TRIBUNAL DE CONTAS DA UNIÃO

Manual de Medição Funcional de Software

Versão 4.0

Sumário

Manual de Medição Funcional de Software *1*
1. Introdução *4*
2. Processo De Medição De Software *5*
2.1. Determinação Do Tipo De Medição **5**
2.2. Identificação Do Escopo Da Medição E Da Fronteira Da Aplicação **7**
2.3. Identificação Das Funções de Dados **8**
2.4. Identificação Das Funções Transacionais **11**
2.5. Mensuração Do Tamanho Funcional De Cada Função **14**
2.6. Aplicação Dos Percentuais Por Atividade Implementada **20**
2.7. Dimensionamento Final **20**
3. Documentação de Medições *20*
3.1. Nomenclatura **21**
3.2. Itens Exigidos Em Uma Contagem Detalhada **21**
4. Itens Não Mensuráveis *22*
4.1. Manutenção Corretiva Em Sistema Legado **22**
4.2. Documentação De Sistema **22**
4.3. Atualização De Plataforma **22**
4.4. Manutenção Cosmética **22**
4.5. Adaptação De Funcionalidades Sem Atualização De Requisitos Funcionais **23**
4.6. Múltiplas Mídias **23**
4.7. Primeira Referência A Arquivos Lógicos Em Contextos Diferentes De Desenvolvimentos **24**
4.8. Dados De Código **24**
4.9. Criação E Alteração De Scripts De Atualização Do Siga **24**
4.10. Reexecução De Scripts De Atualização E De Carga De Dados **24**
5. Aplicação da Métrica em Contextos Específicos de Desenvolvimento *25*
5.1. Medição De Desenvolvimento Por Parametrização **25**
5.2. Medição De Portais Web **25**
5.3. Medição De Componentes De Software **26**
5.4. Medição de Data Warehouse **28**
5.5. Medição de Projetos Ágeis **32**
6. Métrica Na Gestão De Contratos Externos *33*

6.1. Tipos De Medição Aplicados 33
6.2. Funções Já Existentes Na Aplicação Não Serão Remuneradas 33
6.3. Aceite De Medição É Obrigatório 33
6.4. Diferenças De Contagem 33
6.5. Reaproveitamento De Funcionalidades 34
6.6. Tratar Revisões do Manual 34
6.7. Documentação Exigida 34
6.8. Revisão Dos Percentuais Por Atividade Implementada 34
6.9. Manutenção Corretiva 34
6.10. Prazo Máximo De Desenvolvimento 35
7. Glossário 36
8. Referências Bibliográficas 40
Anexo I - Relatório de medição estimativa de tamanho funcional (exemplo) 41
Anexo II - Relatório de contagem funcional detalhada (exemplo) 42

1. Introdução

Este manual visa definir as regras de contagem funcional a serem utilizadas para o processo de desenvolvimento e manutenção de software no âmbito do Tribunal de Contas da União.

A definição e o estabelecimento de métricas são fundamentais para o dimensionamento de um projeto e para o acompanhamento de seu desenvolvimento. A partir das informações obtidas com o uso de métricas, pode-se avaliar a qualidade do processo de desenvolvimento e verificar o resultado da utilização de uma técnica ou ferramenta. Por essa razão, os sistemas de verificação de qualidade, como a norma ISO 9000:2000 e o CMMI, normalmente exigem a definição de métricas.

A medição funcional é um termo geral para métodos de dimensionamento de software baseados nas funções requeridas pelos usuários.

A norma ISO/IEC 14143 foi desenvolvida para garantir que todos os métodos de medição de tamanho funcional sejam baseados em conceitos similares e se comportem de maneira similar.

O presente manual se baseia na métrica Pontos de Função definida pelo IFPUG no Manual de Práticas de Contagem (CPM) versão 4.3.1 (IFPUG, 2010), e na métrica EFPA definida pela Nesma no documento *Function Point Analysis for Software Enhancement* versão 1.0. Logo, conceitos e detalhes das métricas devem ser buscados nos documentos citados, desde que não conflitantes com os explicitados neste manual.

Traz também a possibilidade de dimensionamento de software pela métrica Elementos Funcionais (EF) proposta por CASTRO e HERNANDES (2013) e suas submétricas Elementos Funcionais de Transação (EFt) e Elementos Funcionais de Dados (EFd), que se baseiam na métrica Pontos de Função, mas sem algumas de suas deficiências.

Complementa este manual o guia “Melhores Práticas de Medição Funcional” do TCU. É comum o surgimento de dúvidas e divergências em contagens funcionais, tendo em vista ser a visão do usuário, alvo da medição, subjetiva. Nesses casos, as interpretações validadas e acordadas são registradas no guia, que será constituído ao longo do tempo. Esse guia tem como objetivo preservar as decisões para referência futura, eliminando o retrabalho e divergências.

Para efeito deste manual, os termos “projeto de melhoria” e “projeto de desenvolvimento” do método FSM do IFPUG equivalem aos termos “caso de melhoria” e “caso de desenvolvimento” respectivamente. Preferiu-se usar essa terminologia para não haver confusão com o conceito de projeto já empregado na metodologia de desenvolvimento do TCU.

O presente manual trata inicialmente do processo de contagem. Em seguida apresenta algumas considerações sobre a utilização da métrica no processo de terceirização. Por fim, apresenta um glossário com algumas definições de conceitos usados no corpo deste manual.

2. Processo De Medição De Software

2.1. Determinação Do Tipo De Medição

2.1.1. Quanto ao alvo da medição

Consiste na qualificação do alvo da medição. Podemos medir as funcionalidades envolvidas em casos de desenvolvimento ou de manutenção e também as funcionalidades presentes em uma aplicação. Detalhamos nesta seção os tipos de medição quanto ao alvo.

2.1.1.1. Caso de desenvolvimento

Situação de criação de demanda do usuário relativa a uma nova aplicação. O tamanho funcional de um caso de desenvolvimento mede a funcionalidade fornecida aos usuários finais do software quando da sua primeira instalação. Isso significa que essa medição também abrange as eventuais funções de conversão de dados necessárias à implantação da aplicação.

Os componentes para o cálculo do tamanho funcional de um caso de desenvolvimento são:

- Funcionalidades da aplicação requisitadas pelo usuário para o caso (FInc) – funções utilizadas após a instalação do software para satisfazer as necessidades correntes do negócio do usuário;
- Funcionalidades de conversão requisitada pelo usuário para o caso (FConv) – funções disponíveis no momento da instalação da aplicação para converter dados ou fornecer outros requisitos de conversão especificados pelo usuário, como relatórios de verificação de conversão e rotinas de migração. Após a instalação, essas funções não serão mais usadas.

A fórmula de cálculo do tamanho funcional do caso de desenvolvimento (FDes) é a seguinte:

FDes = FInc + FConv

2.1.1.2. Caso de melhoria ou redesenvolvimento

Trata-se da situação em que uma aplicação já foi desenvolvida, mas o usuário solicita algum tipo de manutenção adaptativa, evolutiva ou mesmo corretiva ou trata-se da situação em que o usuário solicita o redesenvolvimento total ou parcial de uma aplicação existente. O tamanho funcional de um caso de melhoria ou redesenvolvimento mede as funções adicionadas, modificadas ou excluídas da aplicação pelo caso e também as eventuais funções de conversão de dados. Também podem ser incluídas no escopo de um caso de melhoria ou redesenvolvimento funcionalidades que serão alvo somente de teste, tendo em vista possuírem alto acoplamento com as funcionalidades integrantes da melhoria ou redesenvolvimento.

2.1.1.3. Aplicação

Situação em que se deseja dimensionar o tamanho funcional disponibilizado para o usuário em uma determinada aplicação ou de parte da mesma.

2.1.2. Quanto ao método de medição

2.1.2.1. Medição indicativa

Este método é utilizado antes da iniciação de um projeto de desenvolvimento de software. Na contagem indicativa, em conformidade com o trabalho *Early Function Point Counting*, publicado pela Nesma, admite-se que já foram identificadas as funções de dados do projeto: AIEs e ALIs. O método atribui então 35 FP para cada ALI e 15 FP para cada AIE identificado, sendo esses números obtidos conforme o detalhamento a seguir.

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

Para uma contagem indicativa em elementos funcionais, deve-se considerar 25 EF para cada ALI e 12 EF para cada AIE.

2.1.2.2. Medição estimativa

O objetivo específico dessa contagem, baseada no método da Nesma, é determinar o tamanho funcional do software de modo a sustentar ações gerenciais para planejamento do projeto, a partir de um segundo detalhamento do escopo do software a ser desenvolvido, o que permite resultado mais próximo da realidade.

Em uma fase geralmente posterior do ciclo de desenvolvimento, quando já são conhecidas, em linhas gerais, todas as funcionalidades do sistema, deverá ser utilizado o método de contagem estimativa, publicado 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.

Para uma contagem estimativa em elementos funcionais, deve-se considerar o tamanho equivalente a 75% do tamanho em pontos de função assumindo complexidade baixa para as funções de dados ALI e AIE, e média para as funções transacionais EE, SE e CE. Ou seja, uma medição estimativa em elementos funcionais equivale a 75% de uma medição estimativa em pontos de função.

2.1.2.3. Contagem detalhada

A contagem detalhada se dá através da identificação, classificação e mensuração das funcionalidades no escopo da medição.

Para que uma funcionalidade seja considerada na contagem, ela deve estar formalmente registrada em artefatos indicados na metodologia de desenvolvimento do TCU como requisito da aplicação alvo da medição.

2.2. Identificação Do Escopo Da Medição E Da Fronteira Da Aplicação

2.2.1. Escopo da medição

O escopo define o conjunto de funções que serão alvo da medição e pode abranger uma ou mais aplicações.

2.2.2. Fronteira da aplicação

Na área de tecnologia da informação, o termo “aplicação” é usado, de modo geral, como sinônimo de programa executável pelo usuário. São exemplos: Word, Excel, Calculadora, Faturamento, Vendas, etc. O termo “aplicação” é ainda usado como sinônimo para módulo, componente, subsistema, sistema, sistema aplicativo ou sistema de informação. Os desenvolvedores costumam segmentar um conjunto de funções relacionadas em visões tecnológicas. São exemplos: plataforma física (computador de grande porte ou microcomputador) e arquitetura de projeto (web, cliente-servidor, etc.).

Mas isso não é uma aplicação. Uma aplicação tem que ser definida segundo a visão do usuário, de acordo com os requisitos de negócio. Não são consideradas questões técnicas voltadas à implementação física. Segundo o CPM, uma aplicação é um conjunto coeso de dados e procedimentos automatizados que suportam um objetivo de negócio, podendo consistir de um ou mais componentes, módulos ou subsistemas.

A correta identificação de uma aplicação (delimitada por sua fronteira) é fundamental para o emprego consistente da métrica, evitando-se contagens superdimensionadas ou subdimensionadas. A fronteira da aplicação pode ser entendida como a interface conceitual que delimita o software que será medido e seus usuários. O posicionamento incorreto da fronteira pode alterar a perspectiva da medição de uma visão lógica (visão funcional) para uma visão física. As principais consequências disso são a contagem duplicada de transações e arquivos de dados, a contagem incorreta de funções de transferência de dados e dificuldade na contagem de arquivos. Uma fronteira de aplicação não pode ser subdivida por contextos gerenciais de desenvolvimento, por exemplo, interno e externo ao órgão.

