

GUIA DE CONTAGEM DE PONTOS DE FUNÇÃO

DO MINISTÉRIO DO PLANEJAMENTO

VERSÃO: 1.0



Histórico de Revisão

Data	Versão	Descrição	Autor
04/2015	1.0	Elaboração do manual.	Márcia Regina Guiotti Bomfim José Romildo Andrade

Sumário

SUMÁRIO.....	3
1. OBJETIVO.....	5
2. REFERÊNCIAS PRINCIPAIS.....	5
2.1. COUNTING PRACTICES MANUAL DO IFPUG (CPM).....	5
2.2. ROTEIRO DE MÉTRICAS DE SOFTWARE DO SISP (ROTEIRO SISP).....	5
3. INTRODUÇÃO.....	5
4. ENTENDIMENTO DA TÉCNICA.....	6
4.2. PROPÓSITO DA CONTAGEM.....	6
4.3. TIPO DE CONTAGEM.....	7
4.4. ESCOPO DA APLICAÇÃO.....	7
4.5. VISÃO DO USUÁRIO.....	7
4.6. FRONTEIRA DA APLICAÇÃO.....	8
4.7. FUNÇÕES DE DADOS.....	9
4.7.1. ARQUIVO LÓGICO INTERNO (ALI).....	10
4.7.2. ARQUIVO DE INTERFACE EXTERNA (AIE).....	10
4.7.3. REGISTRO LÓGICO REFERENCIADO (RLR OU TR).....	10
4.8. FUNÇÕES TRANSACIONAIS.....	10
4.8.1. ENTRADA EXTERNA (EE).....	10
4.8.2. CONSULTA EXTERNA (CE).....	10
4.8.3. SAÍDA EXTERNA (SE).....	10
4.9. NÍVEL DE DETALHAMENTO DE CONTAGENS.....	11
4.9.1. CONTAGEM INDICATIVA.....	11
4.9.2. CONTAGEM ESTIMADA.....	12
4.9.3. CONTAGEM DETALHADA.....	12
4.10. FATOR DE AJUSTE.....	13
4.11. ORIENTAÇÕES PARA AS CONTAGENS.....	13
4.11.1. DADO DE CÓDIGO.....	13
4.11.2. LOG, TRILHA DE AUDITORIA, REGISTRO DE EVENTOS E HISTÓRICO.....	14
4.11.2.1. HISTÓRICO.....	14
4.11.2.2. REGISTRO DE EVENTOS.....	14
4.11.2.3. LOG.....	15
4.11.2.4. TRILHA DE AUDITORIA.....	15
4.11.3. CONSULTAS COM FILTROS DIFERENTES E COM AS MESMAS SAÍDAS.....	15
4.11.4. CONSULTAS COM FILTROS IGUAIS E COM SAÍDAS DIFERENTES.....	16
4.11.5. INTEGRAÇÃO ENTRE APLICAÇÕES.....	16
4.11.5.1. CENÁRIO 1 (SISTEMA “A” REQUISITA DADOS DO SISTEMA “B”).....	16
4.11.5.2. CENÁRIO 2 (SISTEMA “B” FORNECE DADOS PARA O SISTEMA “A”).....	17
4.11.5.3. CENÁRIO 3 (DISPONIBILIZAÇÃO E CONSUMO DE DADOS DENTRO DA MESMA FRONTEIRA).....	18
4.11.5.4. IDENTIFICAÇÃO DO BARRAMENTO COMO UMA FRONTEIRA.....	18
4.11.5.5. CONFIGURAÇÃO/CUSTOMIZAÇÃO DE FERRAMENTA DE BARRAMENTO.....	19
4.11.6. MANUTENÇÃO DE COMPONENTE REUTILIZÁVEL.....	19
4.11.7. MÚLTIPLAS MÍDIAS.....	20
4.11.8. MÚLTIPLOS PROCESSOS COMPONDO UM PROCESSO ELEMENTAR (SUBDIVISÃO DE FUNCIONALIDADES).....	22
4.11.9. CONSULTAS IMPLÍCITAS.....	23

4.11.10.	CONTAGEM DE FUNÇÕES PARA CONVERSÃO DE DADOS.....	23
4.11.11.	PROJETOS DE MIGRAÇÃO DE DADOS.....	24
4.11.12.	PROJETOS DE MELHORIA (MANUTENÇÃO EVOLUTIVA).....	24
4.11.13.	MUDANÇA ESTRUTURAL EM ARQUIVOS LÓGICOS PARA PROJETOS DE MELHORIA.....	25
4.11.13.1.	IMPACTO DAS ALTERAÇÕES DAS CARACTERÍSTICAS DE ITENS DE DADOS DE UM ALI NAS FUNÇÕES TRANSAIONAIS QUE O MANTÉM.....	25
4.12.	ITENS NÃO MENSURÁVEIS.....	26
4.12.1.	MANUTENÇÃO CORRETIVA.....	26
4.12.1.1.	SISTEMA SEM DOCUMENTAÇÃO, DESATUALIZADA OU INCOMPLETA.....	27
4.12.1.2.	SISTEMA COM DOCUMENTAÇÃO COMPLETA E ATUALIZADA.....	27
4.12.2.	MANUTENÇÃO DE INTERFACE.....	28
4.12.3.	MANUTENÇÃO EM DADOS DE CÓDIGO.....	28
4.12.4.	ATUALIZAÇÃO DE DADOS DE CÓDIGO.....	29
4.12.5.	MENSAGENS.....	29
4.12.6.	PROGRAMAS AUXILIARES.....	30
4.13.	MUDANÇA DE REQUISITOS DURANTE O PROJETO DE DESENVOLVIMENTO OU MANUTENÇÃO.....	31
4.13.1.	MUDANÇAS ASSOCIADAS AO DOMÍNIO DO NEGÓCIO.....	32
4.13.2.	DESISTÊNCIA DE INCLUIR, ALTERAR OU EXCLUIR UMA FUNÇÃO.....	35
4.14.	CONTAGEM DE PONTOS DE FUNÇÃO NO DESENVOLVIMENTO DE SOFTWARE COM MÉTODOS ÁGEIS.....	35
5.	DICAS PARA OTIMIZAR O CUSTO DAS MANUTENÇÕES.....	42
6.	ENTREGA DA CONTAGEM.....	43
7.	GLOSSÁRIO.....	43
8.	PROCESSO DE REVISÃO DO GUIA DE CONTAGEM.....	44
8.1.	REVISÃO PARA CORREÇÃO DE INCONSISTÊNCIAS E SITUAÇÕES NÃO PREVISTAS.....	44
8.2.	REVISÃO PARA ADOÇÃO DE NOVAS VERSÕES DO CPM.....	44
9.	REFERÊNCIAS BIBLIOGRÁFICAS.....	44

1. Objetivo

Este documento tem como propósito apresentar o Guia de Contagem de Pontos de Função proposto pelo Ministério do Planejamento, Orçamento e Gestão (MP) para a aplicação no contexto de desenvolvimento e manutenção de softwares do MP. Este guia funciona como um complemento ao Manual de Práticas de Contagem do IFPUG (*International Function Point Users Group*-www.ifpu.com), procurando tornar mais prático o uso dos conceitos e regras definidos pelo IFPUG, exemplificando situações específicas do MP e apresentando situações em que o IFPUG não oferece orientação prática ou objetiva.

2. Referências Principais

2.1. Counting Practices Manual do IFPUG (CPM)

O Manual de Práticas de Contagem do IFPUG, que estabelece o padrão para a contagem de pontos de função, especifica um conjunto de definições, regras e passos para a aplicação do método de medição funcional do IFPUG. O MP adotará como referência a versão 4.3.1, podendo adotar no futuro novas versões conforme descrito no tópico Processo de Revisão para Adoção de Novas Versões do CPM.

2.2. Roteiro de Métricas de Software do SISP (Roteiro SISP)

O Roteiro de Métricas de Software do SISP (Roteiro SISP) tem o objetivo de apresentar métricas, com base nas regras de contagens de pontos de função do CPM, para vários tipos de projetos de desenvolvimento e manutenção de software, promovendo o uso de métricas objetivas em contratos de prestação de serviços de desenvolvimento e manutenção de sistemas.

O Roteiro SISP (versão mais atual) deve ser aplicado de forma complementar a este guia de contagem. Ocorrendo ambiguidade entre as informações descritas neste guia e no Roteiro SISP, devem prevalecer as regras estipuladas neste guia.

No caso de publicação de nova versão do Roteiro SISP, as contratadas terão o prazo estipulado pelo MP, porém nunca inferior a 30 dias corridos para adequarem-se à nova versão.

3. Introdução

A Análise de Pontos de Função é uma técnica de medição das funcionalidades de um software sob o ponto de vista do usuário, ou seja, determina o tamanho funcional do software. Segundo o CPM, a técnica mede o software quantificando as tarefas e serviços (isto é, funcionalidade) que o software fornece ao usuário, primordialmente com base no projeto lógico.

Os objetivos da análise de pontos de função são:

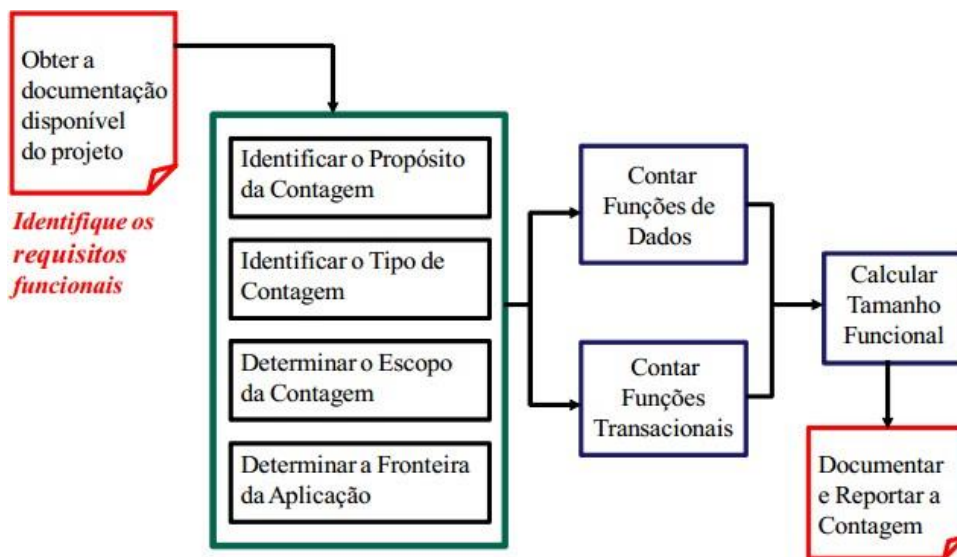
- Medir a funcionalidade implementada no software, que o usuário solicita e recebe e;

- Medir a funcionalidade impactada pelo desenvolvimento, melhoria e manutenção de software, independentemente da tecnologia utilizada na implementação.

O Ponto de Função (PF) é a unidade de medida que tem por objetivo tornar a medição independente da tecnologia utilizada para a construção do software. Essa medida está diretamente relacionada aos requisitos de negócio que o software se destina a abordar, ou seja, busca medir o que o software faz e não como ele foi construído. Portanto, pode ser facilmente aplicada em uma ampla gama de ambientes de desenvolvimento e ao longo do ciclo de um projeto de desenvolvimento, desde a definição de requisitos até o uso operacional completo. A técnica fornece uma medida objetiva e comparativa que auxilia na avaliação, planejamento, gestão e controle da produção de software.

4. Entendimento da Técnica

A figura abaixo ilustra o procedimento de contagem de pontos de função.



