

Avaliação de estimativa de tamanho para Projetos de Manutenção de software

Angélica Toffano Seidel Calazans¹, Marcelo Antonio Lopes de Oliveira²

¹Gerência Nacional de Desenvolvimento e Manutenção de Sistemas – Caixa Econômica Federal – Brasília – DF – Brasil.

angelica.calazans@caixa.gov.br

²Politec Ltda – Brasília – DF – Brasil.

marceloo@bsb.politec.com.br

Abstract. The maintenance is a stage in the lifecycle of a software. This activity is identified as the most difficult and it demands such a great effort. The process of maintenance starts after the implementation of the software product to correct defects as well as to improve performance, add attributes and so on forth. As a stage in the lifecycle of a software, organizations need to identify enough effort for software maintenance, know the software size and the necessary time for its implementation. This document shows a narrative about an experience of the application of a software measurement technique known as Function Point Analysis (FPA) in a federal institution just for maintenance projects.

1 Introdução

A manutenção de software é reconhecida como a atividade que demanda o maior volume de esforço dentre todas as atividades de engenharia de software [13]. Grande parte dos sistemas sofre alterações ao longo do período em que estão sendo utilizados, seus requisitos originais são modificados para refletir algum tipo de mudança das necessidades do usuário, mudança de ambiente, introdução de novo hardware ou software, correção de erros e outros.

O trabalho de manutenção também é realizado sobre sistemas legados construídos há muito tempo, por profissionais que já perderam o vínculo com a empresa, e com pouca ou nenhuma documentação. Estes sistemas foram, com o passar dos anos, migrados para novas plataformas, ajustados devido a mudanças nos sistemas operacionais e melhorados para atender novos requisitos funcionais. Muitas organizações necessitam ainda evoluir estes sistemas para fornecer informações críticas e manter suas operações.

Segundo [14] “a natureza ubíqua da modificação permeia todo o trabalho de software. Modificação é inevitável quando sistemas baseados em computador são construídos; conseqüentemente, precisamos desenvolver mecanismos para avaliar, controlar e realizar modificações”.

As empresas necessitam estimar, de forma acurada, o tamanho dos produtos de software no início do processo de desenvolvimento, visando a realização de um melhor planejamento para a construção de produtos de software e, ainda, diminuir o risco da tomada de decisões errôneas [15].

As empresas também necessitam estimar, de uma forma adequada, o tamanho dos projetos de manutenção antes mesmo de eles realmente iniciarem. Nesse contexto, diferentes abordagens foram propostas para estimar o tamanho de um sistema ou mesmo de uma manutenção, como, por exemplo, a Análise por Pontos de Função – APF.

Contudo, a manutenção de um produto de software tem particularidades específicas que devem ser levadas em consideração. É necessário, portanto, avaliar as abordagens existentes, identificando aspectos positivos e negativos e mesmo alternativas para melhor estimar este tipo de serviço.

Este trabalho apresenta uma avaliação da abordagem de Análise por Pontos de Função para estimativa de tamanho de projetos de manutenção. São apresentados e analisados, também, resultados da aplicação da métrica em projetos reais da indústria.

Este documento está organizado nas seguintes seções: breve descrição do Processo de Manutenção (seção 2), breve descrição sobre uma métrica para estimativa de tamanho, duração e custo (seção 3), aplicação da APF em Projetos de Melhoria (seção 4) e conclusões e sugestões de melhoria do processo (seção 5).

2. Processo de Manutenção de Software

O processo de mudança do software após a sua implementação, para corrigir defeitos, melhorar performance ou adaptar o produto a modificações para atender a novas necessidades do negócio é chamado de Manutenção de Software [14].

A ISO/IEC 12207 [6] considera o desenvolvimento e manutenção do software como processos que compõem o ciclo de vida de software. O Processo de Manutenção é composto de um conjunto completo de sub processos, atividades e tarefas que podem ser aplicados durante a manutenção de produtos de software para modificar um produto de software existente, preservando a sua integridade.