A fronteira é determinada com base na visão do usuário. O foco está no que o usuário pode entender e descrever. A fronteira entre aplicações relacionadas está baseada nas áreas funcionais separadas como pode ser visto pelo usuário, não em considerações técnicas. Observe dados de medição correlatos, tais como esforço, custo e defeitos. As fronteiras consideradas para medições funcionais e para os outros dados de medição tendem a ser as mesmas.

As aplicações são distribuídas em áreas de negócio do usuário, e pode existir uma ou mais aplicações em cada área de negócio. Compete à Secretaria que desenvolve sistemas no TCU definir e atualizar a lista de aplicações. Segue lista não exaustiva de aplicações existentes no TCU: Gestão de Aquisição, Guarda e Controle de Bens de Consumo e Patrimoniais (Adm - área administrativa); Gestão de Atos de Concessão (CE - área de controle externo); Gestão de Contratos (Adm); Gestão de Apreciações (CE); Gestão de Fiscalizações (CE); Gestão de Fundos de Participação Constitucionais (CE); Gestão de Jurisprudência (CE); Gestão de Pessoas

(área de infraestrutura - infra); Gestão de Processos e Documentos (CE), Gestão de Recursos Humanos (Adm); Gestão de Saúde Médica (Adm); Gestão de Tomadas de Contas Especiais (CE); Gestão do Portal (Infra).

2.3. Identificação Das Funções de Dados

As funções de dado representam as funcionalidades fornecidas ao usuário a fim de atender às suas necessidades de persistência de dados internos e externos à aplicação. São classificadas em arquivo lógico interno (ALI) e arquivo de interface externa (AIE).

O termo arquivo não significa um arquivo do sistema operacional, mas sim um grupo de dados logicamente relacionados, reconhecido pelo usuário. Um arquivo para a medição funcional pode estar mapeado em um ou mais arquivos físicos ou em tabelas do banco de dados e recebe a denominação de arquivo lógico.

Em casos de desenvolvimento, os ALIs e os AIEs só podem ser contados uma única vez para a fronteira da aplicação.

Em casos de melhoria, serão contados sempre que forem objetos de manutenção no escopo da medição.

A identificação dos arquivos lógicos deve seguir os seguintes passos:

- Identificação dos arquivos lógicos.
- Classificação de cada arquivo lógico como ALI ou AIE.

2.3.1. Identificação dos arquivos lógicos

Os requisitos de armazenamento, funcionais e não funcionais, de uma aplicação são classificados em dados de negócio, dados de referência e dados de código, conforme definição do CPM.

Devem ser descartados os dados de código, também chamados metadados, que são uma implementação de requisitos técnicos e não devem influenciar o tamanho funcional da aplicação. Contudo, não devem ser descartados os dados de referência, pois suportam regras de negócio enquanto que dados de código podem ter o código substituído pela respectiva descrição nos objetos de negócio em que são utilizados sem que o seu significado seja alterado.

É preciso avaliar como uma entidade candidata é utilizada pelas transações e a sua dependência em relação a outras entidades.

É importante verificar como os processos elementares da aplicação mantêm essas entidades. A inclusão e exclusão conjunta de determinado grupo de dados de entidades é um forte indicador que esse grupo deve ser considerado um único arquivo lógico. A alteração de dados normalmente está direcionada apenas para uma única entidade; consequentemente, ela não é uma orientação efetiva para agrupar entidades. Os processos elementares de extração que consultam essas entidades devem ser verificados bem como se essas entidades também são consultadas conjuntamente.

Entidades com alto grau de dependência também podem indicar um único arquivo lógico. Cada uma das entidades dependentes pode ser um tipo de registro a ser considerado na determinação da complexidade desse

arquivo lógico. Por exemplo, tanto a entidade Nota Fiscal quanto a entidade Itens da Nota isoladamente não são arquivos lógicos, contudo, em conjunto, elas são um arquivo lógico.

As abstrações identificadas na atividade de análise do processo de desenvolvimento do TCU são candidatas a arquivos lógicos em uma aplicação e devem passar pelo crivo das regras de identificação para a correta avaliação, conforme CPM.

2.3.2. Classificação de cada arquivo lógico como ALI ou AIE

A diferença básica entre um arquivo lógico interno (ALI) e um arquivo de interface externa (AIE) é que um AIE não é mantido pela aplicação sendo contada. O AIE está conceitualmente fora da fronteira da aplicação enquanto o ALI está dentro da mesma.

2.3.2.1. Regras de classificação de arquivo lógico interno

Para que determinada função seja identificada como um ALI, todas as regras seguintes devem ser válidas:

- O grupo de dados ou informações de controle é logicamente relacionado e identificável pelo usuário dentro do escopo da medição;
- O grupo de dados é mantido dentro da fronteira da aplicação sendo contada;
- Sua principal intenção é armazenar dados mantidos através de um ou mais processos elementares dentro da fronteira da aplicação sendo contada.

2.3.2.2. Regras de classificação de arquivo de interface externa

Para que determinada função seja contada como um arquivo de interface externa, todas as regras seguintes devem ser válidas:

- O grupo de dados ou informações de controle é logicamente relacionado e identificável pelo usuário dentro do escopo da medição;
- O grupo de dados é referenciado pela aplicação sendo contada, porém é externo a ela;
- O grupo de dados não é mantido pela aplicação sendo contada;
- O grupo de dados é mantido por outra aplicação, isto é, deve ser um ALI para outra aplicação;
- Sua principal intenção é armazenar dados referenciados através de um ou mais processos elementares que estiverem dentro da fronteira da aplicação sendo contada.

Conforme IFPUG (2015), no caso de dados retornados pela Aplicação A de múltiplos AIEs, para a Aplicação B, os campos (TD – Tipos de Dados) e subagrupamentos de dados (TR - Tipos de Registro) são determinados pela visão lógica dos dados do aplicativo B e os atributos realmente utilizados. Se este for considerado um grupo lógico de dados, ele é contado como um AIE no aplicativo B, independentemente da visão do aplicativo A. Se for considerado mais do que um grupo lógico de dados no aplicativo B, seria contado como mais de um AIE. O

CPM afirma que os dados devem ser "identificados em um ALI em uma ou mais outras aplicações". Não estipula que lá apenas seja um ALI para um AIE.

2.3.3. Regras de contagem de campos (TD - Tipos de Dado)

Na contagem de campos (TD) que atravessam a fronteira deve-se abstrair dos detalhes de armazenamento físico da informação, considerando-se sempre a visão do usuário, o reconhecimento por parte do usuário do tipo de dados em questão. As seguintes regras devem ser válidas para contagem de tipos de dados:

- Deve-se contar cada campo único reconhecido pelo usuário e não repetido, mantido ou recuperado de um ALI ou AIE por meio da execução de um processo elementar.
- Quando duas aplicações mantêm ou referenciam o mesmo ALI/AIE, devem ser contados apenas os campos utilizados pela aplicação em análise.
- Deve-se contar cada campo solicitado pelo usuário para estabelecer um relacionamento com outro arquivo lógico (ALI ou AIE).

As considerações a seguir assumem que os tipos de dados são reconhecidos pelo usuário, não repetidos e mantidos por algum processo elementar:

- Campos do tipo data devem ser contados como um único tipo de dado, mesmo que estejam separados em múltiplos campos (dia, mês e ano).
- Uma imagem anterior a uma atualização de um grupo de “n” campos mantidos para propósitos de auditoria é contada como um tipo de dado da imagem anterior e “n” tipos de dados para os campos, totalizando assim “n+1” tipos de dados. No caso da solução de auditoria utilizada pelo TCU, também são contados n+1 tipos de dados, pois conceitualmente equivale à mesma abordagem, i.e., apesar de ter 2n campos físicos, os dados anteriores poderiam estar contidos em uma única imagem do conjunto de itens de dados anteriores à atualização.
- Campos calculados e armazenados em um ALI também devem ser contados como tipos de dados.
- Campos do tipo *timestamp* devem ser contados como tipos de dados.
- Caso a chave estrangeira seja composta por vários campos, cada um deles deve ser contado como um tipo de dado.
- Quando um único arquivo lógico é composto por mais de uma tabela no banco de dados, a chave estrangeira usada para estabelecer o relacionamento entre estas tabelas não deve ser contada mais de uma vez como tipo de dado.
- Os dados de código não devem ser contados como tipos de dados. Os dados de referência devem ser contados normalmente como tipos de dados.

2.3.4. Regras de contagem de subagrupamentos (TR - Tipos de Registro)

As seguintes regras devem ser utilizadas para determinar o número de subagrupamentos de dados (tipos de registro - TR) de um ALI ou AIE.

- Deve-se contar um tipo de registro para cada subgrupo (ou subtipo), obrigatório ou opcional, de um ALI ou AIE.
- Se não houver nenhum subgrupo, deve-se contar o próprio ALI ou AIE como um tipo de registro.

2.3.5. Considerações para funções de dado em casos de melhoria

Uma função de dado é considerada modificada e integrante do escopo de medição de um caso de melhoria se ela for modificada em sua estrutura, ou seja, campos devem ser acrescentados, excluídos ou terem algum atributo alterado. A seguir, são apresentados os procedimentos corretos para algumas situações bem comuns.

- Se a mudança envolve apenas a alteração dos dados armazenados em um arquivo, não se pode considerar que o arquivo foi alterado em sua estrutura, não sendo contado no caso de melhoria.
- Se um campo foi adicionado a um ALI ou AIE, e ele não é mantido ou referenciado na aplicação, então não houve alteração desse arquivo dentro da aplicação alvo da medição. Para confirmar se o campo é utilizado na aplicação ou não, procure alguma função transacional que tenha sido criada ou alterada para manipular esse campo.
- Se uma aplicação passa a manter ou referenciar um campo já existente e que antes não era utilizado, então se considera que o ALI ou AIE foi alterado para essa aplicação (mesmo que não haja nenhuma alteração física no arquivo).
- Se um campo é adicionado, alterado ou excluído de um ALI ou AIE pertencente a várias aplicações e elas referenciam ou mantêm o campo, essa alteração de funcionalidade é contada para cada uma das aplicações.
- Se um arquivo físico ou tabela foi criado pelo caso de melhoria, não necessariamente resultará em um novo ALI ou AIE. Essa tabela pode ser também um novo tipo de registro em um ALI ou AIE existente. Ou também pode não representar nada do ponto de vista do usuário. Devem ser revisadas sempre as regras de identificação das funções de dado.

2.4. Identificação Das Funções Transacionais

As funções transacionais representam as funcionalidades de processamento de dados fornecidas pela aplicação ao usuário. São processos elementares e únicos. Transações semelhantes, que são constituídas do mesmo processo elementar, devem ser consideradas instâncias de uma única função transacional, devendo ser contadas uma única vez dentro de uma aplicação.

As funções transacionais são classificadas em entradas externas, saídas externas e consultas externas.

2.4.1. Regras para determinar se um processo elementar é único

Em todas as funções transacionais, para determinar se devemos contar mais de um processo, uma ou mais das três proposições devem obrigatoriamente ser verdadeiras:

- A lógica de processamento é diferente da executada por outros processos elementares da aplicação. A exceção é quanto à ordenação, ou seja, dois relatórios, cuja diferença seja apenas a ordenação dos dados, constituem um único processo elementar;
- O conjunto de tipos de dados identificado é diferente do identificado para outros processos elementares da aplicação;
- Os ALI e AIE referenciados são diferentes dos arquivos referenciados por outros processos elementares da aplicação.