4.1. Obter documentação disponível do projeto

A contagem de pontos de função se inicia com a análise da documentação disponível do projeto em questão, visando à identificação dos requisitos funcionais.

4.2. Propósito da Contagem

Uma contagem deve prover uma resposta a um problema do negócio e é o problema do negócio que determina o propósito. O objetivo, nesta etapa, é tornar claro o que se pretende atingir com a contagem que será feita.

Exemplos de propósito são:

- Fornecer o tamanho funcional de um projeto como uma entrada para o processo de estimativa a fim de determinar o esforço para desenvolver a primeira versão de uma aplicação;
- Fornecer o tamanho funcional da base instalada das aplicações para determinar os custos de sustentação por ponto de função;
- Fornecer o tamanho funcional de dois pacotes para permitir a comparação de funcionalidade oferecida por cada um.

4.3. Tipo de Contagem

O tipo de contagem é estabelecido pelo responsável da contagem, podendo estar associada tanto a projetos quanto a aplicações. O tipo é determinado pelo propósito da contagem.

Segundo o CPM, há três tipos de contagem de pontos de função, a saber:

- **Projeto de Desenvolvimento:** Este tipo de contagem mede a funcionalidade entregue ao usuário na primeira instalação do software, quando o projeto estiver completo.
- **Projeto de Melhoria (*Enhancement*):** Este tipo de contagem mede as modificações em uma aplicação já existente que adicione, altere ou exclua funções entregues ao usuário quando o projeto estiver completo.
- **Aplicação:** Este tipo de contagem está relacionado à aplicação instalada. Representa a *baseline* da contagem de pontos de função de uma aplicação, ou seja, é uma medida das funções atuais providas ao usuário.

4.4. Escopo da aplicação

Segundo o CPM, o escopo da contagem define o conjunto de requisitos funcionais de usuários para ser incluídos na contagem. O escopo:

- Define o (sub)conjunto do software que está sendo medido;
- É determinado pelo propósito para a realização da contagem de pontos de função;
- Identifica quais funções serão incluídas na medida de tamanho funcional assim como fornecer respostas relevantes para o propósito da contagem;
- Pode incluir mais de uma aplicação.

4.5. Visão do Usuário

Segundo o CPM, um usuário é qualquer pessoa ou coisa que se comunica ou interage com o software a qualquer momento.

A visão do usuário é o Requisito Funcional do Usuário como percebido pelo usuário.

Requisitos Funcionais do Usuário são um subconjunto dos requisitos do usuário que descrevem o que o software deverá fazer em termos de tarefas e serviços.

A visão do usuário representa uma descrição formal das necessidades dos negócios do usuário, na linguagem do usuário. Os desenvolvedores traduzem a informação do usuário para informações em linguagem técnica a fim de prover uma solução.

No contexto da análise de pontos de função, o termo usuário tem um conceito mais amplo do que o usual. Com isso, durante uma contagem de pontos de função, convém buscar dentro de um conjunto de usuários possíveis aquele cuja visão represente melhor as funções que a aplicação fornece. Por exemplo, uma aplicação de autoatendimento de um banco tem como usuários o cliente do banco, o funcionário da agência, o gestor do departamento responsável. Basear a contagem dessa aplicação somente na visão do cliente final do banco e usuário do autoatendimento seria considerar uma visão limitada da aplicação. É fundamental levar em consideração também a visão do usuário que especifica os requisitos e regras de negócio, neste caso, o gestor do departamento.

4.6. Fronteira da aplicação

De acordo com o CPM, a fronteira é uma interface conceitual entre o software sob estudo e seus usuários.

A fronteira da aplicação:

- Define o que é externo à aplicação;
- Indica a fronteira entre o software que está sendo medido e o usuário;
- Atua como uma 'membrana' através da qual os dados processados pelas transações (EEs, SEs e CEs) passam para dentro e para fora da aplicação
- Envolve os dados lógicos mantidos pela aplicação (ALIs);
- Auxilia na identificação dos dados lógicos referenciados, mas não mantidos pela aplicação (AIEs);
- Depende da visão externa do negócio do usuário da aplicação, independente de questões técnicas e/ou implementação.

Assim, a fronteira deve ser definida com base na perspectiva de negócio, nas áreas funcionais separadas como pode ser visto pelo usuário, não em considerações técnicas (por exemplo, arquitetura do sistema).

A fronteira inicial já estabelecida para a aplicação ou aplicações que estejam sendo modificadas não é influenciada pelo escopo da contagem.

Uma das dicas que o CPM descreve para a identificação da fronteira é obter uma documentação do fluxo de dados no sistema e traçar uma linha imaginária ao redor para destacar quais partes são internas e externas à

aplicação. Um artefato que bem representa esse fluxo é o diagrama de contexto, pois representa todo o sistema como um único processo e é composto por fluxos de dados que mostram as interfaces entre o sistema e as entidades externas, permitindo, com isso, identificar os limites dos processos, as áreas envolvidas com o processo e os relacionamentos com outros processos e elementos externos à empresa (ex.: clientes, fornecedores).

Quando houver mais de uma aplicação incluída no escopo da contagem. Nesse caso, múltiplas fronteiras da aplicação deverão ser identificadas.

Para o MP, perfis de usuário que vão utilizar o sistema, distribuição das equipes de desenvolvimento e definições arquiteturais e/ou tecnológicas não são insumos para a definição da fronteira.

Quanto ao uso do documento de visão, o objetivo principal dele é fornecer uma visão completa, em alto nível, do sistema de software em desenvolvimento. Devido ao fato de este documento poder contemplar requisitos funcionais e não funcionais, ainda que nele constem descritas as necessidades requisitadas e entregues ao usuário, a APF se propõe a medir apenas os requisitos funcionais (ou de negócio).

Com base nisso, o MP entende e define que esse documento pode não ser um insumo suficiente para a definição da fronteira, devendo existir uma avaliação mais abrangente sob a perspectiva de negócio, conforme definido neste tópico.

Diante de algum impasse na definição da fronteira, deve prevalecer a visão de negócio do usuário de acordo com o entendimento do MP, uma vez que cabe ao cliente ser o detentor do conhecimento do negócio e, como a fronteira impacta diretamente no resultado da contagem e consequentemente em custos, podem ser gerados conflitos de interesses.

Como o conceito de fronteira é muito importante, o MP reconhece que deve ser estabelecido um inventário das aplicações juntamente com a definição de suas fronteiras para que nas diversas medições tenha-se como base a mesma visão das fronteiras. Enquanto esse trabalho não estiver concluído, as definições da fronteira das aplicações serão definidas pontualmente.

Em projetos de melhoria, a fronteira estabelecida no início do projeto deve estar de acordo com a fronteira já estabelecida para a aplicação que está sendo modificada.

4.7. Funções de Dados

As funções de dados representam a funcionalidade oferecida ao usuário para satisfazer requisitos de armazenamento de dados internos e externos. Uma função de dado pode ser um arquivo lógico interno ou um arquivo de interface externo.

O termo arquivo aqui não significa arquivo físico ou tabela. Nesse caso, arquivo se refere a um grupo de dados logicamente relacionados e não à implementação física destes grupos de dados.

4.7.1. Arquivo Lógico Interno (ALI)

Grupo de dados ou informações de controle logicamente relacionados, identificável pelo usuário, mantido dentro da fronteira da aplicação. A intenção primária de um ALI é armazenar dados mantidos através de um ou mais processos elementares da aplicação sendo contada.

4.7.2. Arquivo de Interface Externa (AIE)

Grupo de dados logicamente relacionados ou informação de controle, reconhecido pelo usuário, referenciado pela aplicação sendo medida, mas que é mantido dentro da fronteira de outra aplicação. A intenção primária de um AIE é armazenar dados referenciados por um ou mais processos elementares dentro da fronteira da aplicação medida. Isto significa que um AIE contado por uma aplicação deve ser um ALI em outra aplicação. Deverão ser considerados na complexidade de um AIE apenas os itens de dados e os registros lógicos referenciados pela aplicação que está sendo contada.

4.7.3. Registro Lógico Referenciado (RLR ou TR)

Um tipo de registro lógico referenciado é um subgrupo de dados reconhecido pelo usuário dentro de uma função de dados (ALI ou AIE). Para a contagem dos RLRs de uma função de dados, devem ser aplicadas as regras estabelecidas no CPM.

4.8. Funções Transacionais

Uma função de transação é um processo elementar que oferece funcionalidade ao usuário para processar dados. Uma função de transação é uma entrada externa, saída externa, ou consulta externa.

4.8.1. Entrada Externa (EE)

Processo elementar que processa dado (ou informações de controle) vindo de fora da fronteira da aplicação. A principal intenção de uma EE é manter um ou mais ALI e/ou alterar o comportamento do sistema.

4.8.2. Consulta Externa (CE)

Processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação. A principal intenção de uma CE é apresentar informação ao usuário por meio de uma simples 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.

4.8.3. Saída Externa (SE)

Processo elementar que gera dados ou informações de controle que saem pela fronteira da aplicação. A principal intenção de uma SE é apresentar

dados ao usuário através de outra lógica de processamento que não apenas a recuperação de dados ou informações de controle. A lógica de processamento deve conter fórmula matemática ou cálculo, criar dados derivados, manter um ou mais ALI e/ou alterar o comportamento do sistema.

4.9. Nível de Detalhamento de Contagens

A contagem de pontos de função pode ser realizada com diferentes níveis de detalhe: indicativa, estimada e detalhada. O nível de detalhamento a ser escolhido depende de alguns fatores como a finalidade da contagem, as informações disponíveis para subsidiar a contagem, a etapa do ciclo de vida do desenvolvimento, dentre outros.

A contagem indicativa e estimada dar-se-á conforme definições da *Netherlands Software Metrics Association* – NESMA.

A orientação é que a contagem estimada seja utilizada para embasar a formalização (abertura) da Ordem de Serviço nos projetos de desenvolvimento e manutenção e para embasar a autorização para execução da mudança nos projetos de manutenção. Cabe ressaltar que, eventualmente, pode ser necessário o uso da contagem indicativa para a mesma finalidade quando houver pouco conhecimento do sistema que se precisa estimar.

Quanto à contagem detalhada, a orientação é que seja utilizada após a homologação e o aceite dos produtos gerados na Ordem de Serviço de desenvolvimento e manutenção para embasar o processo de pagamento. Entretanto, em caso de projetos grandes de longa duração, pode-se identificar a necessidade de realizar contagem detalhada de referência, com base nos requisitos detalhados, para gerar o equilíbrio financeiro do projeto.

É importante destacar que, quanto mais exata se quer uma contagem de pontos de função, mais detalhados devem ser os requisitos do usuário. Dessa forma, as contagens estimadas e indicativas, por não se tratarem de uma medição exata, podem apresentar desvios do tamanho obtido com a contagem detalhada. Um cuidado maior deve-se ter com o uso da contagem indicativa, pois os desvios podem chegar a até 50% na comparação de uma contagem indicativa e uma detalhada para o mesmo sistema, conforme demonstra o resultado de um projeto de pesquisa pela NESMA (<http://fattocs.com/pt/contagem-antecipada>).

4.9.1. Contagem Indicativa

A contagem indicativa é utilizada para estimar de forma rápida o tamanho de projetos de desenvolvimento de novas aplicações. Devido ao pouco conhecimento do sistema que se precisa estimar, ela é baseada somente na identificação de quantos arquivos lógicos (ALIs e AIEs) existirão na aplicação. A contagem indicativa é realizada da seguinte forma:

- determina-se a quantidade das funções do tipo dado (ALIs e AIEs);

- calcula-se o total de pontos de função não ajustados da aplicação da seguinte forma:

tamanho indicativo (pf) = 35 x número de ALIs + 15 x número de AIEs

É importante explicar que a contagem indicativa é baseada na premissa de que existem aproximadamente três EEs (para adicionar, alterar, e excluir dados do ALI), duas SEs, e uma CE na média para cada ALI, e aproximadamente uma SE e uma CE para cada AIE.