Segundo [13] a manutenção de sistemas é uma realidade no mercado de software, sendo considerada em estimativas como até 80% de todo o trabalho de engenharia de software. O autor identifica vários fatores que justificam a necessidade de manter um produto de software, entre eles: dar continuidade ao serviço executado pelo produto, suportar alterações obrigatórias (legais, etc), suportar as melhorias necessárias em termos de funcionalidade do produto e para facilitar futuras manutenções.

Outros autores e [13] classificam a manutenção de software em 4 categorias:

- manutenção corretiva – referente à manutenção que tem como objetivo identificar e remover de falhas no software. Estas falhas podem ser decorrentes de erro de design, erros lógicos ou erros de codificação;
- manutenção adaptativa – modificações no software para acomodar mudanças no seu ambiente externo. Manutenção adaptativa inclui o trabalho para migrar o software para diferentes plataformas de softwares e hardwares

(compiladores, sistemas operacionais, novos processadores, banco de dados, etc);

- manutenção perfectiva – é a manutenção efetuada para aprimorar o software além dos seus requisitos funcionais originais, a partir do momento em que o usuário identifica novas funcionalidades que trarão benefício ao seu negócio. A expansão dos requisitos pode agregar melhoria às funcionalidades existentes do software ou melhorar a eficiência do sistema;
- manutenção preventiva – ao longo do seu ciclo de vida o software se deteriora devido a modificações. A manutenção preventiva é implementada para permitir que software possa ser mais facilmente corrigido, adaptado e melhorado para servir às necessidades de seus usuários finais, aprimorar a confiabilidade ou a manutenibilidade futura.

Segundo [3], apesar da manutenção ser considerada parte do ciclo de desenvolvimento do software (ISO/IEC 12207), existem diferenças fundamentais entre a atividade de desenvolver e a atividade de manter um software. Um novo desenvolvimento constitui construir produtos novos e a manutenção trabalha com parâmetros e construção de algo sobre um sistema existente. Na manutenção há necessidade de se ter um conhecimento geral do que o software faz, como está implementado, identificar aonde as mudanças devem ser introduzidas e analisar todos os seus impactos.

Segundo [13], a manutenção de um sistema é uma atividade difícil e depende de vários fatores como: entendimento limitado do pessoal para estudar a documentação e obter material relevante para o problema a ser resolvido, prioridade de modificar o sistema antigo para atender aos usuários finais (criando um produto difícil de ser entendido e posteriormente de ser alterado), etc

Kusters & Heemstra [10] relatam os seguintes resultados obtidos em uma pesquisa realizada junto a seis grandes organizações alemãs, com a finalidade de dimensionar as dificuldades presentes na manutenção de software: existe uma falta generalizada de percepção sobre o tamanho das manutenções de software, bem como do custo correspondente; o gasto é alto e os orçamentos tem sido extrapolados; a manutenção é uma atividade de difícil planejamento; e, somente uma, entre as seis organizações, registrava dados sobre seus processos de manutenção e os usa para planejamento.

Considerando o acima exposto e a necessidade das organizações de estimar com maior acurácia o tamanho deste produto, a seguir identificamos os conceitos de medição e as características da APF, principalmente com relação à mensuração do produto de manutenção.

3. Medição de software

Segundo [2], medição é o processo de obtenção de uma medida para uma entidade real. Uma medida fornece uma indicação de quantidade, dimensão, capacidade ou tamanho de algum produto de software ou de um processo. Em outras palavras, uma medida refere-se a um valor de uma métrica. Segundo a norma ISO 9126 [5], métrica é a composição de métodos para medição e escalas de medição.

Para se chegar a uma medida de software existem muitas métricas¹ que avaliam as variáveis de esforço e duração e tamanho. A NBR ISO/IEC 14143 [8] tem como objetivo garantir que todos os métodos de Medição Funcional de Tamanho sejam baseados em conceitos similares e que possam ser testados para assegurar que eles se comportam de maneira similar e da forma esperada por um método de medição.

A seguir serão descritas as principais características da Análise por Pontos de Função – APF, por ser o método de medição funcional de tamanho utilizado neste trabalho.