2.4.2. Regras de identificação de entrada externa (EE)

Para que uma função transacional seja classificada como entrada externa, ela deve atender a todas as regras abaixo:

- Ser um processo elementar;
- Processar dados ou informações de controle originadas fora da fronteira da aplicação;
- Ter como principal intenção manter um ou mais arquivos lógicos internos e/ou alterar o comportamento da aplicação.

2.4.3. Regras de identificação de saída externa (SE)

Para que uma função transacional seja classificada como saída externa, ela deve atender a todas as regras abaixo:

- Ser um processo elementar;
- Enviar dados ou informações de controle para fora da fronteira da aplicação;
- Ter como principal intenção apresentar informações ao usuário através de lógica de processamento que não seja apenas uma simples recuperação de dados ou informações de controle. Sua lógica de processamento deve obrigatoriamente conter cálculo, ou criar dados derivados, ou manter um arquivo lógico interno, ou alterar o comportamento da aplicação.

2.4.4. Regras de identificação de consulta externa (CE)

Para que uma função transacional seja classificada como consulta externa, ela deve atender a todas as regras abaixo:

- Ser um processo elementar.
- Enviar dados ou informações de controle para fora da fronteira da aplicação.

- Ter como principal intenção apresentar informações ao usuário através da simples recuperação de dados ou informações de controle de ALIs e/ou AIEs. Sua lógica de processamento não deve conter fórmula matemática ou cálculo, tampouco criar dados derivados. Nenhum ALI pode ser mantido durante seu processamento, nem o comportamento da aplicação pode ser alterado.

2.4.5. Regras de contagem de campos (TD – Tipos de Dado)

As seguintes regras devem ser válidas na contagem de campos (TD - Tipos de Dados):

- Deve-se contar um tipo de dado para cada campo, não repetido e reconhecido pelo usuário, que entra ou sai pela fronteira da aplicação e necessário à conclusão do processo;
- Se um campo tanto entra quanto sai pela fronteira da aplicação, deve ser contado uma única vez;
- Os campos que durante o processo elementar são recuperados ou derivados pela aplicação e armazenados em um ALI, mas não atravessam a fronteira da aplicação, não devem ser contados como tipos de dados;
- Deve-se contar um único tipo de dado para a capacidade de envio para fora da fronteira da aplicação de uma mensagem de resposta da aplicação, indicando um erro verificado durante o processamento, a confirmação da sua conclusão ou a verificação de seu prosseguimento;
- Deve-se contar um tipo de dado para a capacidade de especificar uma ação a ser tomada, mesmo que haja múltiplos meios de ativar o mesmo processo, deve ser contado apenas um tipo de dado;
- Não devem ser contados literais, como título de relatórios, cabeçalhos, etc., como tipo de dados;
- Não devem ser contadas variáveis de paginação ou campos automáticos gerados pela aplicação.

2.4.6. Regras de contagem para arquivo referenciado (AR)

As seguintes regras devem ser válidas na contagem de um arquivo referenciado. As duas primeiras, que tratam da atualização de arquivos, não são aplicáveis para consultas externas.

- Deve-se contar um arquivo referenciado para cada ALI mantido;
- Deve-se contar apenas um arquivo referenciado para cada ALI que seja tanto mantido quanto lido;
- Deve-se contar um arquivo referenciado para cada ALI ou AIE lido durante o processamento.

2.4.7. Considerações para funções transacionais em casos de melhoria

Uma função transacional é considerada modificada e integrante do escopo de medição de um caso de melhoria quando há alteração em alguns dos seguintes itens:

- Tipos de dados: se eles foram adicionados, excluídos ou alterados da função. Se houve alteração apenas de elementos visuais, como literais, cores e formatos, não se considera que a função foi alterada.
- Arquivos referenciados: se eles foram adicionados, excluídos ou alterados pela função.
- Lógica de processamento: uma transação pode ter várias lógicas de processamento, basta que uma delas seja alterada, excluída ou adicionada para que se considere a função como modificada. Embora a ordenação seja a única lógica de processamento que não é suficiente para determinar a unicidade de uma transação, sua alteração também determina uma alteração na função.

2.5. Mensuração Do Tamanho Funcional De Cada Função

Identificadas as funcionalidades que integram o escopo da medição, o próximo passo é derivar efetivamente o tamanho funcional associado a cada função.

2.5.1. Mensuração do tamanho funcional em elementos funcionais e suas submétricas

A métrica Elementos Funcionais (EF) proposta por Castro e Hernandes (2013) baseia-se nos conceitos da métrica Pontos de Função, mas sem algumas de suas falhas conhecidas relativas à forma de cálculo. A métrica foi construída em pós-graduação patrocinada pelo TCU de servidor da Casa junto à Universidade de Brasília. O trabalho foi selecionado no 27º Simpósio Brasileiro de Engenharia de Software em outubro de 2013 e publicado tanto na base digital do *Institute of Electrical and Electronic Engineers* (*IEEE Xplore*) quanto na edição de número 135 da Revista do TCU em 2016. E foi premiado internamente no TCU com o prêmio de Trabalho Inovador no ano de 2015.

Cada tipo de funcionalidade tem sua fórmula de cálculo (ver tabela abaixo) baseada nos quantitativos de atributos funcionais: AR (Arquivos Referenciados), TD (Tipo de Dado) e TR (Tipo de Registro). Em caso de manutenção evolutiva, deve-se considerar os atributos funcionais afetados pela alteração (conceito aplicado também na Nesma). Em caso de exclusão de uma funcionalidade, o tamanho assume o valor da constante, uma vez que não há atributos especificamente impactados por essa operação. Tanto desenvolvimentos quanto manutenções de software são medidos da mesma forma.

Funcionalidade	Fórmula derivada
ALI	EFd = 1,75 + 0,96*TR + 0,12*TD
AIE	EFd = 1,25 + 0,65*TR + 0,08*TD
SE	EFt = 1,00 + 0,81*AR + 0,13*TD
EE	EFt = 0,75 + 0,91*AR + 0,13*TD
CE	EFt = 0,75 + 0,76*AR + 0,10*TD

Tabela 1 - Fórmulas de cálculo de elementos funcionais por tipo de funcionalidade

A métrica Elementos Funcionais, EF, resulta da soma dos Elementos Funcionais de Transação, EFt, com os Elementos Funcionais de Dados, EFd. Essas submétricas representam respectivamente os elementos funcionais associados a transações (CE, EE e SE) e a dados (AIE e ALI).

2.5.2. Mensuração do tamanho funcional em pontos de função

2.5.2.1. Mensuração em pontos de função de desenvolvimento de funcionalidade

Para se dimensionar o desenvolvimento de uma funcionalidade em pontos de função é necessário primeiro classificar a sua complexidade e depois derivar, conforme tabela abaixo, o número de pontos de função a partir das complexidades identificadas.

Funções	**Complexidade**		
	Baixa	**Média**	**Alta**
AIE	5 FP	7 FP	10 FP
ALI	7 FP	10 FP	15 FP
CE	3 FP	4 FP	6 FP
EE	3 FP	4 FP	6 FP
SE	4 FP	5 FP	7 FP

Tabela 2 – Pontos de função por complexidade e tipo de função

2.5.2.1.1. Classificação da complexidade de funções de dado

Cada função de dado é classificada com relação à sua complexidade em baixa, média e alta. A complexidade das funções de dado é determinada pela quantidade de tipos de dados (campos) e tipos de registro (subgrupos de dados dentro do arquivo) visíveis ao usuário na fronteira da aplicação.

A tabela a seguir deve ser usada para derivação da complexidade das funções de dado.

Quantidade de tipos de registro (TR)	**Quantidade de tipos de dados (TD)**		
	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

Tabela 3 – Complexidade das funções de dado

2.5.2.1.2. Classificação da complexidade de funções transacionais

Cada função transacional é classificada com relação à sua complexidade em baixa, média e alta. As funções transacionais têm sua complexidade determinada pela quantidade de tipos de dados (campos) e arquivos referenciados (ALI ou AIE), conforme tabelas apresentadas a seguir.

As tabelas a seguir devem ser usadas para derivação da complexidade das transações de acordo com seu tipo.

Quantidade de arquivos referenciados (AR)	Quantidade de tipos de Dados (TD)		
	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

Tabela 4 – Complexidade das entradas externas

Quantidade de arquivos referenciados (AR)	Quantidade de tipos de Dados (TD)		
	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

Tabela 5 – Complexidade das saídas externas e consultas externas

2.5.2.2. Mensuração em Pontos de função de manutenção evolutiva de funcionalidade

O dimensionamento de pontos de função em um caso de melhoria de uma funcionalidade (no contexto deste manual, o termo 'melhoria' significa mudança funcional) baseia-se nas regras definidas pela Nesma, versão 1.0, que define a unidade ponto de função de melhoria (EFP - *Enhancement Function Point*). Valem as definições deste manual em caso de conflito com a Nesma. Assim, um ponto de função (PF ou FP) será equivalente a um ponto de função de melhoria (EFP).

2.5.2.2.1. Dimensionamento de EFP em funcionalidades incluídas

Segundo a Nesma, as funcionalidades incluídas em um caso de melhoria serão dimensionadas usando o método FSM padrão do IFPUG. Ou seja, o fator de impacto será 1.

EFPinc = FPinc * 1

Em que:

EFPinc – pontos de função de melhoria da funcionalidade envolvida

FPinc – pontos de função da funcionalidade envolvida

2.5.2.2.2. Dimensionamento de EFP em funcionalidades de conversão de dados

Segundo a Nesma, as funcionalidades de conversão em um caso de melhoria serão dimensionadas usando o método FSM padrão do IFPUG. Ou seja, o fator de impacto será 1.

EFPconv = FPconv * 1

Em que:

EFPconv – pontos de função de melhoria da funcionalidade de conversão envolvida

FPconv – pontos de função da funcionalidade de conversão envolvida.

2.5.2.2.3. Dimensionamento de EFP em funcionalidades excluídas

Para funções excluídas, um fator de impacto de 0,4 é usado. O número de pontos de função de melhoria para uma única função excluída é determinado da seguinte forma:

EFPexc = 0,4 * FPexc

Em que:

EFPexc – pontos de função de melhoria da funcionalidade excluída

FPexc – pontos de função da funcionalidade excluída

2.5.2.2.4. Dimensionamento de EFP em funcionalidades alteradas

2.5.2.2.4.1. Funções de dados

As funções de dados que mudam são identificadas e o tamanho de cada função de dados após a mudança é determinado.

Para funções de dados que mudem estruturalmente, um fator de impacto é calculado a partir da porcentagem de elementos de dados mudados. A porcentagem de mudança é definida como a razão definida pelo número de elementos de dados modificados dividido pelo número de elementos de dados originais:

$$\text{Porcentagem de mudança} = \frac{\text{Número de TDs incluídos/alterados/excluídos x 100}}{\text{Número de TDs na função de dados original}}$$

O fator de impacto FI é obtido da tabela abaixo usando a porcentagem de mudança em número de elementos de dados:

Porcentagem de TDs	<=33%	<=67%	<=100%	>100%
Fator de impacto (FI)	0,25	0,50	0,75	1,00

Tabela 6 – Fatores de impacto para funções de dados

Se uma função de dados mudar de tipo (por exemplo, um AIE se tornar um ALI), um valor de 0,4 é usado para o fator de impacto.

