplicação. Uma Saída Externa é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. O objetivo principal de uma SE é apresentar informações para o usuário por meio de um processamento lógico que deve conter: cálculos ou criar dados derivados ou manter ALIs ou alterar o comportamento da aplicação. Uma Consulta Externa é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. O objetivo principal de uma CE é apresentar informação para o usuário por meio apenas de uma recuperação de dados ou informação de controle de um ALI ou AIE.

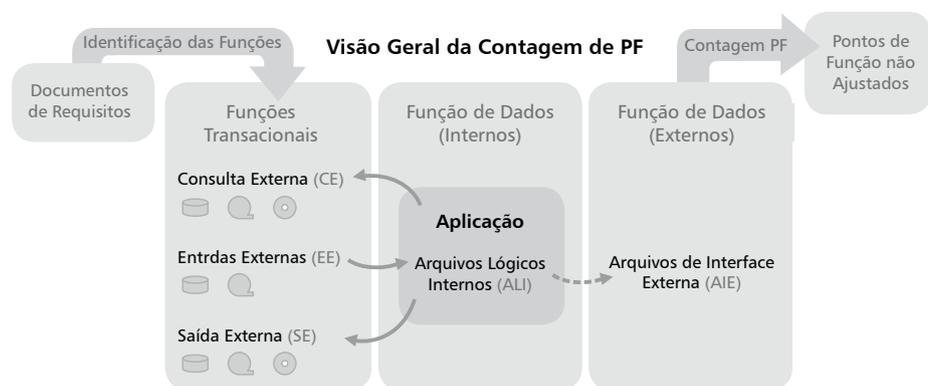


Figura 1: Os Cinco Tipos Funcionais da Análise de Pontos de Função [Hazan, 2008]

Cada função identificada possui uma complexidade associada: Simples, Média ou Complexa e uma contribuição para a contagem de Pontos de Função Não Ajustados, baseada na sua complexidade (Tabela 1). A determinação da complexidade e da contribuição funcional não é subjetiva, sendo baseada nas regras de contagem do CPM (*INTERNATIONAL FUNCTION POINT USERS GROUP*, 2009). Tabela 1: Contribuição Funcional dos Tipos de Função [IFPUG, 2009]

Tipo de Função	Simples	Média	Complexa
Arquivo Lógico Interno (ALI)	7	10	15
Arquivo de Interface Externa (AIE)	5	7	10
Entrada Externa (EE)	3	4	6
Saída Externa (SE)	4	5	7
Consulta Externa (CE)	3	4	6

A utilização do procedimento de contagem de Pontos de Função, descrito no CPM [IFPUG, 2009], implica na existência do projeto lógico da aplicação.

Nas fases iniciais do processo de desenvolvimento de *software*, os Pontos de Função não podem ser medidos e sim estimados. Sugere-se a utilização do método Contagem Estimativa de Pontos de Função (CEPF) nas estimativas de tamanho dos projetos de *software* para o estabelecimento de contratos ou elaboração de Editais [Hazan, 2005a].

3 PROBLEMAS E SOLUÇÕES DE CONTRATOS DE FÁBRICAS DE SOFTWARE

De fato, a métrica PF tem sido muito utilizada como unidade monetária (R\$/PF) nos contratos de desenvolvimento e manutenção de sistemas pelas organizações governamentais brasileiras. Este tipo de contrato permite o melhor balanceamento de riscos entre contratante e contratada (Aguiar, 2000), sendo recomendado pela Instrução Normativa - IN04 e pelos Órgãos de Controle do Governo Brasileiro. Esta seção tem como propósito apresentar algumas dicas para as organizações contratantes evitarem armadilhas em contratos de fábrica de *software* baseados em preço fixo por PF.

3.1 OBTENHA UM DOCUMENTO DE REQUISITOS DE QUALIDADE

Conforme mencionado, a métrica PF mede a funcionalidade requisitada e recebida pelo usuário. O documento de requisitos constitui um acordo comum entre as empresas contratantes e contratadas. Assim, é fundamental a existência de um “Termo de Aceite” associado aos documentos de requisitos, assinado pelo gestor do sistema ou gestor do contrato do Órgão contratante. Além disso, o contratante deve garantir a qualidade do documento de requisitos encaminhado para a fábrica de *software* contratada. Observe que se o contratante fornece um documento de requisitos com um requisito incompleto, a Fábrica entregará o produto sem a funcionalidade esperada e a organização contratante terá que pagar por isso.