3.1 Análise por Pontos de Função - APF

A Análise por Pontos de Função (APF) mede o tamanho do software pela quantificação de suas funcionalidades, baseadas no projeto lógico ou a partir do modelo de dados segundo a visão e os requisitos do usuário final [4]. Atualmente a APF é reconhecida como padrão ISO/IEC 20926 [7].

As principais características da APF são: ser independente da tecnologia, ser aplicável desde o início do sistema, apoiar a análise de produtividade e qualidade e estimar o tamanho do software com uma unidade de medida padrão.

A APF considera as funções de dados, divididas em Arquivos Lógicos Internos (ALIs - que são grupos lógicos de dados mantidos dentro da fronteira da aplicação) e Arquivos de Interface Externa (AIEs – arquivos somente referenciados pela aplicação) e as funções transacionais, divididas em Entradas Externas (EEs), Saídas Externas (SEs) e Consultas Externas (CEs).

Cada função de dado ou transacional terá um peso diferente dependente de sua complexidade. Diversas tabelas baseadas na quantidade de elementos de dados, de registros e de arquivos referenciados são utilizadas para determinar a complexidade de cada função em Baixa, Média ou Alta.

O resultado da contagem de funções de dados e transacionais é uma medida chamada de pontos de função não ajustados (NoPF não ajustado), pois não considera detalhes que afetam o produto e sua construção. O ajuste na mensuração é efetuado através do Fator de Ajuste², conjunto de 14 características que influenciarão a complexidade do software.

3.2 APF para Projetos de Melhoria

Segundo o Manual de Práticas de Contagem do IFPUG [4], uma contagem de um projeto de “melhoria”, mede as alterações realizadas em uma aplicação existente com a finalidade de incluir, excluir ou modificar funcionalidades entregues. Não estão contempladas manutenções corretivas e preventivas.

¹ Ex. LOC, APF, Halstead, COCOMO, COSMIC FFP

² Alguma delas: comunicação de dados, processamento distribuído, performance, utilização de equipamento, volume de transações, etc.

A APF trata como "melhorias" as modificações realizadas nas funções de dados (ALI e AIE) e nas funções transacionais (EE, SE ou CE), resultantes de manutenções. Para cada tipo de função, os desenvolvedores devem determinar se uma nova função está sendo adicionada ao sistema e/ou se uma função existente está sendo modificada e/ou excluída.

A seguinte fórmula é aplicada para mensurar projetos de melhoria:

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

onde:

Sigla	Descrição
EFP	Número de pontos de função do projeto de melhoria
ADD	Número de pontos de função não-ajustados das funções incluídas pelo projeto de melhoria
CHGA	Número de pontos de função não-ajustados das funções modificadas. Reflete as funções depois das modificações
CFP	Número de pontos de função não-ajustados adicionados pela conversão ³ .
VAFA	Valor do fator de ajuste da aplicação depois do projeto de melhoria
DEL	Número de pontos de função não-ajustados das funções excluídas pelo projeto de melhoria
VAFB	Valor do fator de ajuste da aplicação antes do projeto de melhoria

Tran-Cao e Levesque [16] demonstraram que a APF, quando aplicada a alguns projetos de manutenção, não apresentou bons resultados com relação a estimativas de tempo de execução quando comparadas com o tempo real. Meli [11] questiona e identifica a necessidade de uma abordagem de mensuração de tamanho de software mais acurada das manutenções.

4. Aplicação de APF em Projetos de Melhoria

Os projetos de manutenção analisados são de uma instituição do governo federal que possui um processo de desenvolvimento institucionalizado (baseado no modelo estruturado) e utiliza constantemente a APF para mensurar novos desenvolvimentos e manutenção.

Foram analisados 17 projetos de manutenção já finalizados. O objetivo de analisar estes projetos foi poder comparar o esforço estimado (qtd horas estimadas para realizar a manutenção) inferido a partir da aplicação de fatores de produtividade ao

³ Funções disponíveis no momento da instalação da aplicação para converter dados ou fornecer outros requisitos de conversão como relatórios para verificação da conversão (pode ou não existir)

tamanho do projeto obtido com a métrica APF e o esforço real (qtd horas reais utilizadas pelas equipes de manutenção), conforme demonstrado na Tabela 1.