Mudanças de tipo precisam ser avaliadas também para identificar mudanças no número de elementos de dados. Se o número de elementos de dados mudar juntamente com o tipo, o fator de impacto devido à mudança no número de elementos de dados deve ser determinado. O valor do fator de impacto devido à mudança no tipo é comparado com aquele devido à mudança no número de elementos de dados e o maior valor é usado no cálculo dos pontos de função de melhoria.

Se um AIE ou um ALI for dividido em duas (ou mais) funções de dados, uma função de dados excluída e duas (ou mais) adicionadas são contadas.

Se um AIE e um ALI são combinados, duas funções de dados excluídas e uma função de dados adicionada são contadas.

EFPalt = FI * FPalt

Em que:

EFPalt – pontos de função de melhoria da funcionalidade alterada

FI – fator de impacto

FPalt – pontos de função da funcionalidade alterada

2.5.2.2.4.2. Funções transacionais

As funções transacionais que mudam são identificadas e o tamanho de cada transação após a mudança é determinado.

Uma função transacional é considerada mudada se ela é alterada de alguma forma, mas mantém o mesmo nome e propósito tanto após a melhoria quanto antes da melhoria. O padrão EFPA da Nesma é usado para determinar o tamanho da transação após a mudança.

O processo é o definido a seguir:

- Identificar os elementos de dados e arquivos lógicos usados pela transação;
- Determinar as porcentagens de elementos de dados e arquivos lógicos referenciados mudados como resultado da melhoria:

$$\text{Porcentagem de TDs} = \frac{\text{Número de TDs incluídos/alterados/excluídos x 100}}{\text{Número de TDs na função de transação original}}$$

$$\text{Porcentagem de ALRs} = \frac{\text{Número de ALRs incluídos/alterados/excluídos x 100}}{\text{Número de ALRs na função de transação original}}$$

- Determinar o fator de impacto para a transação

Mudança:	**Porcentagem de TDs**		
Porcentagem de ALRs	<=67%	<=100%	>100%
<=33%	0,25	0,50	0,75
<=67%	0,50	0,75	1,00
<=100%	0,75	1,00	1,25
>100%	1,00	1,25	1,50

Tabela 7 – Fatores de impacto para funções transacionais

- Calcular o número de pontos de função da melhoria

  EFPalt = FI x FPalt

  Em que:

  EFPalt – pontos de função de melhoria da funcionalidade alterada

  FI – fator de impacto

  FPalt – pontos de função da funcionalidade alterada

2.5.3. Reflexão sobre a escolha da métrica funcional

A métrica Pontos de Função apresentada na seção 2.5.2 (aplicando a métrica Nesma para manutenções evolutivas) é um padrão mundial de mercado e isso não pode ser ignorado. Mas a métrica Elementos Funcionais apresentada na seção 2.5.1 baseia-se nesses conceitos padronizados pela métrica de Pontos de Função e foi reconhecida pelo meio acadêmico e pelo TCU (ser seção 2.5.1). Além disso, traz algumas vantagens que justificam seu uso:

. exige menor esforço de medição de manutenção, por não exigir detalhamento do tamanho anterior da função (número de arquivos referenciados e campos).

. apresenta regra simples de cálculo. A proporção do impacto preconizado pela Nesma segue uma regra complexa, que leva em conta o percentual de campos impactados e, no caso de transações, também o percentual de arquivos referenciados impactados. Essa complexidade maior pode se traduzir em erros de cálculo em planilhas.

. mede tamanho absoluto, ou seja, não relativo ao tamanho da função de origem, dependendo apenas dos itens impactados. Para ilustrar essa situação, supondo uma alteração hipotética de uma entrada externa com 3 PF que tinha originalmente apenas 1 campo (TD) passando pela fronteira e não referenciava nenhum arquivo (AR) e que passa a apresentar 2 novos campos e a acessar um novo arquivo. Aplicando Nesma essa alteração mediria 150% do tamanho original, ou seja: 4,5 PF. Mas, se a função tivesse outra quantidade de campos e de arquivos acessados, por exemplo 2 campos e 3 AR, a mudança seria dimensionada como 25% do tamanho original e teria um tamanho bem inferior, 1 PF (pois a função original mediria 4 PF). Ou seja, uma mesma alteração, acrescentando 1 TD e 1 AR, poderia ser dimensionada, no exemplo hipotético, com divergência de tamanho de 350% (4,5 PF x 1 PF). Já em elementos funcionais o valor para o exemplo hipotético seria 1,79 EF, independentemente do tamanho da função de origem.

. dimensiona da mesma forma desenvolvimentos e manutenções. Tudo bem que o custo (esforço, por exemplo) tende a ser diferente nesses casos, mas essa diferença deve ser aplicada como um fator de ajuste na correlação com o custo e não no tamanho do objeto medido. Exemplo: considere pintar uma “parede nova” de 5 m2 ou pintar 2 m2 de uma parede já existente. O custo pode variar, não só por ser nova a parede, mas também por outros fatores, como padrão de acabamento, necessidade ou não de impermeabilização, etc. Ajustar o tamanho da área impactada pela aplicação de um fator que influencia o esforço não é uma iniciativa adequada. No caso da parede, seria o equivalente a mudar a área a ser mantida hipoteticamente para 25%, ou seja, 0,5

m2. Melhor aplicar esse fator (e outros que precisam ser considerados) diretamente na derivação do custo (esforço) e não maquiar o tamanho do objeto.

. não apresenta nenhuma das 5 deficiências de cálculo citadas por Castro e Hernandes (2013): baixa representatividade, funcionalidades com complexidades diferentes dimensionadas com o mesmo tamanho, transição abrupta entre faixas, dimensionamento limitado de funcionalidades com alta complexidade e operação em escala ordinal (o que impediria na teoria, cálculo de médias, produtividade, etc).

. demonstra uma melhor correlação com o esforço (Castro e Hernandes; 2013).

2.6. Aplicação Dos Percentuais Por Atividade Implementada

O tamanho funcional será cheio quando forem implementadas todas as disciplinas do ciclo de desenvolvimento. Caso contrário deve-se reduzir do tamanho funcional apurado os percentuais relativos às disciplinas não tratadas. Caso não haja uma tabela específica na metodologia de desenvolvimento, deve-se seguir a tabela abaixo:

Disciplina realizada	**% Atividade**	**% Gestão de projeto**	**% Total**
Engenharia de requisitos	22,5	2,5	25
Design/Arquitetura	9	1	10
Implementação	36	4	40
Testes	22,5	2,5	25
Desenvolvimento completo	**90**	**10**	**100**

Tabela 8 – Percentuais por atividade

2.7. Dimensionamento Final

Para se dimensionar uma Aplicação ou Casos de Desenvolvimento e Casos de Melhoria, deve-se somar os tamanhos funcionais apurados nos passos anteriores de todas as funcionalidades envolvidas no escopo da contagem.

3. Documentação de Medições

Todo dimensionamento deve ser detalhado nos relatórios de medição ou em um sistema de informação para registrar as medições.

Cada caso de desenvolvimento ou de melhoria terá seu próprio relatório, que registrará as funcionalidades e atributos envolvidos.

3.1. Nomenclatura

A utilização do jargão do negócio deve ser cuidadosamente observada para não ser confundido com os hábitos de linguagem do analista de desenvolvimento.

3.1.1. Padrão para nomenclatura de funções de transação

O nome do processo elementar deverá, sempre que possível, ser composto do conceito envolvido e do objetivo da transação (verbo no infinitivo): “conceito – verbo” ou “verbo conceito”. Apenas a inicial da primeira palavra deverá estar com letra maiúscula.

Exemplos: Tramitação de processo - incluir ou Incluir tramitação de processo ou Tramitar Processo

3.1.2. Padrão para nomenclatura de tipos de registro

Deve representar o conceito. Se for um subtipo, pode-se usar expressão “é um”; se for agrupamento de campos, pode-se usar “tem”.

Exemplos:

Suponhamos que um arquivo lógico Ato tenha como subtipo Ato de Admissão. Nesse caso, o nome do registro lógico seria “É um ato de admissão” ou mesmo “Ato de admissão”.

Suponhamos que um arquivo lógico Processo tenha um agrupamento com campos de tramitação do processo. Nesse caso, o nome do registro lógico poderia ser “Tem tramitação” ou “Tramitação”.

3.2. Itens Exigidos Em Uma Contagem Detalhada

É obrigatória a referência à documentação das funcionalidades, fazendo-se distinção da versão do documento.

Essa documentação pode ser um caso de uso ou outro padrão adotado na metodologia de desenvolvimento. Deve conter uma listagem detalhada, aprovada pelos usuários, dos campos de cada tela exibida pelo sistema, bem como a descrição das principais ações, regras de negócio e outras particularidades dessa tela. Caso a lógica envolva uma sequência de telas, informar na primeira tela toda a lógica envolvida nas telas referenciadas. Um cuidado especial deve ser tomado ao registrar a vinculação da execução das ações entre as várias telas exibidas para se evitar erros. Se houver, deverá ser fornecida também uma cópia (*print screen*) de cada tela.

É necessário identificar, no mínimo, as funcionalidades e os seus itens funcionais (AR, TR e TD). Cada item funcional necessário para o cálculo do tamanho funcional deve ser identificado pelo nome. A documentação de uma ordem de serviço (ou instrumento equivalente) deve identificar as funções envolvidas no caso (melhoria ou desenvolvimento) e referenciar os detalhes das alterações em cada funcionalidade.

Em anexo constam modelos ilustrativos de leiautes de contagem. Podem ser usados outros leiautes, desde que sejam detalhadas as informações necessárias para a contagem e para possíveis auditorias, conforme esta seção. A área responsável por medições do TCU possui modelos de planilhas excel que podem ser usadas.

4. Itens Não Mensuráveis

Esta seção avalia a possibilidade de se derivar um número que deve ser somado ao tamanho funcional da medição para casos que não se encaixam claramente no processo de medição deste Manual. É uma forma comum no mercado para se tratar atividades que geram esforço considerável.

Em todos os casos citados nesta seção, deve-se ter claro que o valor apurado refere-se à contratação de todas as disciplinas. Se alguma disciplina não for contratada para a demanda, retira-se o percentual correspondente à disciplina no cálculo do tamanho especificado no passo “Aplicação dos percentuais por atividade implementada” do processo de medição. Ou seja, só as fases contratadas devem ser pagas.

É necessária a comprovação técnica de que o código da funcionalidade foi impactado (adaptado) para que haja a contabilização, exceto no caso de contratação apenas de documentação. Por exemplo, o caso de se transformar uma *view* em materializada (sem outras mudanças) não implica adaptações em transações, ainda que essas tenham alcançado uma melhor performance. Contudo, em caso de alteração do nome de uma *view*, os códigos podem realmente ser adaptados e, nesse caso, impactados.

4.1. Manutenção Corretiva Em Sistema Legado

Quando o sistema não tiver sido desenvolvido pela contratada, deve-se calcular o tamanho da manutenção corretiva e aplicar um fator de ajuste redutor para 75%.

4.2. Documentação De Sistema

Demandas de criação de documentação a partir do zero de um sistema já existente devem ser dimensionadas considerando o percentual da atividade de documentação sobre o tamanho funcional cheio das funcionalidades, sem aplicação de fatores de redução.