Dessa forma, obtém-se uma estimativa ágil do tamanho do sistema, porém não muito precisa.

4.9.2. Contagem Estimada

A contagem estimada é utilizada quando é possível identificar as funções do sistema, porém não se define a complexidade (tipos de dados, tipos de registros e arquivos referenciados).

Dessa forma, assume-se uma complexidade padrão para as funções, sendo as funções de dados (ALIs e AIEs) classificadas como de baixa complexidade, enquanto as funções transacionais (EEs, CEs e SEs) são classificadas como de média complexidade.

A contagem estimada é realizada da seguinte forma:

- determina-se todas as funções de todos os tipos (ALI, AIE, EE, SE, CE);
- toda função do tipo dado (ALI, AIE) tem sua complexidade funcional avaliada como Baixa;
 - toda função transacional (EE, SE, CE) é avaliada como de complexidade média;
 - calcula-se o total de pontos de função não ajustados.

4.9.3. Contagem Detalhada

A contagem detalhada dar-se-á pela aplicação da técnica de contagem seguindo as regras estabelecidas no Manual de Práticas de Contagem (CPM) versão 4.3.1 (ou superior) do IFPUG, acrescidas das definições deste Guia de Contagem do MP (versão mais atual) e do Roteiro de Métricas de Software do SISF na versão 2.0 (ou superior).

A contagem detalhada é a contagem usual de pontos de função e é realizada da seguinte forma:

- determina-se todas as funções de todos os tipos (ALI, AIE, EE, SE, CE);
- determina-se a complexidade de cada função (Baixa, Média, Alta);
- calcula-se o total de pontos de função não ajustados.

4.10. Fator de Ajuste

O Ministério do Planejamento não utiliza a avaliação do valor do fator de ajuste (VAF) em suas medições, sendo este fixado no valor 1,00. Isso significa que as características gerais do sistema, apresentadas no manual de práticas de contagem do IFPUG, não afetam o tamanho dos Pontos de Função obtidos em suas contagens.

Assim sendo, toda contagem deverá resultar em pontos por função brutos – PFB ou não ajustados.

4.11. Orientações para as contagens

4.11.1. Dado de Código

Segundo o CPM, os dados de código, às vezes chamados de dados de lista ou dados de tradução, fornecem uma lista de valores válidos que um atributo descritivo pode ter. Normalmente os atributos de dados de código são código, descrição e/ou outros atributos 'padrão' descrevendo o código; por exemplo, abreviação padrão, data de início de vigência, data de expiração, dados de trilha de auditoria, etc. Ao utilizar códigos em dados de negócio, é necessário ter meios de tradução para converter de código para algo mais reconhecível pelo usuário. Exemplos são valores válidos, descrições de códigos ou tabelas de tradução.

Alguns dados de código são desenvolvidos para atender requisitos específicos do usuário e contém dados que estão dentro do domínio do usuário. Outros dados de código podem ser derivados a partir dos requisitos do usuário para restringir os valores permitidos e também podem ser criados em uma tentativa de reduzir requisitos de espaço em disco. Os requisitos podem também incluir a habilidade de manter dados de código, mas todos esses são requisitos não-funcionais do usuário.

Portanto, mesmo que os requisitos do usuário incluam a habilidade de manter dados de código, isso não os transforma em requisitos funcionais, pois sua natureza é não funcional, conforme definição no CPM, "os dados de código são uma implementação de requisitos não-funcionais do usuário". Dado de código é implementação de requisitos técnicos e não influencia o seu tamanho funcional.

Para o MP, o desenvolvimento de funcionalidades consideradas dados de código, durante o projeto de desenvolvimento de software, não será considerado na contagem. Entretanto, devido a questões de contratações, para projetos de manutenção evolutiva, que implique manutenção nas funcionalidades consideradas dados de código, deve-se aplicar a orientação descrita no tópico Itens não Mensuráveis -Manutenção em Dados de Código.

Na contagem de transações que implementam requisitos funcionais e que acessam também dados de código, só devem ser considerados como

arquivos referenciados os ALIs e AIEs que implementam dados de negócio ou dados de referência e nunca dados de código.

4.11.2. Log, Trilha de Auditoria, Registro de Eventos e Histórico

O objetivo deste tópico é descrever o tratamento que o MP dará a respeito de Log, Trilha de Auditoria, Registro de Eventos e Histórico.

4.11.2.1. Histórico

Para o MP, o histórico é um registro de informações passadas em determinado momento, o que possibilita ao usuário consultar a evolução da informação na linha do tempo. Essas informações tornam-se necessárias para a prestação de contas (a órgãos externos, superiores ou processos internos) ou por exigência do próprio cenário de negócio. Sua existência é justificada pelo negócio, que sofre os impactos e consequências em caso de ausência de histórico. Para fazer parte do tamanho funcional, deve ser solicitado pelo gestor e deverá existir funcionalidade de consulta a tais dados. A função de consulta aos dados deverá ser contada de acordo com as regras de contagem das funções transacionais do CPM.

Nesse caso, o histórico será considerado um registro lógico do ALI relacionado.

Não devem ser consideradas na contagem funções de transação separadas para incluir, alterar e excluir as informações históricas, pois o armazenamento dessas informações é parte integrante das mesmas funcionalidades que processam os dados de negócio.

Observação: Quando o histórico for mantido de forma independente do registro principal e, por exemplo, o registro é excluído do ALI principal, mas o histórico manterá o registro excluído, o histórico se torna um ALI independente e não um registro lógico do ALI relacionado. Lembrando que, para fazer parte do tamanho funcional, deve ser solicitado pelo gestor e deverá existir funcionalidade de consulta a tais dados.

4.11.2.2. Registro de eventos

Para o MP, o registro de eventos tem o objetivo de armazenar os eventos associados à navegação e/ou ao acesso às funcionalidades do sistema para fins de monitoramento estatísticos ou criação de indicadores de uso do aplicativo.

O registro de eventos deve ser solicitado pelo gestor da aplicação e, para a contagem, deve ser considerado como um ALI, devendo existir funcionalidade de consulta a tais dados.

As informações de registro de eventos são consideradas como parte integrante da mesma funcionalidade que processa os dados de negócio. Dessa forma, não devem ser consideradas na contagem funções de transação separadas para incluir, alterar e excluir os dados de registro de eventos.

4.11.2.3. Log

O MP utilizará o termo “Log” como o registro de procedimentos ou ações realizados pela aplicação, em determinado período de tempo, com o objetivo de apoiar a auditoria do ambiente tecnológico e a identificação das causas raízes de falhas em sistemas.

Nesse caso, o log não deve ser mensurado, já que não armazena informações negociais reconhecidas pelo usuário da aplicação.

A principal diferença entre log e trilha de auditoria é:

- Log: apoia a auditoria no âmbito tecnológico (problemas decorrentes da tecnologia que precisam ser investigados, por meio da análise do conjunto de procedimentos executadas pela aplicação como por exemplo: baixa performance no sistema).
- Trilha de Auditoria: apoia a auditoria para os dados de negócio.

4.11.2.4. Trilha de auditoria

Para o MP, a trilha de auditoria tem o objetivo de armazenar informações referentes às ações realizadas pelos usuários da aplicação no passado, de modo que seja possível apurar quais foram as ações executadas quando da utilização do sistema. Para isso, devem existir no mínimo as informações para identificar quem realizou a ação (ID de usuário), quando e o que foi realizado, além de outras informações se necessário.

A trilha de auditoria deve ser solicitada pelo gestor da aplicação e, para a contagem, considera um registro lógico referenciado do ALI relacionado, devendo existir funcionalidade de consulta a tais dados.

Não devem ser consideradas na contagem funções de transação separadas para incluir, alterar e excluir os dados de trilha de auditoria, pois o armazenamento desses dados é parte integrante das mesmas funcionalidades que processam os dados de negócio.

4.11.3. Consultas com filtros diferentes e com as mesmas saídas

Trata-se de consultas com diferentes critérios de filtro, mas uma única saída idêntica em termos de campos.

Por exemplo, numa tela de consulta podem existir opções de filtros como pesquisa de empregados por lotação, data de admissão, data de nascimento, dentre outros, em que, quando não for especificado nenhum filtro, serão retornados todos os empregados de uma empresa, ou seja, a seleção dos filtros é opcional. Mas, caso sejam selecionados alguns filtros, poderá ser retornado nenhum ou vários empregados.

Para esse cenário, entende-se que os itens de dados e arquivos referenciados são os mesmos e o que difere são apenas os dados retornados em função dos parâmetros do filtro.

Nesse caso, considera-se que existe apenas um processo elementar de consulta, que pode ser classificado como CE ou SE.

No caso em que haja evidências de haver diferentes requisitos funcionais referentes a critérios mutuamente exclusivos indicando que a junção em uma única consulta foi opção de projeto, deverá ser avaliado se é o caso de considerar mais de um processo elementar.

4.11.4. Consultas com filtros iguais e com saídas diferentes

Essas consultas constituem processos elementares distintos e, segundo as regras de unicidade de Consultas Externas e Saídas Externas do CPM, devem ser contadas separadamente porque possuem itens de dados distintos na saída. Assim, se a aplicação tiver duas consultas com filtros iguais e saídas diferentes, devem ser contadas consultas separadas.

4.11.5. Integração entre aplicações

Este tópico descreve uma orientação de contagem a respeito de integração entre aplicações sem entrar no mérito de como será o projeto e a implementação dos requisitos. Tipicamente, esses cenários usam *webservices*, visões de banco de dados, *stored procedures* de banco de dados, sub-rotinas exportadas para utilização externa ao sistema (essas formas de implementação fazem parte de uma lista exemplificativa), podendo ser utilizados outros meios a depender de cada especificidade da tecnologia/plataforma utilizada na solução.

Esta orientação também ajuda extrapolar os requisitos funcionais a partir do projeto quando a especificação dos requisitos funcionais não fornece as informações suficientes para a identificação do fluxo de informação por meio da aplicação.

Para a explicação dos cenários 1 e 2, considera-se que os sistemas “A” e “B” sejam de fronteiras distintas.

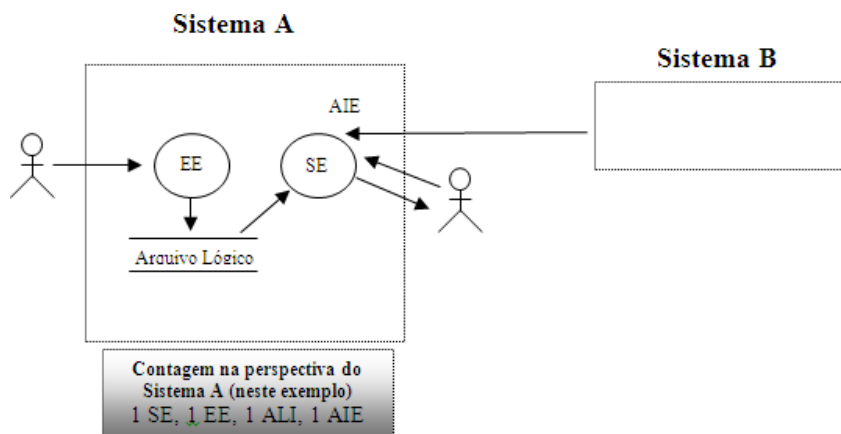
4.11.5.1. Cenário 1 (Sistema “A” requisita dados do Sistema “B”)

Este cenário se aplica quando o Sistema “A” precisa ler/consultar um conjunto de informações mantidas por uma aplicação externa denominada Sistema “B”.