Os projetos foram pontuados seguindo a metodologia da APF para manutenção citada anteriormente. Para cálculo do esforço estimado foram utilizados os percentuais de esforço definidos para cada uma das etapas da metodologia utilizada⁴ e a produtividade (Horas por PF) de cada etapa.

A produtividade (quantidade de horas necessárias para produzir 1 ponto de função), foi baseada na análise das informações do banco de dados do ISBSG - International Software Benchmarking Standards Group ⁵.

Estes valores (percentuais de esforço e produtividade) são utilizados pela empresa em seus contratos de terceirização.

Tabela 1 – Manutenções pontuadas em APF x esforço estimado APF(horas) x esforço real (horas)

Sistema	Quantidade PF	APF - esforço estimado (hs)	Real Esforço (hs)	Real esforço/ APF esforço Impacto
Sist. 1	32,56	414,72	114,78	0,28
Sist. 2	23,76	302,65	207,85	0,69
Sist.3	84,6	750,59	639,36	0,85
Sist. 4	2,61	37,85	19	0,5
Sist. 5	20	291,67	285	0,98
Sist. 6	18	262,5	90	0,34
Sist. 7	22,8	315,4	91	0,29
Sist. 8	3,8	52,57	105	2
Sist. 9	2,55	36,55	7	0,19
Sist. 10	6,12	88,74	8	0,09
Sist. 11	3,4	28	24	0,86
Sist. 12	6,8	62,22	62	1
Sist. 13	35	499,98	270	0,54
Sist. 14	14	199,64	74	0,37
Sist. 15	93	1326,18	522	0,39
Sist. 16	66,12	1208,4	530,36	0,44
Sist. 17	16,95	241,85	136,03	0,56

A Figura 1 demonstra de forma gráfica a diferença de esforço obtido entre a estimativa utilizando APF e o esforço real em horas.

⁴ A metodologia estruturada na organização estudada possui 7 fases: Anteprojeto, Planejamento, Análise, Projeto, Construção, Homologação e Implantação.

⁵ Grupo focado em coletar, validar e publicar, num repositório, valores históricos de produtividade por linguagem em projetos de softwares.

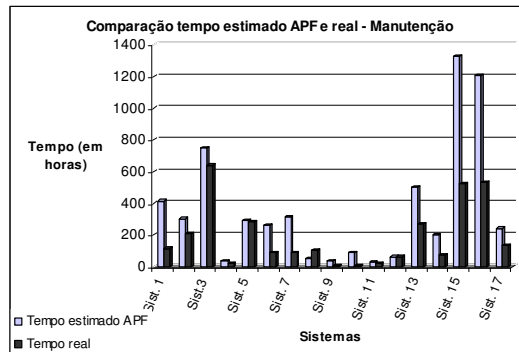


Figura 1 – Comparação do tempo estimado APF e o tempo real (em horas)

4.1 Resultados obtidos

Em 88% dos projetos da amostra estudada, o cálculo do esforço (qtd de horas) obtido com a utilização da APF, ficou acima do tempo real gasto para a efetiva construção da manutenção.

Após esta análise, foi estudado o fator de impacto (valor obtido a partir da divisão do tempo real e o tempo estimado). Este estudo possuiu o objetivo de identificar o impacto médio das diferenças observadas entre o esforço real e o esforço estimado APF (Figura 2). Quanto mais próximo de 1, mais aderente estaria a estimativa ao real.