A Engenharia de Requisitos apresenta várias técnicas para suportar as atividades de verificação e validação de Documentos de Requisitos, no entanto estas técnicas são muito custosas. Sugere-se a utilização do método Contagem Estimativa de Pontos de Função, além de apoiar nas estimativas do projeto, o método suporta a detecção de defeitos em documentos de requisitos pelo estimador,

enquanto ele está estimando o projeto, sem custo ou esforço adicional, conforme demonstrado por Hazan (2005a). Considerando as revisões e auditorias em Contagem de Pontos de Função dos projetos contratados, é importante que o documento de requisitos e o documento de contagem de PF ou estimativas estejam consistentes.

3.2 ESTABELEÇA REGRAS PARA O TRATAMENTO DAS MUDANÇAS DE REQUISITOS

A Engenharia de Requisitos e a indústria reconhecem que os requisitos não permanecem “congelados” até a conclusão do projeto de *software*. Os requisitos evoluem desde a sua concepção até mesmo após o sistema entrar em produção, devido a diversos fatores descritos por Kotonya (1998). Assim, é fundamental que o contrato de *software* estabeleça cláusulas para tratamento das mudanças de requisitos. Este trabalho sugere o estabelecimento no contrato de um percentual para cada atividade do processo de *software*, por exemplo: requisitos: 20%, projeto e arquitetura: 10%, implementação: 50%, Testes: 15%, implantação: 5%. Estes percentuais são hipotéticos, cada organização deve definir estes valores com base no esforço consumido em cada atividade de seu processo de desenvolvimento de *software*. Além disso, o gestor do projeto contratado deve acompanhar semanalmente o progresso de cada requisito funcional identificado no documento de requisitos do projeto de *software*. Assim, quando um requisito for modificado, deve-se identificar no Relatório de Acompanhamento do projeto quais atividades do ciclo de vida foram realizadas para o requisito em questão.

O próximo passo é obter a contagem de Pontos de Função do requisito original para aplicar o percentual das atividades do processo concluídas para este requisito. Por exemplo, suponha uma mudança em um cálculo no “Relatório de Empregados”, identificado como uma Saída Externa – Média – 5 PF e que este requisito se encontre no fim da fase de requisitos. Então, a contagem de PF do projeto para remuneração da empresa contratada deve considerar: o novo requisito modificado, SE – Média – 5 PF, e 20% do requisito original (1 PF), totalizando 6 PFs. Deve-se ressaltar que o gestor do contrato deve evitar encaminhar para a contratada os requisitos de negócio que estejam em fase de

definição, senão poderão emergir muitas mudanças em requisitos elevando o custo do projeto em questão. Recomenda-se a implantação de processos de gerenciamento de projetos e gerenciamento de requisitos pelos contratantes, aderentes às melhores práticas de modelos da Qualidade de *Software*, visando uma gestão efetiva dos projetos contratados.

3.3 ESTABELEÇA CLÁUSULAS DE GARANTIA DA QUALIDADE

Conforme mencionado, a métrica PF considera a funcionalidade requisitada e recebida pelo usuário. Portanto, a remuneração da empresa contratada deve considerar as funcionalidades entregues, somente se estas não apresentam defeitos. Contudo, o seguinte cenário pode ocorrer: a empresa contratada entrega as funcionalidades requisitadas com defeitos; o gestor do contrato reclama, a empresa contratada corrige os erros da funcionalidade em questão; a contratante recebe o sistema de volta com outros defeitos que surgiram com a correção do erro relatado. Esse tipo de problema é comum em fábricas de *software* com um processo de testes inexistente ou inadequado. Observe que essa situação pode gerar um grande atraso no recebimento do sistema, podendo gerar atritos entre a área de TI da empresa contratante e os gestores do sistema que estão aguardando a entrega do sistema funcionando. Assim, recomenda-se o estabelecimento de cláusulas contratuais para garantir a entrega de um projeto de desenvolvimento ou manutenção de sistemas com qualidade. Sugere-se incluir no contrato uma cláusula de multa associada à qualidade do produto entregue, considerando o indicador defeitos/PF. Por exemplo, pode-se estabelecer que não é aceitável a entrega de mais de 0,3 defeitos/PF. É importante definir no contrato os tipos de defeitos, a saber: *bugs*, defeitos na documentação, código fonte não estruturado, etc. Pode-se estabelecer também níveis de severidade de defeitos.