Exemplo:

Existe um requisito no sistema “A” do tipo “Obter o saldo devedor do cliente no sistema de controle de contratos”. Avaliando o requisito, trata-se da referência a um grupo de dados logicamente relacionado (AIE), ainda que no sistema “B” não exista esse campo “saldo devedor” armazenado, já que pode ser calculado a partir de dados de vários arquivos. Nesse caso, na visão do sistema “A”, trata-se de um campo a ser recuperado do sistema “B” e, portanto, considera-se na contagem apenas um AIE independente da quantidade de arquivos lógicos que originaram os dados.

Diante deste cenário, é importante destacar que, para a contagem do AIE, o CPM define que um AIE deve ser um ALI em outro sistema, mas ele não estabelece que seja um, apenas um ou pelo menos um ALI. Com isso, este guia determina que seja utilizada a Visão do Usuário da aplicação em análise e, portanto, no exemplo acima, se o usuário considera que se trata de recuperar um campo, que assim seja medido independentemente de como isso é implementado em outro sistema.



4.11.5.2. Cenário 2 (Sistema “B” fornece dados para o Sistema “A”)

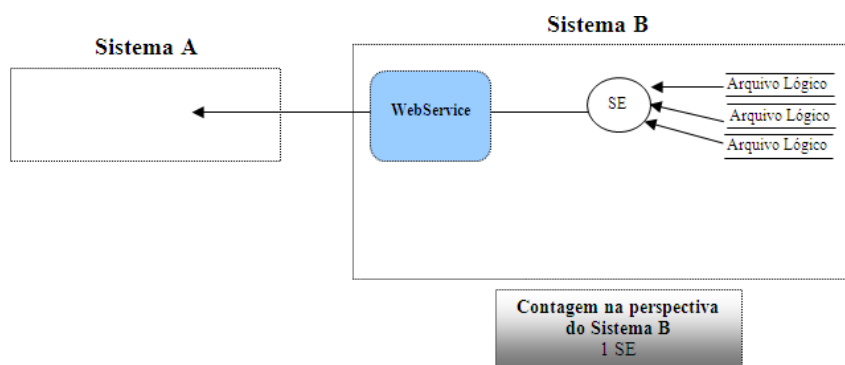
Este cenário se aplica quando o Sistema “B” disponibiliza informações para serem consumidas pelo Sistema “A”, sendo que as regras de negócio para gerar as informações para o Sistema “A” é de conhecimento do Sistema “B”.

Exemplo:

O sistema “B” precisa disponibilizar o saldo devedor do cliente para o sistema “A” utilizar numa transação qualquer.

Na visão do Sistema “B”, por ser de conhecimento dele a definição ou especificação das regras de negócio para a geração do saldo devedor do cliente, considera na contagem a função transacional do tipo “Saída Externa” para a disponibilização da informação.

Nesse caso, conta-se uma CE ou SE de acordo com as regras do CPM.



Caso haja necessidade de o Sistema “B” disponibilizar dados para fins de validação e referência, provendo esses dados, por exemplo, por meio de um webservice, mas não exista qualquer regra de negócio associada a essa construção, sendo que o motivo pelo qual a implementação dos requisitos de “A” se dá desta forma em “B” é por requisitos técnicos, não caberia a contagem de uma CE/SE na perspectiva do Sistema “B”. Nesse caso, entende-se que, se não fosse essa restrição, a aplicação poderia obter diretamente os dados sem qualquer intervenção por parte do Sistema “B”.

4.11.5.3. Cenário 3 (Disponibilização e consumo de dados dentro da mesma fronteira)

Neste cenário, existem dados sendo disponibilizados por meio de *webservices*, visões de banco de dados, *stored de procedures* de banco de dados, sub-rotinas, dentre outras implementações, para serem consumidos por funções transacionais dentro da mesma fronteira.

Diretriz de Contagem:

Nesse caso, as formas de disponibilizar os dados internamente à aplicação exercem o papel de componente interno ao software sendo medido e, portanto, a construção de componentes de código reutilizáveis não será contada em pontos de função.

As funções transacionais e os arquivos referenciados que utilizam os dados do *webservice*, por exemplo, seguem as orientações de contagem do CPM.

Apenas em casos de manutenção unicamente no componente, deve ser aplicada a regra descrita no tópico Manutenção de Componente Reutilizável.

4.11.5.4. Identificação do barramento como uma fronteira

Neste cenário, é necessário o desenvolvimento de funcionalidades no barramento para atender a uma necessidade comercial. Essas funcionalidades não são fornecidas pela ferramenta de barramento, ou seja, para atender a uma demanda comercial do gestor, será desenvolvida no barramento uma funcionalidade que é identificada como um processo elementar.

No que diz respeito à contagem, considera-se:

- Como usuários as aplicações que interagem com o barramento;
- Na fronteira do barramento, contam-se:
 - Os arquivos lógicos identificados;
 - As funções transacionais que serão desenvolvidas (CE, SE ou EE).
- Na fronteira dos outros sistemas, aplica-se o definido pelo CPM.

4.11.5.5. Configuração/Customização de ferramenta de barramento

Este cenário é identificado quando está sendo utilizada uma ferramenta de mercado com a função de barramento para fazer, por exemplo, orquestração de serviços entre dois ou mais sistemas.

Para este cenário são identificadas duas situações distintas:

- Configuração da ferramenta de barramento para possibilitar a comunicação;
- Customização da ferramenta de barramento para atender a uma necessidade específica de negócio.

Para efeito deste Guia de Contagem, essas duas situações não serão objeto de contagem por meio da técnica de Análise de Pontos de Função e, portanto, cabe ao MP definir a forma de mensurar e remunerar serviços que caem nessas situações.

4.11.6. Manutenção de Componente Reutilizável

Para projetos de desenvolvimento de novos sistemas, a construção de componentes de código reutilizáveis não afetará o tamanho da aplicação e nem a remuneração do serviço. Entretanto, a reutilização de componentes é considerada uma boa prática e deve, sempre que possível, ser utilizada, uma vez que o reuso proporcionará melhor produtividade, padronização e facilidade de manutenção.

Para projetos de manutenção/melhoria em um componente que é utilizado por várias funcionalidades da aplicação, esse componente será contado como uma funcionalidade e, por se tratar de manutenção, será aplicado o fator de impacto equivalente ao tipo de manutenção. Para efeito de teste da utilização deste componente pelas funcionalidades que dependem dele, cabe ao MP, após apresentada a lista de dependências deste componente, definir o escopo de teste indicando quais funcionalidades devem ser testadas. Para dimensionar o teste deste conjunto de funcionalidades, deve-se fazer uso da contribuição, em pontos de função, das funcionalidades a serem testadas em conjunto com o percentual da disciplina de teste constante da tabela Percentual de esforço por disciplina no ciclo de vida.

Exemplo: No sistema de Folha de Pagamento, o componente que valida o código do funcionário é utilizado por 25 processos elementares. Esse componente precisa ser alterado para considerar um novo departamento da

empresa que possuirá uma validação diferenciada. O Analista de Sistemas definiu que, para validar essa alteração, devem ser testados os seguintes processos: Incluir Funcionário, Alterar Funcionário, Selecionar Funcionário e Pesquisar Funcionário por código. Na contagem, será considerado: a contagem do componente que valida o código do funcionário como uma funcionalidade alterada; a contagem dos 4 processos elementares que serão testados, devendo ser dimensionados a partir da contribuição dessas funcionalidades de acordo com o percentual da disciplina de teste constante na tabela Distribuição em percentual do esforço por disciplina no ciclo de vida.

4.11.7. Múltiplas Mídias

Ao abordar o tema múltiplas mídias, o IFPUG definiu alguns termos comuns, a saber:

- **Canal:** também refere-se à mídia. Múltiplos canais é sinônimo de múltiplas mídias.
- **Mídia:** descreve a forma com que os dados ou informações se movimentam para dentro e para fora da fronteira de uma aplicação. Por exemplo, apresentação de dados em tela, impressora, arquivo, voz. Este termo é utilizado para incluir, dentre outros: diferentes plataformas técnicas e formatos de arquivos como diferentes mídias.

- Múltiplas Mídias: quando a mesma funcionalidade é entregue em mais de uma mídia. Frequentemente, somente uma mídia é requisitada para um usuário específico em um determinado momento. Por exemplo, consulta de extrato bancário via internet como oposto à consulta de extrato bancário via terminal do banco.
- Multimídia: quando mais de uma mídia é necessária para entregar a função. Por exemplo, uma nova notícia publicada na Internet que é apresentada em vídeo e texto. Observe que a notícia completa só é apresentada para o usuário se ele ler o texto e assistir ao vídeo.
- Abordagem *Single Instance*: esta abordagem não reconhece que a mídia utilizada na entrega de uma função transacional é uma característica de diferenciação na identificação da unicidade daquela função. Se duas funções entregam a mesma funcionalidade usando mídias diferentes, elas são consideradas como a mesma funcionalidade para propósitos de medição.
- Abordagem *Multiple Instance*: esta abordagem especifica que o tamanho funcional é obtido no contexto do objetivo da contagem, permitindo uma função de negócio ser reconhecida no contexto das mídias que são requisitadas para que a funcionalidade seja entregue. A abordagem *multiple instance* reconhece que a mídia para entrega constitui uma característica de diferenciação na identificação da unicidade da função transacional.

Para a aplicação das regras definidas no CPM, o IFPUG reconhece como válidas tanto a abordagem *single instance* quanto a abordagem *multiple instance*. A determinação da abordagem a ser seguida em uma contagem de

PF depende da avaliação da equipe de métricas da organização. As estimativas e contagens de pontos de função do MP deverão considerar a abordagem mais adequada à situação, conforme os cenários descritos nas seções seguintes.

É importante enfatizar que tais cenários não representam uma lista completa de situações de múltiplas mídias, mas o entendimento destes exemplos facilitará a compreensão de outros cenários envolvendo múltiplas mídias. Este guia de contagem será atualizado para considerar as novas diretrizes que forem publicadas pelo IFPUG e os novos cenários que emergirem das contagens de PF dos projetos do MP.

Cenário 1: Mesmos dados preparados para apresentação em tela, em arquivo formato texto e impressos

Neste cenário, uma aplicação apresenta uma informação em uma consulta em tela. Por requisição do usuário, a mesma informação pode ser impressa a partir da tela em questão ou disponibilizada em arquivo formato texto.

Nesses casos, deve ser utilizada a abordagem *single instance*, considerando que dados idênticos sendo apresentados em tela, em relatório impresso e em arquivo devem ser contados como uma única função. Portanto, apenas uma transação deverá ser incluída na contagem de pontos de função.

Cenário 2: Mesmos dados para entradas em lote (batch) e online

Neste cenário, uma aplicação pode receber informações por meio de dois métodos: arquivo batch e entrada de dados online. Tanto o arquivo batch quanto a entrada online executam validações durante o processamento. Se a lógica de processamento utilizada nas validações em modo batch for a mesma daquela utilizada em modo online, deve ser utilizada a abordagem *single instance*, e uma única transação deve ser incluída na contagem de pontos de função. Porém, se a lógica de processamento for diferente, deve-se utilizar a abordagem *multiple instance*, e duas transações distintas devem ser incluídas na contagem de pontos de função.