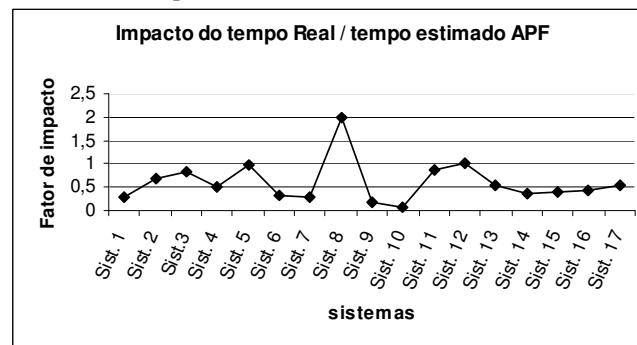


Figura 2 –Fator de Impacto (tempo real/tempo estimado APF) por projeto

Em 94% dos projetos analisados o fator de impacto ficou abaixo de 1 (inclusive). Considerando este resultado, foi desconsiderado o dado referente ao Proj. 8 e calculada a média do fator de impacto para os 16 projetos restantes. A média do fator de impacto para esta amostra ficou em 0,51, ou seja, a utilização da APF, na amostra estudada, superestima em 51% o tempo para manutenção.

A Figura 3 apresenta um relatório de dispersão entre o tempo real/tempo estimado APF (fator de impacto) e a quantidade de Pontos de função. Este gráfico visa identificar a existência ou não de correlação entre estes dois fatores, ou seja, se o tamanho interfere em uma maior ou menor distorção com relação à estimativa/real. Como pode ser verificado existe uma correlação nula entre estas variáveis, o que leva a inferir que não existe, na amostra estudada, relação entre o fator impacto (estimativa/real) e o tamanho da manutenção.

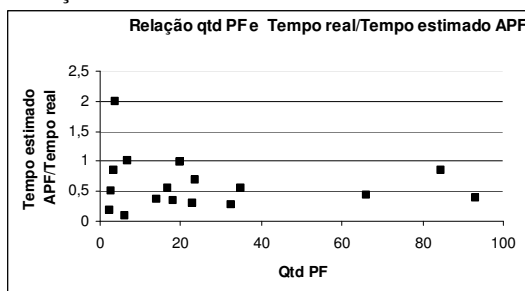


Figura 3 – Relação Fator impacto e Qtd PF

Considerando os resultados de estimativa obtidos, pode-se concluir, que há evidências, na amostra estudada, que os resultados obtidos através da aplicação da métrica APF não possuem um grau de acurácia razoável em 94% dos projetos de manutenção analisados.

Foi realizada uma entrevista estruturada junto às equipes de manutenção e foi consenso que pequenas manutenções, onde há a alteração/exclusão de um atributo de um ALI, ou seja, uma tarefa relativamente simples, a pontuação de todo o ALI na melhoria tem aumentado substancialmente a contagem final da manutenção. Estas equipes reconhecem que, com relação aos Projetos de Melhoria, a APF superestima o tamanho.

É necessário reconhecer esta deficiência para que as organizações possam identificar caminhos alternativos para a mensuração de projetos de manutenção. Aumentar a produtividade (qtd horas/PF) para manutenção pode ser uma alternativa. Mas, pesquisa feita em bases históricas de produtividade de organizações [9], demonstrou que a produtividade de manutenção e a produtividade de novo desenvolvimento destas bases, tem sido aproximada. Ou seja, utilizar este recurso, para garantir uma melhor acurácia da definição de esforço, inferida por meio de estimativa de tamanho errônea, seria mascarar a deficiência da métrica neste escopo. Outra alternativa seria pontuar os projetos com a utilização de outras abordagens.

4.2 Alternativas para mensuração de Projetos de Melhoria

Existem várias outras abordagens para mensuração de tamanho e a maior parte delas se propõe a mensurar inclusive projetos de manutenção. A abordagem COSMIC Full Function Points é uma das abordagens mais atuais, foi proposta inicialmente, em 1997 e em 1999 o grupo Common Software Measurement International Consortium –

COSMIC propôs a abordagem COSMIC – FFP como uma métrica totalmente independente [1]. A mensuração de tamanho com relação ao escopo de projetos de manutenção, nesta abordagem, é mais pontual uma vez que considera somente os dados modificados/alterados ou excluídos. A COSMIC não pontua funções de dados.