3.4 ESTABELEÇA CLÁUSULAS CONTRATUAIS DE PRAZO E TAXA DE ENTREGA

Algumas organizações contratantes estabelecem cláusulas contratuais associadas à produtividade. Por exemplo, a fábrica de *software* contratada deve ter uma produtividade de 15 HH/PF em JAVA. Em alguns

casos, a empresa contratante pede para a contratada relatar a taxa de produtividade. Esta prática não é adequada. A produtividade é uma informação estratégica de uma empresa e ela não pode ser obrigada a divulgar estas informações. Além disso, deve-se ressaltar que em um contrato baseado em PF, o controle da produtividade da empresa contratada não faz sentido. De fato, as empresas contratantes empregam esta prática para resolver o problema de demandas recebidas com atraso de cronograma. A solução é estabelecer no contrato o método de estimativa de prazo a ser utilizado. Recomenda-se que este método utilize o tamanho em PF estimado do sistema na derivação da estimativa de prazo. Além disso, deve-se incluir cláusulas de multa considerando o atraso da entrega do projeto. Para as organizações que não possuem um processo de estimativas definido, sugere-se a utilização da Fórmula de Capers Jones (2007). Esta fórmula é simples de automatizar em uma planilha de cálculo, consistindo em elevar o tamanho do projeto em PF a um expoente t , definido de acordo com o tipo do projeto a ser estimado. Hazan (2005b) apresenta dicas para identificar o expoente t . É importante ressaltar que a fórmula é adequada apenas para projetos maiores que 100 PF. Em relação aos projetos pequenos, o contrato deve fixar prazos de acordo com o tamanho do projeto. Por exemplo, para projetos até 5 PF o prazo de entrega é de 10 dias úteis.

Outro cenário a ser considerado é o seguinte: a empresa contratada ganha um pregão fornecendo um preço muito baixo por PF e ao ganhar o contrato ela busca forçar o aumento do preço do PF contratado, definindo regras próprias para a contagem de PF. Como os órgãos públicos estão se capacitando em contagem de Pontos de Função, o gestor do contrato não aceita a contagem de PF majorada. Então, a empresa contratada aloca apenas um recurso para atendimento daquele contrato, ressaltando que os demais recursos estão trabalhando em contratos mais lucrativos. E as demandas de manutenção críticas do contratante ficam pendentes no atendimento. Portanto, visando evitar este problema, é importante definir cláusulas contratuais estabelecendo uma taxa de entrega mínima de PF/mês, por exemplo, 200 PF/mês. Deve-se incluir uma cláusula de multa tratando essa questão. O estabelecimento de uma taxa de entrega mensal máxima e mínima também

é importante para a fábrica de *software* contratada dimensionar suas equipes para um melhor atendimento ao contrato.

3.5 ESTABELEÇA O CPM COMO A BASE PARA AS CONTAGENS DE PF AO INVÉS DE CONVERSÕES

Algumas empresas contratantes estabelecem seus contratos com base na métrica Pontos de Função, no entanto não possuem capacitação adequada em contagem de Pontos de Função. Em alguns casos, estas empresas delegam a contagem para a empresa contratada que estabelece roteiros de contagem com regras que podem majorar a contagem de PF. Algumas vezes, a dimensionamento do tamanho do projeto em PF é realizado por meio de conversões de horas alocadas em Pontos de Função. Assim, é estabelecido com a empresa contratada um índice de conversão, por exemplo, 8 horas de trabalho corresponde a 1 PF, e então o pagamento da empresa contratada é feito por meio das horas alocadas ao projeto em questão convertidas em PF. Observe que se o recurso de desenvolvimento está em uma fábrica de *software* externa a empresa contratante, esta não pode gerenciar a quantidade de horas alocada ao projeto. Se o analista da empresa contratada está realizando seu trabalho nas instalações da contratante, isto é um tipo de terceirização de mão de obra. E ainda, se a contratada alocar um recurso com baixa produtividade, o custo do projeto será muito alto. A prática de conversão de horas para PF é simples, no entanto é inadequada. Se o contrato é baseado em Pontos de Função, a empresa deve realizar as contagens seguindo as regras de contagem do manual CPM.