Cenário 3: Múltiplos canais de entrega da mesma funcionalidade

Neste cenário, uma funcionalidade deve ser disponibilizada em múltiplos canais. Por exemplo: consulta de dados em página web e consulta de dados em *smartphone*. Nesses casos, deve ser utilizada a abordagem *multiple instance*, tendo em vista que há necessidade de que sejam desenvolvidas duas “versões” da mesma funcionalidade: uma para rodar em navegador web e outra para rodar em plataforma móvel. Portanto, duas transações distintas devem ser incluídas na contagem de pontos de função.

Porém, se as “versões” forem geradas por uma ferramenta de apoio, sem necessidade de customização, deve ser utilizada a abordagem *single*

instance e somente uma transação deve ser incluída na contagem de pontos de função.

É importante destacar a existência de sites responsivos (ou sites flexíveis), que é quando o site é desenvolvido para que de forma automática se encaixe no dispositivo do usuário (PC, celular, tablete, etc.), ou seja, um site responsivo muda a aparência e disposição com base no tamanho da tela em que o site é exibido. Nesse caso, entende-se que deve ser utilizada a abordagem *single instance*.

Cenário 4: Relatórios em múltiplos formatos

Neste cenário, um relatório deve ser entregue em diferentes formatos. Por exemplo: em formato HTML, DOC, XLS, XML, HTML, DOC, CSV, JSON, dentre outros. O usuário terá a oportunidade de escolher em qual formato deseja que os dados sejam apresentados.

Nesses casos, deve ser considerada a ferramenta de desenvolvimento utilizada na geração dos relatórios. Se tiver sido necessária a construção do relatório nos dois formatos, deve ser utilizada a abordagem *multiple instance*, devido à ocorrência da lógica de processamento de análise de condições para verificar quais são aplicáveis, e, portanto, duas transações distintas devem ser incluídas na contagem de pontos de função. Porém, se a ferramenta de desenvolvimento suportar um gerador de relatórios que permita ao usuário visualizar o relatório em tela, imprimir-lo e também salvá-lo em múltiplos formatos, deverá ser utilizada a abordagem *single instance*, tendo em vista que o tal relatório será construído apenas uma vez. Logo, apenas uma transação deve ser incluída na contagem.

Observação 1: Para que uma transação possa ser contada em termos de múltiplas mídias, o requisito funcional deve deixar claro que a funcionalidade deve ser entregue em múltiplas mídias.

Observação 2: Quando existir um requisito dizendo que todo relatório deve ser emitido em PDF e DOC, por exemplo, pelo fato de ser um requisito generalista, ou seja, para todos os relatórios, deve ser avaliado se não se trata de um requisito não funcional.

4.11.8. Múltiplos processos compondo um processo elementar (Subdivisão de Funcionalidades)

Para correta quebra de uma funcionalidade em diversos processos elementares, é importante avaliá-las a partir de uma perspectiva do negócio, verificando quais funcionalidades são completas e reconhecidas pelos usuários do negócio. Por exemplo, um processo elementar com telas encadeadas, em que, quando não concluído o processo, deve ser reiniciado desde a primeira tela, é errado considerar cada tela (ou aba) como um processo elementar.

Cenário: Um único Processo Elementar identificado para a funcionalidade

Por exemplo, se uma funcionalidade é dividida em “abas”, é necessário avaliar se cada uma das telas constitui um processo elementar, atendendo principalmente ao requisito de ser autocontido. Nesse caso, a primeira coisa a se fazer é a de tentar identificar a razão que levou a divisão da funcionalidade em diversas telas, pois é comum que formulários de cadastro sejam quebrados em etapas com o objetivo apenas de tornar a atividade de cadastro mais intuitiva e organizada, ou seja, apenas para atender a requisitos não funcionais de usabilidade. Dessa forma, apenas um único processo elementar pode ser identificado.

Caso contrário, se a funcionalidade tiver sido subdividida em diversas telas para atender a uma necessidade do negócio (Ex: Um departamento possui a competência para o preenchimento de uma das telas enquanto que outro departamento para as demais), isso é indício de que pode se tratar mais de um processo elementar.

Seguem algumas orientações para auxiliar o analista de métricas em cenários semelhantes:

- Verificar se, caso a funcionalidade não fosse fragmentada e houvesse uma única tela, a necessidade de negócio seria atendida independente de a funcionalidade ser menos usual.
- Avaliar se há usuários de áreas de negócio distintas responsáveis por preencher telas específicas da funcionalidade, não tendo competência (mesmo que munidos de todas as informações necessárias) para o preenchimento completo do formulário.

4.11.9. Consultas Implícitas

São consultas que apresentam dados para o usuário (geralmente precedendo outra transação a ser realizada), mas que não estão claramente explícitas nos requisitos ou no próprio sistema. Podem ser classificadas como CE ou SE.

Geralmente são comuns em telas de alteração ou exclusão de registros de um arquivo, pois, antes de alterar ou excluir o registro, os dados são apresentados ao usuário e na sequência o usuário efetua a alteração ou exclusão. Cabe ressaltar que, quando essa consulta é idêntica a uma consulta explícita, apenas um processo elementar deve ser contado.

4.11.10. Contagem de Funções para Conversão de Dados

Conforme consta no CPM, a funcionalidade de conversão de dados existe quando há requisitos para migrar ou converter dados durante um novo projeto de desenvolvimento, projeto de melhoria. Assim, essas funcionalidades de conversão de dados deverão ser contadas como Entrada Externa para a funcionalidade de migração ou carga inicial dos dados e Consultas ou Saídas Externas quando forem requisitados pelo usuário relatórios associados à funcionalidade de migração de dados. Os arquivos do antigo sistema que está sendo substituído não devem ser contados como Arquivos de Interface Externa

(AIEs) da nova aplicação e as extrações dos dados do sistema antigo, se for o caso, não serão contadas como CEs ou SEs.

4.11.11. Projetos de Migração de Dados

A critério do MP, quando for identificada complexidade no processo de migração de dados, os pontos de função de conversão (PF_CONVERSÃO) poderão ser suprimidos das fórmulas de contagem de pontos de função de Projetos de Desenvolvimento e de Melhoria e as funcionalidades de migração de dados podem ser tratadas de forma separadas como projetos de migração de dados, mas aplicando integralmente os conceitos do IFPUG.

4.11.12. Projetos de Melhoria (Manutenção Evolutiva)

Trata-se de projeto de manutenção evolutiva ou melhoria funcional. Seu tamanho funcional é a medida das funcionalidades incluídas, alteradas e excluídas em aplicações em produção. O tamanho funcional de um projeto de melhoria pode incluir o tamanho da funcionalidade de conversão de dados.

As regras de como contar as funções de dados e de transação em um projeto de melhoria são as mesmas apresentadas no Manual de Práticas de Contagem (CPM), porém devem ser consideradas as orientações descritas neste tópico, pois, para calcular o tamanho funcional do projeto de melhoria, devem ser aplicados os fatores de impacto definidos nas funções alteradas e excluídas, conforme fórmula abaixo.

Segue a Fórmula:

$$PF_MELHORIA = PF_INCLUÍDO + (FI \times PF_ALTERADO) + (0,3 \times PF_EXCLUÍDO) + PF_CONVERSÃO$$

Definições:

PF_INCLUÍDO = pontos de função das funções que estão sendo adicionadas pelo projeto de melhoria.

PF_ALTERADO = pontos de função das funções que sendo alteradas pelo projeto de melhoria (como elas são ou serão após a implementação).

PF_EXCLUÍDO = pontos de função das funções sendo excluídas pelo projeto de melhoria.

PF_CONVERSÃO = pontos de função das funcionalidades de conversão de dados quando existirem.

O **Fator de Impacto (FI)** pode variar conforme condições abaixo:

- **FI = 50%** para funcionalidade de sistema que possua documentação.

- **FI = 60%** para funcionalidade de sistema que não possua documentação.

Observação 1: Destaca-se que deve haver correção e/ou atualização da documentação das funcionalidades da aplicação impactadas pela manutenção evolutiva quando a funcionalidade possuir documentação atualizada.

Observação 2: Quando a funcionalidade impactada pela manutenção evolutiva não possuir documentação ou a documentação estiver desatualizada, deverá ser contemplada a redocumentação ou atualização da documentação das funcionalidades.

Observação 3: Se durante o atendimento de uma demanda de projeto de melhoria for detectada a manutenção evolutiva e corretiva na mesma funcionalidade, para efeito de contagem, deve ser avaliado se as duas manutenções podem ser disponibilizadas juntas no ambiente produtivo. Confirmada a disponibilização em conjunto, o MP entende que deve ser contada apenas a manutenção evolutiva. Caso contrário, a contagem da manutenção corretiva deve ser feita separada da manutenção evolutiva, desde que a correção não esteja coberta pela garantia. Para a contagem da correção, devem ser seguidas as regras descritas no tópico de Manutenção Corretiva deste guia.

4.11.13. Mudança estrutural em Arquivos Lógicos para Projetos de Melhoria

Segundo o CPM, para que uma função de dado seja contada como uma função alterada, é obrigatório que a função seja estruturalmente alterada.

Mudança estrutural é toda inclusão ou exclusão de atributo em um arquivo lógico ou alteração de suas características (alteração de tamanho, tipo – numérico para alfanumérico, etc.), desde que a mudança decorra de alteração de regra de negócio. Simples alterações de valores válidos em um ALI não serão consideradas mudanças estruturais.

4.11.13.1. Impacto das alterações das características de itens de dados de um ALI nas funções transacionais que o mantém.

Quando as mudanças estruturais em uma função de dados implicarem mudança de lógica de processamento nas funções transacionais como, por exemplo, mudança em validações, as funções transacionais impactadas devem ser consideradas alteradas em um projeto de melhoria.

Exemplo: Numa aplicação, o gestor solicitou que o campo de número do telefone residencial do cliente passe a suportar até oito dígitos. Além disso, foi solicitado que, nas funcionalidades de inclusão e alteração de clientes, caso o cliente resida no Distrito Federal, seja obrigatório que seu telefone residencial tenha oito dígitos, sendo que o primeiro dígito à esquerda seja igual a três.

Nesse caso, observa-se alteração na lógica de processamento das entradas externas de inclusão e alteração de clientes e, portanto, ambas seriam pontuadas na manutenção evolutiva como “alteradas”. As funcionalidades que não sofreram alteração em decorrência da mudança do DER não são pontuadas.

Quando as mudanças estruturais em uma função de dados não implicarem mudança de lógica de processamento nas funções transacionais, o simples fato de o DER alterado cruzar a fronteira da aplicação nas transações que o mantêm ou referenciam **não** é suficiente para que essas transações sejam consideradas como alteradas na contagem da manutenção evolutiva.

Exemplo: Suponha que deva ser feita uma mudança do número do telefone de sete para oito dígitos e do CEP de cinco para oito dígitos. Se somente for aplicada a alteração na função de dados, não sendo identificada a necessidade de alterar qualquer lógica de processamento nas transações que utilizam essa função, só deve ser considerada no projeto de melhoria a mudança na função de dados.

4.12. Itens não mensuráveis

Algumas manutenções em softwares não são passíveis de medição segunda a técnica Análise de Pontos de Função, mas, devido à existência dessas demandas e aos contratos de prestação de serviços, para calcular o esforço das atividades envolvidas nesses tipos de manutenção, houve uma equiparação do esforço ao ponto de função e, portanto, seguem as orientações descritas abaixo. Os percentuais são estimados, podendo ser reajustados conforme avaliação da base histórica dos serviços realizados no MP.

Os exemplos descritos neste tópico não são exaustivos, ou seja, identificando-se novos itens não mensuráveis, eles devem ser adicionados a essa lista em versões posteriores deste guia.

Caso seja identificado algum item não contemplado neste tópico, deverá ser encaminhada solicitação fundamentada ao MP, que analisará a pertinência da inclusão de um novo item não mensurável.