4.3. Atualização De Plataforma

Nesta categoria encontram-se as demandas de conversão de funcionalidades face a atualização de versão de linguagem de programação ou de navegador ou de banco de dados. Para serem consideradas, as funcionalidades precisam ter seu código modificado. As funções de dados não devem ser contadas. As funcionalidades impactadas pela conversão devem ser tratadas como mantidas e ao tamanho funcional apurado deve-se aplicar um fator de ajuste redutor para 30%.

4.4. Manutenção Cosmética

A manutenção em interface, denominada na literatura manutenção cosmética, é associada às demandas de alterações de interface em funções transacionais, por exemplo, fonte de letra, cores de telas, logotipos, mudança de botões na tela, mudança de posição de campos ou texto na tela. Também se enquadram nessa categoria as mudanças de texto em mensagens de erro, validação, aviso, alerta ou conclusão de processamento. No caso de mudanças em elementos de interface que se repetem em várias telas, tais como folhas de estilo, será feito o pagamento equivalente a apenas uma mudança. Deve-se contabilizar uma manutenção por funcionalidade impactada.

Para cada manutenção, conta-se: 0,6 PF ou 0,45 EF

Obs.: em PF, equivale a 20% da contagem de uma função transacional de mais baixa complexidade (3 PF).

4.5. Adaptação De Funcionalidades Sem Atualização De Requisitos Funcionais

São consideradas nesta categoria as demandas de manutenção adaptativa associadas a solicitações que envolvem aspectos não funcionais, sem alteração em requisitos funcionais. Por exemplo: replicação de funcionalidade (chamar uma consulta existente em outra tela da aplicação), replicação de base de dados ou criação de base temporária para resolver problemas de *performance* ou segurança, alteração na aplicação para adaptação às alterações realizadas na interface com rotinas de integração com outros *softwares* (ex: alteração em sub-rotinas chamadas por este *software*).

As funcionalidades impactadas pela conversão devem ser tratadas como mantidas e ao tamanho funcional apurado deve-se aplicar um fator de ajuste redutor para 75%.

Por exemplo, no caso de adaptação para fins de performance de uma funcionalidade existente, sem alteração funcional, considerando que o trabalho envolva todas as disciplinas, o cálculo ficará assim: 0,75 x tamanho funcional apurado da manutenção. Nesse caso, como se trata apenas de uma alteração de lógica de processamento, o tamanho funcional equivale a 25% do tamanho original da função. Logo, nesse caso, pode-se considerar 18,75% do tamanho original da função.

4.6. Múltiplas Mídias

Considerando-se a contagem funcional de funcionalidades entregues em mais de uma mídia, a aplicação das regras de contagem funcional definidas no CPM tem levado a duas abordagens alternativas, a saber: *single instance* e *multiple instance*.

A abordagem *single instance* considera que a entrega de uma função transacional em múltiplas mídias não deve ser utilizada na identificação da unicidade da função. A abordagem *multiple instance* leva em consideração que a mídia utilizada na entrega da funcionalidade é uma característica de identificação da unicidade da função. Assim, funcionalidades únicas são reconhecidas no contexto da mídia na qual elas são requisitadas para operar.

É importante enfatizar que o IFPUG reconhece ambas as abordagens, *single instance* e *multiple instance*, para a aplicação das regras definidas no CPM. As estimativas e contagens de FP abordadas neste manual serão baseadas em *single instance*.

Contudo, havendo mudança de código para a implementação da funcionalidade em outras mídias, considera-se uma alteração de função. Por exemplo, a solicitação de geração de arquivo .xls de uma consulta existente (com impacto no código) deve ser contada como alteração de uma função pré-existente (mas não como uma nova funcionalidade).

4.7. Primeira Referência A Arquivos Lógicos Em Contextos Diferentes De Desenvolvimentos

Em contratos externos, um arquivo lógico já existente em uma aplicação alvo de manutenção evolutiva poderá ser contado na primeira ordem de serviço em que o arquivo for referenciado pela contratada, desde que haja esforço que justifique a contagem. Deve-se considerar somente os itens tratados no escopo da contagem (TR e TD). Deverão ser aplicados os percentuais por disciplina em que houver esforço comprovado para cada funcionalidade.

Referências futuras ao arquivo não deverão ser contabilizadas, ainda que contemplem novos itens (TR e TD) no contexto da contratada, exceto se houver mudança estrutural do arquivo lógico em relação à aplicação como um todo.

4.8. Dados De Código

Os dados de código, conforme definido pelo CPM, não serão contados mesmo que estejam definidos nos requisitos do usuário. A contagem de dados de código acarretaria graves distorções na contagem funcional, bem como na estimativa de esforço e prazo. O esforço para seu desenvolvimento é muito menor em relação a requisitos funcionais e seu custo deverá estar inserido no valor acordado no contrato com terceiros.

4.9. Criação E Alteração De Scripts De Atualização Do Siga

A carga inicial dos dados de configuração do SIGA (Sistema De Gerenciamento De Acesso que contempla itens de menu, perfil de acesso, etc) faz parte do processo de desenvolvimento e é pré-requisito de implantação de uma aplicação ou mesmo de uma nova funcionalidade e não é alvo de contabilização.

Porém, acontece de usuários solicitarem alterações nessas informações (não funcionais para a aplicação em desenvolvimento) sem associação com mudanças funcionais da aplicação. Essas alterações são implementadas em forma de scripts, que executam funcionalidades pré-existentes na aplicação Siga com a passagem de novos valores como parâmetros. Em princípio, não deveriam ser contabilizados. Porém, dado o esforço alocado, poderão ser contabilizados da seguinte forma:

Criação do script pela primeira vez na aplicação: 3 PF ou 2,25 EF.

Alteração de script (estrutura ou lógica): 0,75 PF ou 0,56 EF (não se encaixa neste ponto a simples mudança de valores passados como parâmetros. Nesse caso poderia ser contabilizado como uma reexecução de script, conforme seção deste Manual).

4.10. Reexecução De Scripts De Atualização E De Carga De Dados

Demandas de reexecução de scripts de atualização e de carga de dados poderão ser dimensionadas com 0,1 PF ou 0,07 EF.

5. Aplicação da Métrica em Contextos Específicos de Desenvolvimento

5.1. Medição De Desenvolvimento Por Parametrização

O desenvolvimento por parametrização é uma das formas utilizadas para a adaptação de um software adquirido à realidade dos processos de trabalho do contratante. Esse tipo de desenvolvimento é caracterizado pela ativação de funcionalidades pré-existentes pela simples entrada de dados em uma interface administrativa para a definição de novos campos, regras de validação, metadados e outras características que não envolvem customização por meio de linguagens de programação.

As funcionalidades adaptadas por esse método devem ser medidas como melhorias em funcionalidades pré-existentes no software adquirido.

Assim, se uma função, que não seja identificada como dados de código ou transações sobre dados de código, que originalmente possua uma determinada estrutura em seu estado original cuja ativação é solicitada mediante a inclusão de alguns itens de dados, exclusão de outros e mudança de nome de campos e validações, será medida pelo método descrito neste manual.

No caso da simples ativação de uma funcionalidade pré-existente, realizado por um usuário intermediário administrador, deve-se considerar para fins de dimensionamento o equivalente a 25% da funcionalidade original que for ativada pela primeira vez, os quais não serão contados em nenhuma circunstância, mesmo que esteja(m) presente(s) nos requisitos do usuário.

O desenvolvimento por parametrização e customização ao mesmo tempo, desde que utilize linguagens de programação, é outro método de se adaptar um software adquirido às funcionalidades solicitadas pelo contratante e sua contagem deve seguir as mesmas regras de desenvolvimento de um sistema de informação tradicional contidas neste manual.

5.2. Medição De Portais Web

O desenvolvimento de portais pode utilizar linguagens de programação para construir o núcleo da camada de negócios. Nesse caso, são utilizadas as regras contidas neste manual para medição de sistemas de informação tradicionais.

Outra possibilidade no desenvolvimento de funcionalidades de portais web é a parametrização por meio de arquivos de configuração XML que especificam o que deve ser exibido ou não em determinada funcionalidade previamente especificada. Cada funcionalidade alterada por parametrização e disponibilizada para o usuário final em uma instância distinta não é considerada uma nova funcionalidade, mas uma extensão de uma versão anterior de uma funcionalidade especificada e, nesse caso, os itens de dados alterados/incluídos/excluídos por meio dos arquivos de configuração são contados.

A organização de dados para apresentação por meio de XSLT e Javascript, apesar de atuarem apenas em uma das camadas do portal, deve ser considerada como alteração nas funcionalidades presentes nas interfaces modificadas por esses meios e, portanto, as funcionalidades alteradas são contadas.

A simples ativação de funcionalidades pré-existentes deve ser dimensionada conforme descrito neste manual quando um usuário intermediário administrativo utiliza um recurso técnico para essa finalidade.

Alterações de layout e estilo da apresentação são itens não mensuráveis de acordo com o CPM e considerados como manutenções cosméticas. No entanto, cada arquivo HTML que sofrer alterações individuais que não se repetem em outras páginas, deve ser contado conforme seção de manutenção cosmética deste manual. No caso de arquivos CSS que contenham as definições de formatação utilizadas por várias páginas web e de partes de um arquivo HTML que são incluídas em várias outras páginas, deve-se contar apenas um arquivo mantido, não importando o número de páginas que forem afetadas. Isso ocorre frequentemente com barras de navegação, cabeçalhos e outros elementos que se repetem em várias páginas web.

5.3. Medição De Componentes De Software

Componentes são funcionalidades implementadas normalmente como serviços SOA, webservices ou microserviços. São exemplos de componentes: um serviço rest de validação de CPF e um serviço middleware com a capacidade funcional de promover conversão de protocolo e de acionar um ou mais serviços pré-existentes.

5.3.1. Quem ou o que é o usuário?

O usuário de um componente pode ser além de uma pessoa ou aplicação que interage com o sistema, um desenvolvedor que define a solução visando a um melhor reaproveitamento.

5.3.2. Qual a definição que deve ser utilizada para “componente”?

Um componente pode ser definido sob a perspectiva dos desenvolvedores, ou sob a perspectiva dos usuários que necessitam dele. As suas funcionalidades podem não estar sempre visíveis ao usuário final, mas existem na forma de processos elementares especificados para a equipe de desenvolvimento.

5.3.3. Requisitos para contagem de componentes

Há requisitos obrigatórios para um componente ser contado:

- ser um processo elementar;
- ser criado com a visão de reuso por outras aplicações (se for um serviço de negócio ou middleware);

5.3.3.1. Ser um processo elementar

Como toda funcionalidade, para poder ser contado, um componente (ex.: webservice) precisa ser um processo elementar. Com diz o CPM, no item 5.5.2 da Parte 1, um processo elementar tem as seguintes características:

1) ser reconhecido pelo usuário;

2) constituir uma transação completa;

3) ser autocontido;

4) deixar o negócio da aplicação em estado consistente.

Percebe-se no *paper Pontos de Função & Contagem de Software Aplicativo Middleware* (IFPUG, 2009a) que, para cada funcionalidade avaliada, faz-se a pergunta se "é um processo elementar". No outro *paper* do IFPUG *Utilizando Pontos de Função para medir software reutilizável* percebe-se que as funcionalidades também são auto-contidas.