A abordagem proposta pela NESMA (Netherlands Software Metrics Users Association) para manutenção é uma alternativa para este escopo [12]. A NESMA é uma associação de usuários de métricas que tem indicado alternativas de contagem, utilizando a APF, de forma a possibilitar medir de forma mais acurada alguns escopos em que a APF não tem se mostrado eficaz. Sua proposta permite a redução do tamanho do Projeto de Melhoria considerando uma relação proporcional das alterações feitas com o que existia anteriormente.

5. Conclusão

A manutenção faz parte do ciclo de vida de um sistema. Os sistemas são como os negócios, estão constantemente em evolução. Mensurar o tamanho da manutenção para inferir prazos em que a demanda será atendida, é necessidade de todas as organizações.

Neste trabalho, foi apresentada a aplicação da Análise por Pontos de Função em 17 projetos de melhoria de uma instituição governamental. Conforme pode ser observado, os tempos estimados, obtidos após a aplicação da métrica, não se mostraram compatíveis com os tempos reais, utilizados para manutenção destes sistemas.

O reconhecimento da falta de acurácia da métrica, na amostra estudada, é um fator importante para identificar necessidade de melhoria. É claro que um número maior de projetos deve ser analisado para confirmar a distorção apontada, mas a identificação deste problema é essencial para que outras alternativas de mensuração sejam analisadas e mesmo propostas para este escopo de serviço.

Como trabalhos futuros serão aplicados outros tipos de abordagens, visando obter informações que tornem possível a realização de estudos comparativos entre as técnicas.

A abordagem de mensuração de tamanho deve ser escolhida e/ou adequada, dependendo das características do sistema que se pretenda desenvolver ou manter.

Referências

1. ABRAN A.; SYMONS C.; OLIGNY S. An Overview of Cosmic – FFP field trial results. In: The 12nd European Software Control and Metrics Conference - ESCOM. 2001.
2. FENTON, N., PFLEEGER, S. Software Metrics A Rigorous & Practical Approach. Boston: PWS Publishing Company, 1997. 638 p.
3. GRUBB, Penny; TAKANG, Armstrong. Software Maintenance Concepts and Practice. Second edition. Singapore: Word Scientific Printers, 2003.
4. IFPUG. International Function Point Users Group. Function Point Counting Practices Manual: Release 4.1. Ohio: IFPUG. 2000. 1 v.

5. ISO/IEC 9126-1. Software Engineering – Product Quality. 2001.
6. ISO/IEC 12207. Information technology - Software Life Cycle Process. 1998.
7. ISO/IEC 20926. Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting practices manual. 2003.
8. NBR ISO IEC 14143-1:2000 - Medição de software – Medição de tamanho funcional – Parte 1: Definição de conceitos
9. ISBSG. Benchmarking Repository, Release 6. ISBSG. Abr, 2002.
10. KUSTYERS; HEEMSTRA. Software Maintenance: an approach towards control. IEEE International conference on Software Maintenance, Italy. 2001.
11. MELI, Roberto. Functional and Technical software measurement: conflict or integration? Data Processing Organization. 2000.
12. NESMA, Function Point Analysis for Software Enhancement. Disponível em : [http://www.nesma.nl/download/gidsen/FPA for Software Enhancement \(1.0\).pdf](http://www.nesma.nl/download/gidsen/FPA_for_Software_Enhancement_(1.0).pdf). Consultado em : 26/12/2003.
13. PFLEEGER, Shari. Engenharia de software: teoria e pratica. 2ª. Edição. São Paulo: Prentice Hall, 2004.
14. PRESSMAN, Roger. Engenharia de Software. 5ª. Edição. Rio de Janeiro: McGraw-Hill, 2002.
15. STUTZKE, R. Predicting Estimation Accuracy. In: The European software control and metrics conference – ESCOM, Alemanha, p. 211 – 220, 2000.
16. TRAN-CAO, De ; LEVESQUE, Ghislain. Maintenance effort and cost estimation using software functional sizes. University of Quebec in Montreal (UQAM), 2002.