Importante:

A medição não é cumulativa dentro da mesma funcionalidade, ou seja, caso uma funcionalidade possua itens mensuráveis e itens não mensuráveis (uma alteração no processo elementar e uma alteração de layout na mesma tela, por exemplo), apenas os itens mensuráveis devem ser contados.

4.12.1. Manutenção Corretiva

A manutenção corretiva consiste na correção e alteração de requisitos, projeto, código, conjunto de testes e documentação para correção de defeitos. Encontram-se nesta categoria as demandas de correção de erros (bugs) de

funcionalidades em sistemas, comportamentos inadequados que causem problemas de uso.

A estimativa e dimensionamento de tamanho de manutenções corretivas em pontos de função devem levar em consideração a documentação do sistema disponível e os artefatos a serem mantidos.

Observação 1: Destaca-se que deve haver correção e/ou atualização da documentação do sistema. Caso não exista documentação para as funcionalidades alteradas, não será contemplada a redocumentação das funcionalidades da aplicação impactadas pela manutenção corretiva.

Observação 2: Se durante a investigação do problema for constatado que não há erro no sistema, a demanda deverá ser atendida de acordo com o tópico Verificação de Erros do Roteiro SISP.

Observação 3: Quando o sistema em produção estiver dentro da garantia da qualidade, a manutenção corretiva será do tipo Garantia, conforme prazos e demais cláusulas do contrato em questão e não incorrerão em ônus financeiro para o MP.

Os deflatores são definidos de acordo com os cenários abaixo.

4.12.1.1. Sistema sem documentação, desatualizada ou incompleta

Neste caso, deve-se aferir o tamanho em pontos de função das funcionalidades impactadas e considera 60%.

Segue a Fórmula:

$$PF_INM = PFfi * 0,60$$

Onde:

PF_INM: valor em pontos de função do item não mensurável
PFfi: total de pontos de função das funcionalidades impactadas

4.12.1.2. Sistema com documentação completa e atualizada

Neste caso, deve-se aferir o tamanho em pontos de função das funcionalidades impactadas e considera 50%.

Segue a Fórmula:

$$PF_INM = PFfi * 0,50$$

Onde:

PF_INM: valor em pontos de função do item não mensurável

PFfi: total de pontos de função das funcionalidades impactadas

4.12.2. Manutenção de Interface

Contemplam as alterações de interface, por exemplo: layouts de telas, fonte de letra, cores de telas, logotipos, mudança de botões na tela, mudança de posição de campos ou texto na tela, máscaras, layout de relatórios ou de arquivos, divisão de telas e/ou relatórios, sem que haja alteração em elementos de dados, arquivos referenciados ou informações de controle.

Nestes casos, deve-se aferir o tamanho em Pontos de Função da funcionalidade ou das funcionalidades impactadas e considera 20% da contagem de uma função transacional de mais baixa complexidade (3 PF), ou seja 0,6 PF.

Caso seja utilizada uma mesma tela para duas ou mais funcionalidades, deve ser contada **apenas uma** função transacional.

Observação 1: Destaca-se que deve haver correção e/ou atualização da documentação do sistema. Caso não exista documentação para as funcionalidades alteradas, não será contemplada a redocumentação das funcionalidades da aplicação impactadas pela manutenção nas demandas desta categoria.

Observação 2: Este percentual não é cumulativo, ou seja, caso duas ou mais alterações de layout sejam realizadas na mesma tela, o percentual será aplicado apenas uma vez.

Segue a Fórmula:

$$PF_INM = PFfi * 0,6 PF$$

Onde:

PF_INM: valor em pontos de função do item não mensurável

PFfi: quantidade de funções transacionais impactadas

4.12.3. Manutenção em Dados de Código

Contempla a necessidade de criar ou alterar funcionalidades de cadastro, alteração, exclusão e consulta de dados de código, desde que sejam mantidas e solicitadas pelo usuário.

Nesses casos, para as funções transacionais, aplica-se 50% da função de mais baixa complexidade (3 PF) e, para as funções de dados, aplica-se 50% do ALI de mais baixa complexidade (7 PF).

Segue a fórmula:

$$PF_INM = PFfi * FI$$

Onde:

PF_INM: valor em pontos de função do item não mensurável

PFfi: quantidade de funcionalidades impactadas

FI: 1,5 PF para funções de transação e 3,5 PF para funções de dados.

Observação 1: Esse tópico não se aplica para atividades de inclusão, alteração e exclusão de dados de domínio em tabelas classificadas como dados de código, o qual é tratado pelo tópico Atualização de Dados de Código.

Observação 2: Conforme consta no tópico Dados de Código, durante o projeto de desenvolvimento de software, não será mensurado em pontos de função o desenvolvimento de funcionalidades consideradas dados de código. Este tópico aplica-se somente para manutenções.

4.12.4. Atualização de Dados de Código

Contemplam a necessidade de inclusão, alteração ou exclusão de dados pertencentes a listas (combo box) ou tabelas físicas.

Neste caso, considera-se 50% da contagem de pontos de função de uma Entrada Externa de complexidade baixa, ou seja, 1,5 PF por lista ou tabela física alterada, incluída ou excluída.

Segue a fórmula:

$$PF_INM = PFfi * 1,5 PF$$

Onde:

PF_INM: valor em pontos de função do item não mensurável

PFfi: quantidade de listas ou tabelas físicas impactadas

4.12.5. Mensagens

Contemplam a necessidade de alterações de mensagens de retorno ao usuário, desde que não acessem ALI ou AIE.

Nesses casos, deve-se aferir o tamanho em pontos de função das funcionalidades impactadas e aplicar um redutor de 10%.

Segue a Fórmula:

$$PF_INM = PFfi * 0,10$$

Onde:

PF_INM: valor em pontos de função do item não mensurável

PFfi: total de pontos de função das funcionalidades impactadas

Observação: Está contemplada a atualização da documentação das funcionalidades impactadas.

4.12.6. Programas Auxiliares

São programas compostos de procedimentos de atualização da base de dados, que podem ser desenvolvidos por meio de scripts ou não, a serem executados apenas uma vez para corrigir dados incorretos ou para atualizar dados em bases de dados de aplicações; ou ainda para gerar um relatório específico ou um arquivo para o usuário por meio de recuperação de informações armazenadas na aplicação.

Deve-se destacar que esses programas não fazem parte da aplicação e são executados apenas uma vez, podendo ser descartados em seguida.

Para serem considerados em uma contagem de pontos de função, os programas auxiliares devem figurar claramente nos requisitos do usuário.

Se, para atender à necessidade do usuário for necessário escrever um script para consultar o banco de dados, uma CE ou SE deverá ser considerada na contagem. Se for necessário atualizar, deletar ou incluir dados no banco, uma EE deverá ser considerada na contagem de pontos de função, em que as tabelas envolvidas serão ALR e os atributos envolvidos serão DER para o processo elementar EE.

É importante ressaltar que as funções de dados associadas aos dados atualizados não devem ser contadas, considerando que não há mudanças nas estruturas dos arquivos lógicos.

Nesses casos, deve-se medir o tamanho funcional do programa auxiliar construído.

Por exemplo, se para atender a uma necessidade do usuário for criado um programa que realize procedimentos (inserção, consulta, atualização ou exclusão) em registros de várias tabelas, não será considerado na contagem um programa separado para o procedimento realizado em cada tabela do banco de dados envolvida, mas sim será contado um único programa auxiliar criado para atender a necessidade do usuário.

Este tópico substitui o conceito de Apuração Especial, utilizado no Roteiro SISP.

Considerações sobre Programas Auxiliares:

1. Todo programa auxiliar deve ser documentado, devendo a documentação conter, minimamente, um objetivo, requisitante, data de requisição, descritivo funcional, data de descarte (nunca inferior a 18 meses);
2. Os Programas Auxiliares devem ser armazenados, pelo período mínimo descrito em sua documentação, de forma a estarem disponíveis para re-execuções quando solicitado;
3. Quando houver reexecução de Programas Auxiliares, deve ser aplicado um deflator de 5% sobre o programa construído. A reexecução será considerada quando:
 - For reexecutado o mesmo programa (idêntico ao original);
 - Forem aplicados ajustes no script para a troca de parâmetros, por exemplo, nos campos de inclusão/alteração, nas chaves de pesquisa, dentre outros. Entende-se que esses tipos de ajustes não são caracterizados alterações de estrutura do script original.

Segue a fórmula:

$$PF_INM = PFfi * 0,05$$

Onde:

PF_INM: valor em pontos de função do item não mensurável

PFfi: total de pontos de função do script construído

4. Quando houver reexecução de Programas Auxiliares que sofreram alterações na estrutura do script original, de forma análoga a um projeto de melhoria numa funcionalidade, deve ser aplicado um deflator de 50% sobre o programa construído.

Segue a fórmula:

$$PF_INM = PFfi * 0,5$$

Onde:

PF_INM: valor em pontos de função do item não mensurável

PFfi: total de pontos de função do programa construído

4.13. Mudança de requisitos durante o projeto de desenvolvimento ou manutenção

Em um processo de desenvolvimento de software, é comum mudanças de requisitos durante o andamento do projeto ou da manutenção.

Quando essas alterações ocorrerem, primeiramente é necessário identificar se as mudanças de requisito estão relacionadas a:

- Alterações em funcionalidades que ocorreram em virtude de mudanças no domínio do negócio, como alteração de escopo, regras de negócio ou mudanças legais/regulamentares durante a execução da *release* ou;
- Alterações referentes à evolução natural dos requisitos, provocada, neste último caso, pelo aprofundamento, detalhamento e complementação de requisitos durante o processo de desenvolvimento ágil.

O segundo caso (alterações inerentes ao desenvolvimento ágil) é tratado no tópico Contagem de pontos de função no desenvolvimento de software com métodos ágeis.

Quanto ao primeiro caso, seguem as orientações.

4.13.1. Mudanças associadas ao domínio do negócio

Para tratar o dimensionamento das mudanças de requisitos, decorridas do domínio do negócio, deve-se levar em consideração o trabalho executado até o momento da solicitação de mudança a fim de apurar o retrabalho gerado pela mudança. Nesse caso, sugere-se a utilização de percentual por disciplina do ciclo de desenvolvimento, conforme os valores definidos na Tabela 1. Ressalta-se que esses valores são apenas uma referência, podendo ser ajustados e redefinidos em contratos.

Disciplina	Percentual de esforço
Engenharia de Requisitos	25%
Análise e Design	10%
Implementação	40%
Teste	15%
Homologação	5%
Implantação	5%

Tabela 1 – Distribuição em percentual do esforço por disciplina no ciclo de vida.

Para realizar o cálculo dos percentuais das disciplinas executadas, devem-se considerar somente as disciplinas que tiveram seus produtos reconhecidos pelo MP como concluídos.

Quando for utilizado um processo de desenvolvimento incremental, ágil, deve ser avaliada a viabilidade pelo MP sobre a adoção de percentuais por

disciplina, pois, em determinados cenários, pode ser inviável calcular o percentual de participação de cada disciplina.

A Tabela 2 resume os percentuais que devem ser aplicados sobre as funções alteradas (considerando o tamanho antes da mudança) para a obtenção do PF_RETRABALHO.

Natureza da Mudança	Acréscimo (%)
Inclusão de Requisito	0%
Alteração de Requisito	50%
Exclusão de Requisito	30%

Tabela 2 – Percentuais definidos para a mudança de requisitos

O processo de cálculo do retrabalho, considerando o percentual das disciplinas executadas, ocorre da seguinte forma:

Calcula-se o total de pontos de função das funcionalidades que sofreram mudança nos requisitos, baseando-se no requisito antes da mudança, e aplica-se o percentual do fator de impacto referente ao tipo de mudança de requisito. Em seguida, multiplica-se pelo percentual das disciplinas executadas até o momento da solicitação da mudança, conforme os percentuais referenciados na Tabela 1.