5.3.3.2. Ser criado com a visão de reuso por outras aplicações (se for um serviço de negócio ou middleware)

Não pode ser criado apenas para uso interno a uma aplicação, caso típico de uma subrotina. No mesmo *paper* Utilizando Pontos de Função para medir software reutilizável, consta que um propósito de análise dessas funções (página 5), seria identificar "quais funcionalidades são fornecidas para os desenvolvedores de aplicações para apoiar atividades negociais comuns e recorrentes". Desse *paper* infere-se que os serviços são criados com a visão de reuso e não por uma necessidade técnica (conforme resumo na página 10). Essa inferência é confirmada com a leitura do *paper Sizing Component-Based Development using Function Points* (IFPUG, 2009b) que diz (página 3):

> *"In general, components are simply a reuse strategy. A reuse "code it once, use it many times" strategy is nothing new, but the emergence and acceptance of Object-Oriented methodologies has many more organizations recognizing the potential value of Component-Based Development. For this paper, we will focus on components that are integrated into a larger business application".*

Esse pensamento bate com a orientação dada por especialistas renomados na métrica Pontos de Função. Destaca-se o texto do Carlos Eduardo Vazquez sobre o assunto (um dos autores do livro *Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software*):

> "Mesmo em uma arquitetura SOA, numa perspectiva de negócio, existem fronteiras que delimitam um conjunto coeso de funções entregues ao usuário. A medição deve preliminarmente estabelecer essa premissa e a partir dai, partir para a identificação das funções. (...) Se esse WebService for apenas para uso interno a aplicação, em termos da APF, será como uma sub-rotina, não sendo contado como uma EE. Para que seja contado deve ser como descrito no cenário VIII em dados compartilhados". (Questão: Assunto: Dúvida Contagem SOA Data: 12/02/2010, em http://www.fattocs.com/files/pt/livro-apf/discussoes/livro-apf-2011-02.pdf).

5.3.4. Ajustes na técnica de contagem

Para fins de contagem de componentes, valem as seguintes considerações:

. Não devem ser contados arquivos lógicos (para transações e nem para fronteiras de aplicações) se os arquivos não forem acessados diretamente pelas transações;

. O número de componentes referenciados (CR), se houver, deve ser somado ao número de ALR na derivação da complexidade da transação. Contudo, esses componentes não devem ser considerados como arquivos lógicos da aplicação.

5.3.5. Medição de front-end

A camada de interface com o usuário (*front-end*) pode ter seu desenvolvimento solicitado separadamente das camadas de negócio para se fazer uso do arcabouço de componentes existentes. Nesse caso, valem as seguintes considerações

. Não devem ser contados arquivos lógicos (ALRs para transações e Arquivos Lógicos para fronteiras de aplicações) se eles não forem acessados diretamente pelas transações;

. O número de componentes referenciados (CR), se houver, deve ser somado ao número de ALR na derivação da complexidade da transação. Contudo, esses componentes não devem ser considerados como arquivos lógicos da aplicação.

5.4. Medição de Data Warehouse

5.4.1. Estimativa do tamanho funcional

De posse do documento de visão do projeto, devem ser contadas as tabelas fato e as tabelas dimensão. Se não for possível identificar a complexidade das mesmas, devido a ausência dos atributos das tabelas, considera-se a complexidade baixa. Deve-se contar duas entradas externas associadas às cargas das tabelas fato e das tabelas dimensão, a complexidade de tais funcionalidades deve ser avaliada como média, considerando a ausência de definição detalhada das necessidades de informações. Para cada estrela, deve-se considerar uma saída externa complexa, considerando a geração do contexto de análise. Se os relatórios estiverem definidos nessa fase, estes devem ser contados como saídas externas médias. Senão, não serão contados. Para o cômputo de elementos funcionais, se houver necessidade de estimar funções, deve-se aplicar o fator de 75% do tamanho estimado em pontos de função.

5.4.2. ETL

5.4.2.1. Entradas externas

Em casos de melhoria e desenvolvimento de *data warehouse*, geralmente existem funcionalidades de cargas de dados nas tabelas do DW. Estas tabelas são denominadas tabelas fato e tabelas dimensão, pertencentes a um modelo multidimensional em um diagrama estrela. As funcionalidades de carga de dados são classificadas como entradas externas.

Uma situação a considerar é a da substituição da implementação de uma carga de dados pela cópia direta de dados do sistema de origem dentro da fronteira do DW, em ambiente de produção. Nesse caso, a cópia dos dados em produção é uma solução técnica e a funcionalidade de carga continua existindo, devendo ser contada como entrada externa.

Geralmente, os dados do DW provenientes de outras aplicações, denominadas de aplicações de origem dos dados, são armazenados em uma base de dados temporária, denominada *Data Staging Area* (DSA). Assim, os dados são importados da aplicação de origem para a DSA e então, em outro processo de integração, importa os dados da DSA para as tabelas fato e dimensão do DW. Observe que a utilização da DSA é uma solução técnica, portanto não tem contagem funcional. No entanto, é importante ressaltar que em alguns casos, o usuário

deseja realizar consultas e emitir relatórios diretamente sobre os dados da DSA. Nesses casos, as funcionalidades da DSA serão consideradas na contagem. Os dados da DSA serão contados como arquivos lógicos internos. As cargas de dados serão contadas como entradas externas.

5.4.2.2. Funções de dados relacionadas a entradas externas

Em um modelo de dados multidimensional, esquema estrela, são reconhecidos dois tipos de entidades: tabelas fato e tabelas dimensão.

As tabelas dimensão mantidas por um ou mais processos de ETL devem ser contadas como um arquivo lógico interno. Assim, para determinar a quantidade de entradas externas, deve-se definir quantos registros lógicos distintos podem ser identificados dentro da dimensão. Deve ser observada a quantidade de níveis na dimensão e se estes níveis são tratados de forma diferente (por exemplo, diferença no tratamento dos atributos). Caso não existam níveis hierárquicos ou subgrupos de dados dentro da dimensão, deve ser considerado apenas um registro lógico.

Conhecendo-se os registros lógicos da dimensão, deve ser contada uma entrada externa para incluir novas informações no registro lógico. Frequentemente, a atualização nos registros da dimensão ocorre por adição de dados. Assim, não são contadas entradas externas para alteração de dados. Deve-se ressaltar que a carga inicial de dados nas tabelas dimensão também deve ser contada separadamente como uma entrada externa, sendo uma função de conversão de dados. Se existir uma funcionalidade para exclusão de dados, esta será contada como entrada externa. Em geral, conta-se uma entrada externa para cada registro lógico da tabela dimensão. Algumas vezes, as tabelas dimensão não são mantidas por carga, possuindo dados estáticos. Nessas ocasiões, a dimensão não deve ser contada como arquivo lógico interno, nem como registro lógico. Essas tabelas são classificadas como dados de código (*code data*).

As tabelas fato são contadas como um arquivo lógico interno. Deve ser contada uma entrada externa para a carga de dados na tabela fato. Deve-se ressaltar que a carga inicial de dados nas tabelas fato também é contada separadamente como uma entrada externa, sendo uma função de conversão de dados.

O DW pode ter como fonte de dados vários sistemas. Assim, os dados de uma tabela fato ou de uma tabela dimensão podem ser carregados de vários sistemas de origem. Geralmente, o processamento dos dados de cada arquivo proveniente desses sistemas é diferente dos demais. Portanto, conta-se um arquivo lógico interno para a tabela fato ou tabela dimensão e uma entrada externa para cada carga de dados de um sistema de origem distinto.

Se houver leitura de dados de outras aplicações para validação de informações durante as cargas de dados, estas tabelas que são arquivos lógicos internos de outras aplicações e são apenas lidas pelo DW serão contadas como arquivos de interface externa.

Algumas vezes, o usuário requer a combinação de tabelas fatos gerando outra tabela fato ou uma estrutura de agregação, visando apoiar a geração de consultas. Em certas situações, a estrutura de agregação

pode ser formada por uma tabela fato e tabelas dimensão. A estrutura de agregação é contada como arquivo lógico interno e a carga de dados é contada como uma entrada externa.

Em algumas situações, o usuário com receio de perder dados das aplicações de origem, requisita que os dados dos sistemas de origem sejam copiados para uma área de armazenamento de dados operacional (*Operational Data Store* – ODS) do DW. Nessas ocasiões, os dados são copiados do sistema transacional de origem para a ODS. Assim, quando os dados da ODS são apenas uma cópia dos dados do sistema de origem, os dados do sistema de origem serão contados como arquivo de interface externa. Posteriormente, os dados são integrados dentro de um novo arquivo lógico interno (tabela fato ou tabela dimensão). Cada funcionalidade de carga de dados para o arquivo lógico interno é contada como uma entrada externa.

5.4.2.3. Consultas e saídas externas

Frequentemente, em casos de melhoria e desenvolvimento de DW, existem funcionalidades que geram arquivos de dados consolidados nas aplicações de origem (aplicações que fornecem os dados para o DW). Estas funcionalidades de exportação de dados da aplicação de origem podem ser contadas como saídas externas ou consultas externas na fronteira da aplicação de origem como manutenção evolutiva. Observe que estas funcionalidades não fazem parte da fronteira da aplicação de DW. No entanto, fazem parte do escopo da contagem do caso de melhoria ou desenvolvimento de DW.

Em alguns momentos, o DW acessa diretamente o banco de dados das aplicações de origem, por meio de ferramentas. Observe que nesses momentos não há transferência de dados para o banco de dados do DW. Assim, os dados do sistema de origem são contados como arquivos de interface externa e as consultas são contadas como consultas externas ou saídas externas.

5.4.3. OLAP

Em aplicações de *data warehouse,* existem requisitos para geração de relatórios usando ferramentas. Os relatórios requisitados pelo usuário e implementados pela equipe de desenvolvimento são contados como saídas externas. Os relatórios gerados pelo usuário por meio da ferramenta OLAP não são contados, porque não constituem um requisito do usuário para a equipe de desenvolvimento.

5.4.3.1. Tabelas de visualização – geração de cubos ou contexto de análise ou universo

Esse tipo de tabela normalmente é utilizado para consumo por outras aplicações ou pelo próprio *data mart*. A geração do contexto de análise deve ser contada como uma saída externa por tabela fato, considerando a estrela, ou seja, a tabela fato e as dimensões. Os arquivos referenciados serão as tabelas fato e cada tabela dimensão, identificada como arquivo lógico interno, e os itens de dados serão os atributos de todos os arquivos referenciados (tabela fato e dimensão) e as fórmulas associadas. Em algumas situações específicas, quando a geração do contexto de análise não possuir lógicas de processamento de cálculos ou criação de dados derivados, esta funcionalidade deve ser contada como consulta externa.

Nos casos de melhoria que possuem como requisitos alteração de fórmulas existentes ou criação de novas fórmulas em uma tabela fato, deve ser contada a funcionalidade de geração de contexto de análise como EFP alterado de acordo com o padrão da Nesma reproduzido neste manual.

5.4.3.2. Caso de melhoria – criação de fórmulas

As fórmulas são atributos lógicos associados às tabelas fato ou tabelas dimensão. Estas são criadas com a geração do contexto de análise da tabela fato. Assim, caso o usuário solicite a criação de uma nova fórmula, a contagem de EFP será a seguinte:

SE: geração do contexto de análise da tabela fato

Arquivos referenciados – tabela fato e suas tabelas dimensões.

Itens de dados – todos os campos alterados, incluídos ou excluídos da tabela fato, dimensão e fórmulas.