Deve-se ressaltar que uma contagem de PF errônea pode levar a conseqüências desastrosas, tais como: pagamento incorreto do projeto contratado por PF, geração de dados para indicadores de qualidade – defeitos/PF e produtividade – horas/PF incorretos, geração de estimativas incorretas. É fundamental que as organizações que desejam estabelecer contratos de fábrica de *software* com base em Pontos de Função criem seu Escritório de Métricas com profissionais especialistas em contagem de Pontos de Função. É recomendado que estes profissionais possuam Certificação CFPS (*Certified Function Point Specialist*) e possuam experiência prática em contagem de PF e métodos de estimativas de projetos

de *software*. As fábricas de *software* contratadas também devem ter seu escritório de métricas para revisar a contagem de PF da empresa contratante. Hazan (2008) apresenta os dez erros de contagem de Pontos de Função mais observados.

3.6 ESTABELECEER REGRAS PARA DIMENSIONAR PROJETOS DE MANUTENÇÃO

Muitas organizações estabelecem em seus contratos de fábrica de *software* a prestação de serviços de desenvolvimento e manutenção de sistemas com base na métrica Pontos de Função. No entanto, o manual de práticas de contagem (CPM) trata apenas os projetos de desenvolvimento e de manutenção evolutiva. Assim, torna-se importante a definição de métricas para os demais projetos de manutenção. Pode-se contratar tais projetos em homem-hora, no entanto, conforme mencionado a IN 04 preconiza evitar a contratação de fábrica de *software* por meio da métrica homem-hora. Assim, recomenda-se a utilização de métricas baseadas nas regras de contagem de Pontos de Função para dimensionar os projetos de manutenção não contemplados no manual CPM.

O primeiro passo é a identificação de todos os tipos de projetos de manutenção que podem ser contratados pela organização. Posteriormente, deve-se definir métricas para dimensionar tais projetos. Sugere-se utilizar um percentual da contagem de Pontos de Função das funcionalidades impactadas pelo projeto de manutenção em questão, conforme descrito a seguir:

- a) Manutenção Evolutiva: A manutenção evolutiva, também denominada de projeto de melhoria funcional ou simplesmente projeto de melhoria (enhancement), está associada às mudanças em requisitos funcionais da aplicação, ou seja, a inclusão de novas funcionalidades, alteração ou exclusão de funcionalidades em aplicações implantadas. Segundo o padrão IEEE Std 1229 [INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS COMPUTER SOCIETY, 1998], esta manutenção é um tipo de manutenção adaptativa, definida como: modificação de um produto de software concluído após a entrega para mantê-lo

funcionando adequadamente em um ambiente com mudanças. A contagem de Pontos de Função desses projetos é realizada segundo a fórmula abaixo (INTERNATIONAL FUNCTION POINT USERS GROUP, 2009).

$$\text{PF} = \text{PF INCLUÍDO} + \text{PF ALTERADO} + \text{PF CONVERSÃO} + \text{PF EXCLUÍDO}$$

Onde:

- PF INCLUÍDO** = Pontos de Função associados às novas funcionalidades que farão parte da aplicação.
- PF ALTERADO** = Pontos de Função associados às funcionalidades existentes na aplicação que serão alteradas no projeto de manutenção.
- PF EXCLUÍDO** = Pontos de Função associados às funcionalidades existentes na aplicação que serão excluídas no projeto de manutenção.
- PF CONVERSÃO** = Pontos de Função associados às funcionalidades de conversão de dados dos projetos de manutenção evolutiva (Enhancement). Exemplos de funções de conversão incluem: migração ou carga inicial de dados para popular as novas tabelas criadas e relatórios associados à migração de dados.

b) **Manutenção Corretiva:** A manutenção corretiva altera o *software* para correção de defeitos. Encontra-se nesta categoria, as demandas de correção de erros em funcionalidades em sistemas em produção. Quando o sistema em produção for desenvolvido pela contratada, a manutenção corretiva será do tipo Garantia, não sendo cobrada do cliente. As cláusulas de Garantia devem ser estabelecidas em contratos. Nestes casos, a aferição do tamanho em Pontos de Função das funcionalidades corrigidas considera 60% a 80% do PF ALTERADO. Este percentual é definido de acordo com a existência de documentação atualizada do sistema em questão. Por exemplo, para um sistema com documentação desatualizada ou sem documentação, onde não é solicitada a atualização da documentação do projeto, utiliza-se a fórmula abaixo:

$$\text{PF} = \text{PF ALTERADO} \times 0,70$$

c) **Verificação de Erros:** São demandas referentes a um comportamento anormal ou indevido apontado pelo contratante em sistemas em produção. Neste caso, a fábrica de *software* contratada se mobilizará para encontrar a causa do problema. Se for constatado erro de sistema, a demanda será atendida como manutenção corretiva. Entretanto, uma vez não constatado o problema ou o mesmo for decorrente de regras de negócio implementadas ou utilização incorreta das funcionalidades, deve ser realizada a aferição do tamanho em PF das funcionalidades verificadas, considerando-se 25%, segundo a fórmula abaixo.