Segue a fórmula:

$$\text{PF_Retrabalho} = (\text{PFfi} * \text{FI}) * \text{P}$$

Onde:

PFfi: pontos de função da funcionalidade impactada antes da alteração.

FI: Fator de Impacto referente ao tipo de mudança, conforme percentuais da Tabela 2.

P: Somatório do percentual das disciplinas executadas.

Para exemplificar a aplicação dos cálculos descritos acima, o cenário abaixo é apresentado.

Suponha que, ao final da etapa de implementação de um novo relatório de clientes, seja solicitada uma alteração para acrescentar a exibição de uma nova informação. A equipe de desenvolvimento terá um retrabalho de algumas disciplinas do ciclo de vida. Assim, o tamanho dessa mudança deve ser calculado da seguinte maneira:

- Tamanho do relatório de clientes (Antes da Alteração)
 Função: Saída Externa
 Complexidade: Média
 Contribuição: 5PF
- Tamanho do relatório de clientes (Após a Alteração)
 Função: Saída Externa
 Complexidade: Média
 Contribuição: 5PF

O cálculo do tamanho da funcionalidade relatório de clientes será definido conforme Tabela 3:

Disciplina executadas	Percentual de esforço
Engenharia de Requisitos	25%
Análise e Design	10%
Implementação	40%
Total de disciplinas executadas	75%

Tabela 3 - Disciplinas executadas para a confecção do relatório antes da mudança

Assim, aplicando a fórmula para o exemplo, tem-se o seguinte resultado:

$$PF_Retrabalho = (5PF * 50\%) * 0,75 = 1,875 PF$$

Ao final do projeto, o pagamento referente ao desenvolvimento dessa funcionalidade de Relatório de Clientes seria considerado conforme expresso na Tabela 4.

Descrição	Tamanho faturável
Relatório de Clientes - PF_Retrabalho	1,875 PF
Relatório de Clientes (Requisitos Alterados)	5 PF
Total de PF para pagamento:	6,875 PF

Tabela 4 - Demonstrativo de PF para pagamento

Para efeito de gestão das mudanças e geração de indicadores, recomenda-se que as demandas de mudanças, de acordo com as características descritas neste tópico, sejam contadas em planilha separada da contagem do projeto de desenvolvimento ou da manutenção.

4.13.2. Desistência de incluir, alterar ou excluir uma função

Quando a mudança no projeto ou demanda for referente à desistência de incluir, alterar ou excluir funcionalidades, a diretriz é:

Para o cálculo do PF_Retrabalho neste tipo de mudança, deve-se considerar o requisito original e aplicar os percentuais que constam na Tabela 2 de acordo com o tipo de mudança de desistência.

Entretanto, antes de aplicar os percentuais relacionados ao tipo de mudança, o MP deve avaliar junto à contratada o esforço efetivamente aplicado na execução desse tipo de mudança para negociar a remuneração somente do esforço gasto.

Por exemplo, numa requisição de mudança que solicite a exclusão de uma funcionalidade, talvez o retrabalho seja apenas de descarte de algum artefato que compõe a documentação do sistema e, neste caso, pode ser que o esforço não justifique a utilização do percentual para excluir a funcionalidade.

Dessa forma, a aplicação desses percentuais dependerá de uma decisão do MP, após a devida análise técnica da situação e a comprovação pela contratada de todo o retrabalho referente à desistência do requisito.

A Tabela 5 resume os percentuais que devem ser aplicados quando a mudança se referir à desistência de incluir, alterar ou excluir uma função (considerando o tamanho antes da mudança) para a obtenção do PF_RETRABALHO.

Fator	Requisito Original		
	Incluir Função	Alterar Função	Excluir Função
Mudança de requisito de desistência	130%	80%	30%

Tabela 5 – Percentual para mudança de requisitos de desistência

É importante destacar que cabe ao MP decidir pela adoção do percentual por disciplina do ciclo de desenvolvimento durante o cálculo do PF_RETRABALHO nesta situação de mudanças referentes à desistência de incluir, alterar ou de excluir funcionalidades que constavam no escopo do projeto desde que as regras estabelecidas em contrato não sejam infringidas.

4.14. Contagem de pontos de função no desenvolvimento de software com métodos ágeis.

O objetivo deste tópico é descrever orientações sobre a aplicação da Análise de Pontos de Função em projetos de desenvolvimento de software com métodos ágeis para subsidiar os processos de contratação de desenvolvimento e manutenção de software no MP.

Este tópico foi elaborado por meio de um estudo preliminar que considerou o seguinte cenário:

- Processo de desenvolvimento de software orientado a metodologias ágeis;
- Divisão do projeto de desenvolvimento ou manutenção em *releases*;
- Utilização de contagem estimativa do projeto definindo o tamanho estimado de cada uma das *releases* para efeito de planejamento do projeto e geração das Ordens de Serviço de desenvolvimento ou manutenção de software;
- Utilização dos conceitos de *release*, *Sprint*, produto pronto, *Delta*, que constam abaixo.

Release: Definida como um ciclo que perpassa sequencialmente pelas fases do processo de desenvolvimento de software com o objetivo de entregar, ao final do ciclo, um produto pronto. Com base nessa definição, entende-se que um produto pronto está associado aos produtos entregues e aceitos ao final de cada *release*. Recomenda-se definir um tempo não muito longo para a *release*, uma vez que o conceito de produto pronto estaria associado a seu término. Logo, sugere-se um tempo entre 2 e 4 meses.

Produto Pronto: Com base na definição de *release*, entende-se que o produto pronto está associado aos produtos entregues e aceitos com a aprovação do dono do produto ao final de cada *release*.

Sprint: Uma unidade de período, com datas de início e fim pré-definidas, dentro da qual é executado um conjunto de atividades previamente estabelecidas.

Delta%: Valor percentual definido pela instituição ou órgão, utilizado para comparar o volume das alterações inerentes ao processo ágil e a contagem detalhada final da *release*.

Considerando os aspectos do desenvolvimento ágil, é prática comum a existência de evoluções em uma funcionalidade durante a execução das *sprints* de uma *release*. Assim, este guia orienta que as alterações ocorridas dentro dessas características até o valor de Delta% em relação ao tamanho da *release* não sejam remuneradas de forma adicional ao total de pontos de função da contagem detalhada final da *release*, pois se entende que são relativas à evolução natural de requisitos, o que é uma característica inerente ao processo ágil e deve ser considerado na definição do valor do ponto de função.

O cálculo do quantitativo de pontos de função de cada uma das *sprints* de uma *release* é de decisão e responsabilidade da empresa contratada, pois, para reivindicar e fazer jus ao recebimento do volume de alteração que

ultrapassar o *Delta%* previsto, a contratada deverá apresentar memórias de cálculos e evidências dos produtos gerados antes e depois das evoluções.

O Ministério do Planejamento Orçamento e Gestão (MP), através de sua Diretoria de Tecnologia da Informação (DTI), com base em experiências dos projetos com desenvolvimento ágil, calculou seu fator *DELTA em 30%*.

Considera-se que o ciclo de trabalho evolutivo inerente ao desenvolvimento ágil encerra-se a cada *release*. Dessa forma, as alterações que ocorrerem em funcionalidades existentes após o término da *release* devem ser tratadas como projeto de melhoria.

Com relação às alterações em funcionalidades ocorridas em virtude de mudanças no domínio do negócio, devem ser aplicadas as regras descritas no tópico Mudanças associadas ao domínio do negócio.

Na planilha onde será registrada a contagem referente à requisição de mudança, devem ser consideradas todas as funcionalidades incluídas, alteradas e excluídas, porém, para efeito de faturamento, as funcionalidades incluídas serão remuneradas somente na contagem da *release* a fim de não existir duplicidade de remuneração.

Cálculo do Fator Evolutivo Ágil

Para o cálculo do volume de evoluções inerentes ao processo ágil (**Fator Evolutivo Ágil**) de uma *release*, devem ser aplicados os seguintes passos:

1. Ao final de cada *sprint*, deve ser efetuada, pela contratada, uma contagem das funcionalidades solicitadas e entregues;
2. Realizar, ao final da *release*, uma contagem final detalhada da *release*;
3. Somar a contagem de todas as *sprints* executadas na *release*;
4. Calcular o **Fator Evolutivo Ágil**, que é o volume das alterações realizadas ao longo do ciclo da *release* de acordo com a fórmula abaixo:

$$Fator\ Evolutivo\ Ágil = \left(\left(\frac{\sum PF\ Sprints}{PF\ Release} \right) - 1 \right) * 100$$

Onde:

$\sum PF\ Sprints$ = somatório das contagens de pontos de função das *sprints*.

PF Release = contagem final detalhada da *release*.

Observação:

- Na contagem das *sprints*, são consideradas as funções desenvolvidas e as alterações decorrentes da evolução dos requisitos.
- Na contagem final da *release*, são consideradas apenas as funções desenvolvidas na *release*.
- Com relação às alterações referentes à evolução de requisitos ocorridas entre as *sprints* da *release*, devem ser aplicadas as regras descritas no tópico Projeto de Melhoria, levando em consideração o fator de impacto de 50% para funcionalidades alteradas e 30% para as excluídas.

5. Após o cálculo do Fator Evolutivo Ágil, compara-se o resultado com o **Delta%** e verifica-se:

- Se o **Fator Evolutivo Ágil** for menor ou igual ao **Delta%**, então não haverá remuneração adicional à contratada além do valor referente à contagem da *release*.
- Se o **Fator Evolutivo Ágil** for maior que o **Delta%**, então deve ser calculado o quantitativo de pontos de função excedentes a **Delta%** (**PF Excedente**) a fim de fazer a remuneração adicional. O cálculo ocorre da seguinte forma:

$$FP\ Excedente = \left(\sum PF\ Sprints - PF\ Release \right) - (PF\ Release * Delta\%)$$

Onde:

PF Excedente = Quantidade de pontos de função excedentes ao Delta %.

$\sum PF\ Sprints$ = somatório das contagens de pontos de função das *sprints*.

PF Release = contagem final detalhada da *release*.

Delta% = Fator Delta definido pela instituição ou órgão.

Para exemplificar a aplicação dos cálculos descritos acima, são apresentados dois cenários. Nesses cenários, as funcionalidades alteradas que constam na contagem das *sprints*, registradas na planilha de contagem, são consideradas como evolução dos requisitos inerente ao processo de desenvolvimento ágil.

Cenário 1 – Cenário em que o volume das evoluções não ultrapassa o **Delta%** definido com o valor de 30%.