É importante ressaltar que caso seja solicitada alteração em campos ou criação de campos em tabelas fato, a contagem será a seguinte:

ALI: tabela fato

EE: atualização de dados da tabela fato (conversão)

EE: carga de dados na tabela fato

SE ou CE: geração de contexto de análise

É importante ressaltar que se for solicitada alteração em campos ou criação de campos em tabelas dimensão, a contagem será a seguinte:

ALI: tabela dimensão

EE: atualização de dados da tabela dimensão (conversão)

EE: carga de dados na tabela dimensão

SE ou CE: geração de contexto de análise

5.4.4. Funcionalidades de controle do data warehouse

Como um dos propósitos do *data warehouse* é o de disponibilizar dados históricos, as funções de limpeza de dados são usualmente incorporadas na área de controle do DW, como guardar 60 meses de dados históricos. Esta função de limpeza é contada como uma entrada externa.

São exemplos de dados utilizados para gerenciar o DW: datas nas quais uma funcionalidade inclui dados em uma tabela fato a partir dos dados de um sistema de origem, a quantidade de registros adicionados, a quantidade de registros rejeitados, ou parâmetros utilizados para o processamento. Os processos elementares da aplicação devem ler e editar esses metadados. Estas funções não são identificadas pelo usuário final. No entanto, estes mecanismos de controle devem ser criados para o DW, sendo consideradas pelo perfil administrador. Assim, estas funcionalidades devem ser contadas.

5.4.5. Medição de relatórios

Os relatórios serão identificados como processos elementares sempre do ponto de vista negocial independentemente do leiaute adotado, assim como deve ser feito para qualquer tipo de sistema (DW ou OLTP). A disposição de gráficos ou tabelas em uma única aba ou em abas diferentes, em um mesmo documento ou em documentos diferentes não deve ser critério de identificação dos processos elementares, mas sim a diferenciação entre itens de dados, arquivos referenciados, possíveis agrupamentos e filtros comuns entre relatórios. Por exemplo, supondo a existência de dois gráficos, um com total de processos no estado aberto por

tipo de processo e outro com total de processos no estado encerrado por tipo de processo. Embora sejam dois gráficos, a estrutura é a mesma: estado (encerrado ou aberto), quantidade de processo e tipo de processo. Logo, seria computada apenas uma funcionalidade. Se uma mesma informação for apresentada tanto em um gráfico como em uma tabela, só se considerará uma funcionalidade.

5.5. Medição de Projetos Ágeis

Deve-se buscar seguir o modelo proposto no *Roteiro de Métricas de Software do Sisp versão 2.2 (SLTI, 2016)* no que diz respeito à não contabilização de refinamentos sucessivos em *sprints* distintos dentro de uma *release* (ver detalhes na tabela 12, página 56, do Roteiro). Consideram-se as seguintes definições no Roteiro:

Release: É um ciclo que perpassa pelas fases do processo de desenvolvimento de software com o objetivo de entregar, ao final do ciclo, um produto pronto (entregue e aceito pelo usuário) a ser colocado em produção para uso. A duração de cada release será definida pela contratante na fase de planejamento do projeto conforme seu *backlog* priorizado de forma a garantir uma entrega de valor antecipada aos usuários.

Sprint: É uma unidade de período de tempo fixo (time box) dentro da *release*, com datas de início e fim pré-definidas, dentro da qual é executado um conjunto de atividades de desenvolvimento do projeto previamente estabelecidas, gerando ao final um incremento do produto aceito e potencialmente implantável.

Refinamentos: são quaisquer mudanças ocorridas sobre uma função transacional ou de dados já previamente trabalhada(s) na *release* corrente (seja por meio de uma inclusão, alteração ou exclusão), provocadas pelo aprofundamento, detalhamento e complementação de requisitos durante o processo de desenvolvimento.

Para facilitar a apuração e para viabilizar o uso das vantagens citadas na seção *2.5.3.Reflexão sobre a escolha da métrica funcional*, alternativamente à aplicação da seção *2.5.2. Mensuração do tamanho funcional em pontos de função*, pode-se aplicar a correlação que se segue para derivação de pontos de função a partir de elementos funcionais em contratos externos em que a precificação não se dá por ponto de função, mas se exige o dimensionamento do tamanho em pontos de função para fins de Acordo de Nível de Serviço:

PF = EF * 1,36

Onde:

PF – tamanho derivado de pontos de função

EF – tamanho medido em elementos funcionais conforme seção *2.5.1. Mensuração do tamanho funcional em elementos funcionais e suas submétricas*

O coeficiente 1,36 foi obtido a partir da correlação linear[1] entre o tamanho em PF e o tamanho em EF apurados no dimensionamento da produção funcional de *sprints* de desenvolvimento ágil interno ao TCU do sistema e-Contas no período de 4/11/2012 a 13/12/2012. A tabela que se segue apresenta os números apurados no período.

Data medição	EF	PF	Proporção PF/EF
04/11/2014	11,99	17,25	144%
11/11/2014	15,52	18,00	116%
18/11/2014	20,21	30,00	148%
25/11/2014	16,21	21,50	133%
02/12/2014	10,56	13,00	123%
09/12/2014	21,76	30,25	139%
16/12/2014	5,64	7,00	124%
Total	101,89	137,00	134%
Média	14,56	19,57	134%

Tabela 9 – Produção funcional de sprints do e-Contas

6. Métrica Na Gestão De Contratos Externos

Alguns detalhes são apresentados para subsidiar as medições aplicadas em projetos alvo de gestão de contratos externos.

6.1. Tipos De Medição Aplicados

Para efeito de remuneração da empresa, serão aplicáveis ao processo de gestão de contratos externos os tipos de medição caso de melhoria, caso de desenvolvimento e aplicação, sempre usando o método de contagem detalhada.

6.2. Funções Já Existentes Na Aplicação Não Serão Remuneradas

Funções pré-existentes no sistema contratado não serão remuneradas, exceto se sofrerem alteração no escopo da medição de um caso de melhoria.

6.3. Aceite De Medição É Obrigatório

A medição realizada deve ser auditada por especialistas do TCU.

6.4. Diferenças De Contagem

Casos que exigirem revisão na aplicação da métrica (por exemplo: dupla interpretação, omissão) e não estiverem previstos neste manual nem no guia “Melhores Práticas de Medição Funcional” do TCU, serão resolvidos por acordo entre as partes do contrato, tomadas como referência as melhores práticas de contagem

[1] Foi apurado um coeficiente de correlação de 99,36% (R^2) entre as duas métricas a um p-value de 8,15E-08, ou seja, trata-se de uma correlação confiável estatisticamente falando.

usadas em contratos com a Administração Pública. O padrão seguido para sanar a diferença deve ser registrado no guia “Melhores Práticas de Medição Funcional” do TCU e deve ser adotado nas contagens seguintes.

6.5. Reaproveitamento De Funcionalidades

Funcionalidades que fizerem parte do framework já implementado pelo TCU não devem ser remuneradas. São exemplos de funcionalidades já implementadas: controle de acesso de usuário, consulta a lista de unidades do TCU, funções associadas à gestão eletrônica de documentos, entre outras.

6.6. Tratar Revisões do Manual

Este Manual de Medições é dinâmico e está em constante evolução. É importante que seja tratado no contrato o impacto de possíveis revisões. Se nada for dito em contrário, poderão ser aplicadas novas versões do Manual, desde que haja concordância das partes envolvidas.

6.7. Documentação Exigida

É responsabilidade da contratada manter um *baseline* de funcionalidades das aplicações (pode conter apenas as funcionalidades alvo de solicitações de contagem) e o histórico das referências a essas funcionalidades em contagens, detalhando também a solicitação (OS) associada a cada evolução funcional. Esse histórico deve ser acessível à equipe do TCU.

Esse *baseline* impedirá contagens em duplicidade de uma mesma operação (inclusão, alteração ou exclusão de funcionalidade). Cabe ao gestor técnico de uma solicitação ou ordem de serviço (ou papel semelhante) a fiscalização de contagens repetidas. Casos de mensuração em duplicidade podem justificar adequada punição à contratada.

A documentação deve seguir o padrão definido neste manual.

6.8. Revisão Dos Percentuais Por Atividade Implementada

O contrato pode revisar os percentuais das fases do ciclo de desenvolvimento de software constantes da metodologia de desenvolvimento e da tabela apresentada no passo “Aplicação dos percentuais por atividade implementada” do processo de medição.

6.9. Manutenção Corretiva

O contrato deve detalhar como se dá a garantia. Uma manutenção corretiva faz parte da garantia do contrato. Caso não exista cláusula contratual de garantia, deve ser considerada a garantia de seis meses, preconizada por lei (Código do Consumidor).

6.10. Prazo Máximo De Desenvolvimento

É importante que o contrato estabeleça critério para definição de um prazo máximo de entrega dos projetos. Se o contrato não especificar percentuais, deve-se seguir os indicados abaixo.

Tamanho do Projeto (em PF)	Tamanho do Projeto (em EF)	Prazo máximo (em dias úteis)
Até 10	Até 7,50 EF	10 dias
De 11 a 20	De 7,51 a 15,00	20 dias
De 21 a 30	De 15,01 a 22,50	30 dias
De 31 a 40	De 22,51 a 30,00	40 dias
De 41 a 50	De 30,01 a 37,50	50 dias
De 51 a 60	De 37,51 a 45,00	60 dias
De 61 a 70	De 45,01 a 52,50	70 dias
De 71 a 85	De 52,51 a 60,00	88 dias
De 86 a 99	De 60,01 a 67,50	104 dias

Tabela 10 - Prazo máximo de desenvolvimento por faixa de tamanho

O método utilizado para estimar o prazo máximo dos projetos com tamanho superior ao constante da tabela acima é dado pela seguinte fórmula

$$T = V^{0,35}$$

Onde:

T: prazo máximo de desenvolvimento em meses

V: tamanho funcional do projeto

7. Glossário

Nessa seção são apresentadas algumas definições usadas nesse documento.

A	
AIE	Arquivo de Interface Externa.
ALI	Arquivo Lógico Interno.
ALR	Arquivo Lógico Referenciado.
APF	Análise de Pontos de Função.
Aplicação	Representa o sistema na visão do usuário. Pode estar segmentada em uma ou mais unidades de software.
AR	Arquivos Referenciados (o mesmo que ALR).
Arquivo	No contexto da APF este termo não significa arquivo no sentido tradicional de processamento de dados. Neste caso, Arquivo refere-se a um grupo lógico de dados ou informações de controle, e não à implementação física destes.
Arquivo de Interface Externa	Grupo de dados ou informações de controle, logicamente relacionados, referenciados pela aplicação mas mantidos dentro da fronteira de outra aplicação. Sua principal intenção é armazenar dados referenciados através de um ou mais processos elementares da aplicação sendo contada. Um AIE contado para uma aplicação deve ser um ALI para outra aplicação.
Arquivo Lógico Interno	Grupo de dados ou informações de controle, logicamente relacionados, mantidos dentro da fronteira da aplicação. Sua principal intenção é armazenar dados mantidos através de um ou mais processos elementares da aplicação sendo contada.
Arquivo Referenciado	É um arquivo lógico interno lido ou mantido pela função transacional, ou um arquivo de interface externa lido pela função transacional.
C	
Características Gerais do Sistema	Refletem as funcionalidades gerais fornecidas pela aplicação ao usuário, não aplicadas neste manual.
CE	Consulta externa.
Consulta Externa	Processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação. Sua principal intenção é apresentar informação ao usuário através da recuperação de dados ou informações de controle de um ALI ou AIE. A lógica de processamento não deve conter fórmula matemática ou cálculo, criar dados derivados, manter um ou mais ALI e/ou alterar o comportamento do sistema.
CPM	*Counting Practices Manual* ou Manual de Práticas de Contagem, versão 4.3.1 (IFPUG, 2010)
CMMI	*CMMI – Capability Maturity Model* para software é um conjunto de processos desenvolvido pela *SEI – Software Engineering Institute* (www.sei.cmu.edu) em 1986 para melhorar o desenvolvimento de Aplicações em organizações que trabalham com tecnologias de software. O processo é divido em 5 níveis de desenvolvimento: Inicial, repetível, definido, gerenciado com métricas e otimizado.
D	
Dados de código	Também chamados metadados, em geral não são especificados pelo próprio usuário, sendo identificados pelo desenvolvedor em resposta a um ou mais