PF = PF NÃO AJUSTADO X 0,25

- d) **Manutenção Cosmética – Adaptação de Interface:** São consideradas manutenções cosméticas as demandas associadas às alterações de interface, por exemplo, fonte de letra, cores de telas, logotipos, mudança de botões na tela. Nestes casos, a aferição do tamanho em Pontos de Função das funcionalidades corrigidas considera 10% do PF_ALTERADO. Não será contemplada a redocumentação das funcionalidades da aplicação impactadas pela manutenção nas demandas desta categoria.

PF = PF ALTERADO X 0,10

- e) **Manutenção Adaptativa:** São consideradas manutenções adaptativas as mudanças em requisitos não funcionais da aplicação. As demandas de manutenção adaptativa são dos seguintes tipos: Redesenvolvimento de projetos em outra plataforma, Atualização de plataforma, Adequação de funcionalidades às mudanças de negócio. Frequentemente, o redesenvolvimento de um sistema em outra plataforma é tratado como um novo projeto de desenvolvimento. No caso de atualização de plataforma, por exemplo, realizar uma manutenção em um Portal para que este suporte mais um *browser* ou uma versão mais atual de um *browser*, deve-se estabelecer um índice percentual redutor para a contagem de Pontos de Função, considerando cada tipo de atualização de plataforma. As mudanças em requisitos não funcionais podem ser diversas, por exemplo otimização de algoritmos para melhora de performance, etc. Nesses casos, deve-se estabelecer em contrato o percentual para cada tipo de manutenção adaptativa.
- f) **Apuração Especial:** São funcionalidades executadas apenas uma vez para: corrigir problemas de dados incorretos na base dados das aplicações ou atualizar dados em bases de dados de aplicações; ou gerar um relatório específico ou arquivo para o usuário por meio de recuperação de informações nas bases da

aplicação. Estes projetos são dimensionados seguindo a fórmula de manutenção evolutiva do CPM, onde as funcionalidades de atualização de dados ou geração de relatórios são consideradas PF_INCLUÍDOS. De um modo geral, a atualização de dados de sistemas em produção será contada como Entrada Externa e a geração de relatórios ou arquivos será contada como Consulta Externa ou Saída Externa.

Outros tipos de projetos de manutenção podem ser identificados nas organizações e descritos nos contratos. Algumas organizações contratam a documentação de sistemas legados, nesses casos, pode-se considerar 20% do tamanho do projeto em PF, caso seja requisitada apenas a documentação dos requisitos. Caso a demanda seja a geração de artefatos de documentação de todas as fases do processo de desenvolvimento, deve-se considerar um percentual de 30% a 50%, dependendo dos artefatos a serem gerados.

4 CONCLUSÃO

De fato, a utilização da métrica Pontos de Função Não Ajustados como unidade de medida de tamanho funcional em contratos de prestação de serviços de desenvolvimento e manutenção de sistemas é uma excelente prática. Além da existência de regras de contagem de Pontos de Função bem definidas e publicadas no manual de práticas de contagem (CPM), pode-se ressaltar os seguintes benefícios na utilização da métrica, dentre outros: independência da tecnologia e metodologia utilizadas, baseada na visão do usuário, facilidade de estimativa nas fases iniciais do ciclo de vida do *software*, a utilização da métrica como dado padrão para a definição de indicadores da qualidade de *software* e apoio na avaliação dos benefícios de aquisição de pacotes de *software* (análise “*make or buy*”).

É importante ressaltar que em um processo de contratação de fábrica de *software*, além da estimativa de tamanho do projeto em Pontos de Função, outros aspectos devem ser considerados, a saber: definição do processo de desenvolvimento

a ser seguido pela contratada com detalhamento dos artefatos a serem entregues, cronograma do projeto, definição dos acordos de nível de serviço, definição com clareza do objeto a ser contratado, considerando os requisitos funcionais e os requisitos não funcionais do projeto. Observe que os requisitos não funcionais (performance, segurança, padrão de interface...) não contam Pontos de Função, no entanto impactam nas estimativas de esforço, custo e prazo do projeto.