Release 1 (composta de 3 Sprints)	Nome da Função	Tipo	Complexidade	PF	Observação

Guia de Contagem de Pontos de Função do MP

Contagem da Sprint 1	Incluir Empregado	EE	Baixa	3	
	Alterar Empregado	EE	Baixa	3	
	Empregado	ALI	Baixa	7	
Contagem da Sprint 2	Excluir Empregado	EE	Baixa	3	
	Empregado (alteração)	ALI	Baixa	3,5	Alteração caracterizada como evolução de requisitos. Aplicado o fator de impacto de 50%.
Contagem da Sprint 3	Consultar Empregado	CE	Baixa	3	
	Relatório XPTO	CE	Baixa	3	
	Incluir Empregado (alteração)	EE	Baixa	1,5	Alteração caracterizada como evolução de requisitos. Aplicado o fator de impacto de 50%.
Total de PF das Sprints				27	

	Nome da Função	Tipo	Complexidade	PF
Contagem da Release 1	Incluir Empregado	EE	Baixa	3
	Alterar Empregado	EE	Baixa	3
	Excluir Empregado	EE	Baixa	3
	Consultar Empregado	CE	Baixa	3
	Relatório XPTO	CE	Baixa	3
	Empregado	ALI	Baixa	7
	Total de PF da Release			

Os cálculos aplicados para este Cenário 2 são:

- Calcular o **Fator Evolutivo Ágil**, de acordo com a fórmula;

$$Fator\ Evolutivo\ Ágil = \left(\left(\frac{\sum PF\ Sprints}{PF\ Release} \right) - 1 \right) * 100$$

Fator Evolutivo Ágil = ((27 PFs / 22 PFs) - 1) * 100 = 22,72%

- Verificar se o resultado do Fator Evolutivo Ágil é maior ou igual a 30% (**Delta%**). No caso deste cenário, 22,72% é menor que 30% e, portanto, não haverá remuneração adicional ao total de pontos de função da *release*. A remuneração será referente à quantidade de 22 PFs (PF *Release*).

Observação: Cabe destacar que a diferença de 5 PFs, identificada entre o somatório de PF das *sprints* (no caso 27 PFs) e o tamanho da *release* (no caso 22 PF), deve ser absorvida pela contratada, pois de acordo com os conceitos utilizados neste guia, refere-se à evolução de requisitos, inerente ao processo de desenvolvimento ágil.

Cenário 2 – Cenário em que o volume das evoluções ultrapassa o **Delta%** definido com o valor de 30%.

Nesse caso, a contagem das *sprints* deve ser feita em planilha separada da contagem da *release*.

Release 1 (composta de 3 Sprints)	Nome da Função	Tipo	Complexidade	PF	Observação
Contagem da Sprint 1	Incluir Empregado	EE	Baixa	3	
	Alterar Empregado	EE	Baixa	3	
	Empregado	ALI	Baixa	7	
Contagem da Sprint 2	Excluir Empregado	EE	Baixa	3	
	Empregado (alteração)	ALI	Baixa	3,5	Alteração caracterizada como evolução de requisitos. Aplicado o fator de impacto de 50%.
Contagem da Sprint 3	Consultar Empregado	CE	Baixa	3	
	Relatório XPTO	CE	Baixa	3	
	Incluir Empregado (alteração)	EE	Baixa	1,5	Alteração caracterizada como evolução de requisitos. Aplicado o fator de impacto de 50%.
	Empregado (alteração)	ALI	Baixa	3,5	Alteração caracterizada como evolução de requisitos. Aplicado o fator de impacto de 50%.
Total de PF das Sprints				30,5	

	Nome da Função	Tipo	Complexidade	PF
Contagem da Release 1	Incluir Empregado	EE	Baixa	3
	Alterar Empregado	EE	Baixa	3
	Excluir Empregado	EE	Baixa	3
	Consultar Empregado	CE	Baixa	3
	Relatório XPTO	CE	Baixa	3
	Empregado	ALI	Baixa	7
Total de PF da Release				22

Os cálculos aplicados para este Cenário 2 são:

- Calcular o **Fator Evolutivo Ágil**, de acordo com a fórmula;

$$Fator\ Evolutivo\ \acute{A}gil = \left(\left(\frac{\sum PF\ Sprints}{PF\ Release} \right) - 1 \right) * 100$$

$$Fator\ Evolutivo\ \acute{A}gil = ((30,5\ PFs / 22\ PFs) - 1) * 100 = \mathbf{38,63\%}$$

- Verificar se o resultado do Fator Evolutivo Ágil é maior ou igual a 30% (Delta%). No caso deste cenário, 38,63% é maior que 30% e, portanto, deve ser calculado o quantitativo de pontos de função excedentes a fim de fazer a remuneração adicional. O cálculo se dá da seguinte forma:
- Calcular os pontos de função excedentes, aplicando a fórmula:

$$FP\ Excedente = \left(\sum PF\ Sprints - PF\ Release \right) - (PF\ Release * Delta\%)$$

$$PF\ Excedente = (30,5\ PFs - 22\ PFs) - (22\ PFs * 30 / 100) = \mathbf{1,9\ PFs}$$

A contratada será remunerada em 22 PF (**PF da Release**) + 1,9 PF (pontos de função excedentes ao **Delta%**). Neste caso, a remuneração da contratada será de 23,9 PF.

Diante disso, no caso de desenvolvimento com métodos ágeis, não serão remuneradas as evoluções inerentes ao processo ágil ocorridas durante a

execução da *release* até 30% (*DELTA*). Somente haverá remuneração do valor que exceder esse percentual.

5. Dicas para Otimizar o Custo das Manutenções

É fundamental aprimorar a gestão sobre as demandas de manutenção de sistemas. Se as demandas de manutenção em uma mesma funcionalidade forem executadas individualmente, a tendência é que o custo destas manutenções ao final sejam superiores ao que poderiam ser, caso houvesse um agrupamento de pequenas solicitações em uma única demanda à Fábrica de Software.

Seguem algumas dicas que podem ajudar a melhorar o cenário.

5.1. Consolidação de manutenções

Consolidar manutenções na mesma funcionalidade em uma única demanda é a maneira mais fácil de racionalizar o custo, pois fazer uma manutenção para atender um único requisito ou para atender a vários requisitos de manutenção na mesma funcionalidade terá o mesmo tamanho funcional quando solicitadas ao mesmo momento. Se solicitadas em momentos distintos, as mesmas funcionalidades serão pagas várias vezes, uma vez para cada um dos projetos de melhoria/manutenção.

No entanto, nem sempre é possível represar uma necessidade do usuário para que esta seja agrupada com outras, já que há demandas com prazos críticos. O importante é tentar avaliar ao máximo quais ajustes realmente são críticos e quais não são, visando minimizar o fracionamento das solicitações.

5.2. Análise crítica dos requisitos

Em muitas situações é possível ter uma única funcionalidade que faça o papel de duas existentes. Isto é muito comum no caso de consultas e relatórios com diferença apenas de alguns atributos apresentados. Ou seja, uma transação mais completa poderia ser elaborada para evitar a criação de várias funcionalidades distintas, porém semelhantes.

É mais barato pagar pela criação de uma funcionalidade nova do que por uma manutenção em duas ou mais funcionalidades, principalmente a longo prazo.

O mesmo acontece na definição dos requisitos quanto às regras de ordenação em relatórios e consultas. Sempre que possível, deve-se prever as possíveis opções de ordenação durante o desenvolvimento da funcionalidade, pois isso reduzirá a necessidade de manutenção para atender a novos critérios de ordenação.

5.3. Programas Auxiliares

Para evitar o custo de realização de sucessivos Programas Auxiliares, deve-se ponderar o custo/benefício de se elaborar a criação de novas funcionalidades para manter dados que precisam ser atualizados com uma certa frequência.

6. Entrega da contagem

Todas as contagens apresentadas ao MP deverão ser feitas utilizando o modelo de planilha adotado pelo órgão.

Para o preenchimento da planilha, devem-se seguir as regras estabelecidas pelo MP.

7. Glossário

Para a uniformização do entendimento do conteúdo deste guia, serão listadas definições e premissas utilizadas pelo MP na contagem dos pontos de função:

- **Análise de Pontos de Função (APF)**
Método para a medição de tamanho funcional de um software. A técnica mede as funcionalidades de um software sob o ponto de vista do usuário.
- **Counting Practices Manual (CPM)**
Manual de Práticas de Contagem de pontos de função, mantido pelo IFPUG.
- **International Function Point Users Group (IFPUG)**
Grupo Internacional de Usuários de Pontos de Função.
- **Processo Elementar (PE)**
É a menor unidade de atividade significativa para o usuário. Deve ser completo em si mesmo, independente, e deixar o negócio da aplicação em estado consistente.
- **Lógica de Processamento**
O CPM 4.3 define lógica de processamento como requisitos especificamente solicitados pelo usuário para completar um processo elementar. Esses requisitos devem incluir as seguintes ações:
 - Validações são executadas;
 - Fórmulas matemáticas e cálculos são executados;
 - Valores equivalentes são convertidos;
 - Dados são filtrados e selecionados através da utilização de critérios;
 - Condições são analisadas para verificar quais são aplicáveis;
 - Um ou mais ALIs são atualizados;
 - Um ou mais ALIs e AIEs são referenciados;
 - Dados ou informações de controle são recuperados;

- Dados derivados são criados através da transformação de dados existentes, para criar dados adicionais;
 - O comportamento do sistema é alterado;
 - Preparar e apresentar informações para fora da fronteira;
 - Receber dados ou informações de controle que entram pela fronteira da aplicação;
 - Dados são reordenados.
 - A determinação do tipo e da unicidade de um processo elementar ocorre pela lógica de processamento vinculada a tal processo, e cada processo elementar pode incluir múltiplas alternativas ou ocorrências das ações acima.
- Registro Lógico Referenciado (RLR ou TR)
Um tipo de registro lógico referenciado é um subgrupo de dados reconhecido pelo usuário dentro de uma função de dados (ALI ou AIE).
 - Arquivo Lógico Referenciado (ALR ou AR)
Função de dados lida e/ou mantida por uma função transacional.
 - Dado Elementar Referenciado (DER ou TD)
Atributo único, reconhecido pelo usuário e não repetido.

8. Processo de Revisão do Guia de Contagem

8.1. Revisão para Correção de Inconsistências e Situações Não Previstas

As adequações a este guia poderão ser realizadas pelo Ministério do Planejamento, Orçamento e Gestão para atender normas vigentes, situações não previstas, recomendações de órgão de controle, bem como ajuste de texto para evitar duplicidade no entendimento.

8.2. Revisão para Adoção de Novas Versões do CPM

Com o lançamento de novas versões do manual (CPM), sua adoção como referência para este guia de contagem deverá ser acordada entre o MP e a contratada.

9. Referências Bibliográficas

1. Manual de Práticas de Contagem de Pontos de Função, Versão 4.3.1, International Function Point Users Group (IFPUG), janeiro/2010;
2. Roteiro de Métricas de Software do SISP, Versão 2.0, disponível em http://www.sisp.gov.br/ct-gcie/download/file/Roteiro_de_Metricas_de_Software_do_SISP_-_v2.0.pdf. Acessado em: novembro de 2014;
3. Vazquez, C. E. ; Simões, G. S. ; Albert, R. M. Análise de Pontos de Função, Medição, Estimativas e Gerenciamento de Projetos de Software. 10.ed. São Paulo: Editora Érica, 2010;

4. Análise de pontos de função para melhoria de software, versão 2.2.1, Associação Holandesa dos Usuários de Métricas de Software (NESMA);
5. <http://fattocs.com/pt/blog-pt>, site acessado em novembro e dezembro de 2014;
6. Artigo sobre Diferentes aplicações, diferentes visões dos dados, elaborado por Carlos Eduardo Vazquez da FATTO Consultoria e Sistemas.
7. Encarte IX – Guia de Contagem e Estimativa, Lote 02, Pregão Eletrônico SRP Nº 28/2012-MME;
8. Guia de Contagem de Pontos de Função do INEP, versão 7.3, 2012;
9. Guia de Contagem de Pontos de Função STN – SERPRO, versão 2.0, 2013;
10. Guia de Contagem de Pontos de Função da Infraero, versão 6.0.1, 2014;
11. Guia de Contagem do PROCERGS, versão 2.0, 2013;
12. Guia de Contagem do Ministério da Justiça, versão 2.0, 2013.
13. <http://fattocs.com/pt/contagem-antecipada>, site acessado em fevereiro de 2015.