	requisitos técnicos. A codificação de atributos descritivos em objetos de negócio, sua descrição, nome ou outros dados que também o descrevam, como a data de início ou término de sua vigência, são os atributos típicos desses arquivos.
Dados de negócio	São os dados necessários ao negócio do usuário. Por exemplo, em um sistema de recursos humanos, são dados de negócio as informações sobre os funcionários, como nome e endereço, entre outros.
Dados de referência	São definidos como requisitos de armazenamento que suportam regras de negócio na manutenção de dados de negócio. Os dados de código podem ter o código substituído pela respectiva descrição nos objetos de negócio em que são utilizados sem que o significado destes últimos sejam alterados, enquanto o mesmo não pode ser feito com os dados de referência. Por exemplo, para um sistema de cálculo de folha de pagamento, os percentuais e valores das faixas de imposto são considerados dados de referência.
Dado derivado	Informação criada a partir da transformação de dados existentes. Requer outro processamento além da recuperação, conversão e edição direta de dados.
E	
EE	Entrada Externa
EF	Elementos Funcionais
EFP	*Enhancement Function Points* - é o número de pontos de função do caso de melhoria.
EFPALT	É o número de pontos de função das funções modificadas na aplicação pelo caso de melhoria. Reflete as funções depois das modificações.
EFPCONV	É o número de pontos de função das funções de conversão em casos de desenvolvimento e de melhoria.
EFPEXC	É o número de pontos de função das funções excluídas da aplicação pelo caso de melhoria.
EFPINC	É o número de pontos de função das funções incluídas na aplicação pelo caso de melhoria.
Elementos Funcionais	Representa o número relativo de elementos que compõem uma função. Para cada tipo de função existem dois tipos de elementos funcionais que são utilizados para a derivação de complexidade nas tabelas de cálculo. No caso de funções de dado, temos: tipos de dados e registros lógicos. Para funções transacionais, temos: tipos de dados e arquivos referenciados.
Entrada Externa	Processo elementar que processa dados ou informações de controle vindas de fora da fronteira da aplicação. Os dados processados mantêm um ou mais ALI enquanto as informações de controle podem ou não manter um ALI. A principal intenção de uma EE é manter um ou mais ALI e/ou alterar o comportamento do sistema.
Escopo da Medição	Define as funcionalidades que serão incluídas em determinada medição de pontos de função.
F	
FP	*Function Point*, ou traduzindo, Pontos de Função.
FSM	*Functional Size Measurement* ou, traduzindo, Medição de Tamanho Funcional.
FDES	É o tamanho funcional do caso de desenvolvimento.

FP_ALTERADO	Equivale ao EFP (EFPinc, em caso de inclusão, EFPalt em caso de alteração, etc), seguindo os conceitos deste manual
FP_NAO_AJUSTADO	Equivale ao tamanho da funcionalidade segundo o IFPUG, sem aplicação de fatores de impacto.
Fronteira da aplicação	É a interface conceitual que delimita o software sendo dimensionado e o mundo exterior.
Funções tipo dados	Representam as funcionalidades fornecidas pelo sistema ao usuário, para atender suas necessidades de dados.
Funções tipo transação	Representam as funcionalidades de processamento de dados fornecidas pelo sistema ao usuário.
I	
IFPUG	International Function Point Users Group
Informações de controle	São dados que influenciam um processo elementar da aplicação sendo contada. Eles especificam o que, quando ou como os dados devem ser processados. No caso das funções de dado, esses parâmetros são armazenados e mantidos em conjunto com a aplicação. São exemplos comandos de ação, parâmetros de consulta, enfim, informação que especifica o que, quando, ou como os dados devem ser processados.
ISO	International Organization for Standardization.
ISO / IEC	Refere-se a um padrão estabelecido em conjunto pelas organizações internacionais ISO e IEC.
IEC	International Engineering Consortium.
L	
Lógica de Processamento	É definida como qualquer dos seguintes requisitos especificamente solicitados pelo usuário para completar um processo elementar: a. Realização de validações b. Realização de cálculos e fórmulas matemáticas c. Conversão de equivalência entre montantes d. Filtragem e seleção de dados utilizando determinados critérios para comparar múltiplos conjuntos de dados e. Análise de condições para determinação de qual se aplica f. Atualização de um ou mais ALI g. Referência a um ou mais ALI ou AIE h. Recuperação de dados ou informações de controle i. Criação de dados derivados pela transformação dos dados existentes em novos dados j. Alteração do comportamento da aplicação k. Preparação e apresentação de informação para fora da fronteira da aplicação l. Capacidade de aceitar dados ou informação de controle que entra na fronteira da aplicação m. Ordenação ou organização de dados.
M	

Manual de Práticas de Contagem	Documento editado pelo IFPUG que descreve toda a técnica da APF.
N	
Nesma	*Netherlands Software Metrics Association*
P	
Pontos de função não ajustados	São os pontos de função encontrados para uma função ao se aplicar as regras dessa métrica, excetuando a utilização dos fatores de impacto.
Processo Elementar	É a menor unidade de atividade significativa para o usuário final. Esse processo elementar deve ainda ser completo em si mesmo e deixar a aplicação em estado consistente.
T	
TD	Tipo de Dado
Tipo de Dado	Campo único, reconhecido pelo usuário, não repetido.
Tipo de Registro	É um subgrupo de tipos de dados, reconhecido pelo usuário, componente de um arquivo lógico interno ou arquivo de interface externa. Existem dois tipos de subgrupo: os opcionais, os quais o usuário tem a opção de não informar no processo elementar que cria ou adiciona dados ao arquivo, e os obrigatórios, os quais o usuário requer que sejam sempre utilizados pelo processo elementar que cria ou adiciona dados ao arquivo.
TR	Tipo de Registro
U	
Usuário	É qualquer pessoa que especifica requisitos funcionais do usuário e /ou qualquer pessoa ou coisa que, a qualquer momento, se comunique ou interaja com o sistema.
V	
Visão do usuário	Representa uma descrição formal das necessidades do negócio do usuário em sua própria linguagem, sendo compreendida por usuários e desenvolvedores.

8. Referências Bibliográficas

CASTRO, M.V.B.; HERNANDES, C.A.M., **A Metric of Software Size as a Tool for IT Governance**, *Software Engineering (SBES), 2013 27th Brazilian Symposium on*, vol., no., pp.99,108, Disponível em: <http://revista.tcu.gov.br/ojsp/index.php/RTCU/article/view/1325>. 1-4 Oct. 2013.

IFPUG - International Function Point Users Group. **Framework for Functional Sizing.** Disponível em: www.ifpug.org. 2003.

_____. **Pontos de Função & Contagem de Software Aplicativo Middleware.** Disponível em: www.ifpug.org. 2009a.

_____. **Sizing Component-Based Development using Function Points.** Disponível em: www.ifpug.org. 2009b.

_____. **Manual de Práticas de Contagem de Pontos de Função**, Versão 4.3.1 IFPUG. Disponível em: www.ifpug.org. 2010.

_____. **Shared Data Real-time Requests** - iTip 5 - Version 1.1, Disponível em: : www.ifpug.org. 21 set. 2015.

NESMA. Netherlands Software Metrics Association. **Function Point Analysis For Software Enhancement**. Disponível em: <http://www.nesma.nl/download/boeken_NESMA/N13_FPA_for_Software_Enhancement_(v2.2.1).pdf>. Acesso em: 16 out. 2012. 2009.

SLTI. Secretaria de Logística e Tecnologia da Informação. Ministério do Planejamento, Orçamento e Gestão. **Roteiro de Métricas de Software do SISP**, versão 2.2. Brasília: SISP. 2016.

Anexo I - Relatório de medição estimativa de tamanho funcional (exemplo)

1. ID da medição :_____

2. Nome do caso :_____

3. Nome do autor da medição :_____ _____**Data:**__/__/____

4. Propósito da medição :_____

5. Tipo da medição : ( ) Aplicação ( ) Desenvolvimento ( ) Melhoria

6. Escopo da medição :_____

7. Fronteira(s) :

N°	ID	NOME

8. Pressuposições adotadas :_____

10.Documentação utilizada

N°	NOME	URL	ID

11.Funções de dados :

N°	NOME	ID FRONT.	ID. DOC.	ID REQ.	TIPO	COMPL.	PF/EF	INC/ALT/ EXC	FI
SUBTOTAL									

12.Funções de transação :

N°	NOME	ID FRONT.	ID. DOC.	ID REQ.	TIPO	COMPL.	PF/EF	INC/ALT/ EXC	FI
SUBTOTAL									

13.ALRs :

N°	NOME	ID FRONT.

14.Funções de transação x ALR:

ALR / FUNÇÃO	1	2	3	4												
1	LE	E	L	NA												
2	NA	NA	NA	NA												

15.Resultado da estimativa :

1. TAMANHO FUNCIONAL: ____FP ou ____ EF
2. ESFORÇO: ____PESSOA-MÊS
3. TEMPO : ____MESES

_____<assinatura>_____

Anexo II - Relatório de contagem funcional detalhada (exemplo)

1. ID da contagem :_____

2. Nome do caso :_____

3. Nome do autor da contagem :_____**Data:**__/__/____

4. Propósito da contagem :_____

5. Tipo da contagem : ( ) Aplicação ( ) Desenvolvimento ( ) Melhoria

6. Escopo da contagem :_____

7. Fronteira(s) :

N°	ID	NOME

8. Pressuposições adotadas :_____

10.Documentação utilizada :

N°	NOME	URL	ID

11.Funções de dados :

N°	FUNÇÃO	ID FRONT	ID DOC	ID REQ	TIPO	COMPLX	FP/ EF	INC/ALT/ EXC	FI
1	XXXXXXXXXXXXXXXXXX	F0001	D001	R001	ALI	BAIXA	7	ALT	0,5

ELEMENTO	TIPO	ORIG	INC	ALT	EXC
XXXXXXXXXXXX	TR	X			
WWWWWWWWWWW	TD		X		

12.Funções de transação :

N°	FUNÇÃO	ID FRONT	ID DOC	ID REQ	TIPO	COMPLX	FP/ EF	INC/ALT /EXC	FI
1	XXXXXXXXXXXXXXXXXX	F0002	D002	R002	EE	BAIXA	3	ALT	0,25

ELEMENTO	TIPO	ORIG	INC	ALT	EXC
XXXXXXXXXXXX	ALR	X			
WWWWWWWWWWW	TD		X		

13.Itens não mensuráveis :

N°	NOME	ID FRONT.	ID. DOC.	ID REQ.	TIPO	COMPLX	FP/ EF	INC/ALT/EXC	FI

14. TAMANHO FUNCIONAL FINAL : ____FP ou ____EF

_____<assinatura>_____