Este artigo apresentou os principais problemas observados em contratos de fábrica de *software* baseados em Pontos de Função e sugestões de solução para os mesmos. O principal problema observado é a falta de maturidade das empresas na utilização da métrica PF levando às contagens de PF erradas e atritos entre contratantes e contratadas. Portanto, recomenda-se a criação de um Escritório de Métricas nas empresas contratantes e contratadas sob a coordenação de um consultor de métricas com experiência teórica e prática, visando assegurar a qualidade das contagens de Pontos de Função dos projetos contratados.

5 REFERÊNCIAS

AGUIAR, M. Contratando o desenvolvimento com base em métricas. In: PROJECT MANAGEMENT INSTITUTE, 2000, Rio de Janeiro. *Reunião*. Disponível em: <<http://www.bfpug.com.br/Artigos/PMI-RIO-27-11-2000.htm>>. Acesso em: 27 ago. 2010.

ALBRECHT, A. J. Measuring application development productivity. In: JOINT SHARE/GUIDE/IBM Application Development Symposium, 1979, California. *Proceedings...* [S.l. : s.n., 1979?], p. 83-92. Disponível em: <<http://www.bfpug.com.br/Artigos/Albrecht/MeasuringApplicationDevelopmentProductivity.pdf>>. Acesso em: 27 ago. 2010.

ALBRECHT, A.; GAFFNEY, J. Software function, source lines of code, and development effort prediction: a software science validation. *IEEE Transactions on Software Engineering*, Piscataway, NJ, USA, v. 9, n. 6, p. 639-648, 1983. Disponível em: <<http://www.computer.org/portal/web/csdl/doi/10.1109/TSE.1983.235271>>. Acesso em: 27 ago. 2010.

DEKKERS, C. Measuring the "logical" or "functional" size of software projects and software application. *ISO Bulletin*, May, 2003, p. 10-13. Disponível em: <http://www.bfpug.com.br/Artigos/ISO-0926_Article.pdf>. Acesso em: 27 ago. 2010.

HAZAN, C.; BERRY, D.M.; LEITE, J.S.P. É possível substituir processos de engenharia de requisitos por contagem de pontos de função? In: INTERNATIONAL WORKSHOP ON REQUIREMENTS ENGINEERING (WER), 8th, 2005, Porto, Portugal. *Anais...* [Porto: s.n., 2005?]. p. 197-208. Disponível em: <http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER05/claudia_hazan.pdf>. Acesso em: 27 ago. 2010. [2005a].

HAZAN, C.; OLIVEIRA, E.A.; BLASCHEK, J.R. How to avoid traps in contracts for software factory based on function point metric. In: INTERNATIONAL SOFTWARE MEASUREMENT & ANALYSIS CONFERENCE, 3rd, 2008, Washington, USA. *Track presentation abstracts*. [Resumo] disponível em: <<http://www.ifpug.org/conferences/2008/trackPresentationAbstracts.htm>>. Acesso em: 27 ago. 2010.

HAZAN C.; VON STAA, A. Análise e melhoria de um processo de estimativas de tamanho de projetos de software. *Monografias em Ciência da Computação*, Rio de Janeiro, n. 4, 2005. (MCC 04/05). ISSN 0103-9741. [Resumo] disponível em: <<http://bib-di.inf.puc-rio.br/techreports/2005.htm>>. Acesso em: 27 ago. 2010. [2005b].

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS COMPUTER SOCIETY (IEEE). *IEEE standard for software maintenance*. [New York], 1998. (IEEE Std 1219, 1998.). ISBN 0-7381-0336-5.

INTERNATIONAL FUNCTION POINT USERS' GROUP (IFPUG). *Counting practices manual*. Version 4.3. [Princeton, NJ], 2009.

JONES, C. *Estimating software costs: bringing realism to estimating*. 2nd ed. New York: Mc Graw Hill, 2007.

JONES, C. Function points. *Computer*, v. 27, n. 8, p. 66-67, Aug. 1994.

KARNER, G. *Resource estimation for objectory projects*. 1993. Disponível em: <<http://www.bfpug.com.br/Artigos/UCP/Karner%20-%20Resource%20Estimation%20for%20Objectory%20Projects.doc>>. Acesso em: 27 ago. 2010.

KOTONYA, G.; SOMMERVILLE, I. *Requirements engineering: processes and techniques*. New York : J. Wiley, 1998.

PERRY, W.E. The best measures for measuring data processing quality and productivity. *Quality Assurance Institute Technical Report*. 1986.