

Roteiro SERPRO de Contagem de Pontos de Função e Estimativas

Histórico de Versões

Data	Versão	Descrição	Autor	Revisor	Aprovado por
15/04/2010	1.0	Roteiro Corporativo de Contagem de Pontos de Função.	Claudia Hazan		
07/05/2010	2.0	Atualização na definição de Manutenção Cosmética Atualização Tabela Distribuição de Esforço	Claudia Hazan		
24/05/2010	3.0	Atualização estimativa de equipe alocada.	Claudia Hazan		
25/05/2010	3.1	Manutenção Adaptativa - Iterações de textos em mensagens de erro, alerta, aviso e validação.	Claudia Hazan		
01/12/2010	3.2	Correção das Tabelas de Esforço por Fase	Claudia Hazan		
01/04/2011	4.0	Conteúdo do Roteiro SISP	Claudia Hazan	Eduardo Oliveira Giovana Freitas	
02/07/2012	5.0	Considerações Roteiro RFB, STN e SISP Atualização da Tabela de Produtividade por Linguagem	Claudia Hazan		
18/07/2012	5.1	Projetos de Melhoria – Funções Alteradas; Contagem PF Retrabalho; Atualização de Dados de Domínio; PF testes; Massa de Testes para Homologação	Claudia Hazan		
23/08/2012	5.2	. Inclusão do Projeto Teste Integrado. Inclusão PF Roteiro Homologação. Atualização de Texto para compatibilidade com Roteiro RFB	Claudia Hazan		
08/01/2013	5.3	Atualização texto de componentes	Claudia Hazan		
24/06/2013	5.4	Atualização Produtividade – MOBILE	Claudia Hazan		
17/12/2013	5.5	Atualização Linguagens MOBILE, JCUPIIM, LIFERAY, RUBY ON RAILS	Claudia Hazan		
28/2/2014	6.0	Atualizações propostas pelo GT Estimativas – PSDS e Roteiro SISP 2.0	Claudia Hazan		
10/3/2014	6.1	Conversão Formato Br Office	Claudia Hazan		
25/03/14	6.2	Pequenas alterações de texto para esclarecer que Td representa o prazo em meses e que a fórmula de Capers Jones deve ser utilizada para projetos a partir de 100 PF.	Neila Azevedo		
04/04/14	6.3	Alteração do Roteiro para que existam as 5 linguagens Mobile a seguir: Mobile – Android, Mobile – iOS, Mobile - Android e iOS, Mobile - HTML 5 e JQuery Mobile e Mobile – PhoneGAP.	Neila Azevedo		

12/05/2014	6.4	Correção Fórmula DW – item 8.11. Atualização texto PF Help, considerando Manual de usuário, orientações PF Componente, Orientações Retrabalho RFB na Homologação, Atualização texto Múltiplas Mídias, Atualização texto Manutenção Adaptativa	Claudia Hazan, Bruno Aroxa, Luciano Buzzacaro		
05/08/14	6.5	Alteração dos itens “3.2 Estimativa de Esforço de Projetos de Software”, “3.2.1 Distribuição de Esforço por Fase do Projeto” e “6.1 Considerações sobre Mudanças de Requisitos”, ajustando texto e retirando as tabelas 6 e 11 (Distribuição de Esforço por Macroatividades do projeto) e adaptando o exemplo de cálculo do retrabalho para utilizar a tabela 12 (Distribuição de esforço do Projeto conforme Roteiro SISP). A tabela 7 foi renomeada para 6, tabela 8 foi renomeada para 7, a tabela 12 foi renomeada para Tabela 10 e a Tabela 13 foi renomeada para Tabela 11. Atualização da Tabela de Produtividade por Linguagem conforme levantamento realizado em julho/2014.	Neila Azevedo		
19/09/2014	6.6	Alterações realizadas: 1) Inclusão da linguagem “COMPONENTE – CÓDIGO ABERTO” na tabela 5 de produtividade. Inclusão da informação de quais linguagens são consideradas Código Aberto. Mudança dos nomes de linguagens para DW PENTAHO (Extração e OLAP), DW PENTAHO (Apenas OLAP), Java Web não Distribuída, COMPONENTE – CÓDIGO PROPRIETÁRIO, DW OUTRAS (apenas OLAP) e DW OUTRAS (Extração e OLAP). 2) Alteração do item “4.17 Pontos de Função de Massa de Teste para Homologação” para inclusão do último parágrafo. 3) Alteração do item “8.14 Massa de Dados para Homologação em DW” para inclusão do último parágrafo.	Neila Azevedo		
19/09/2014	6.7	Alteração do item “4.5 Redesenolvimento de Projetos em outra Plataforma” para ficar igual ao Roteiro da RFB v4.9, alterando a fórmula $PF_REDESENVOLVIMENTO_BD_RELACIONAL = (PF_ALTERADO \times 0,30) + PF_CONVERSÃO$ para $PF_REDESENVOLVIMENTO_BD_RELACIONAL = (PF_ALTERADO \times 0,50) + PF_CONVERSÃO$	Neila Azevedo		

27/05/2015	6.8	Alteração do percentual de PF Corretiva de 50% para 75% no item 4.4 Manutenção Corretiva.	Neila Azevedo	Rosana Disconzi e Nádía Costa	
12/06/2015	6.9	Inclusão das linguagens DATA DISCOVERY, PROJETOS DE GEORREFERENCIAMENTO, MIDDLEWARE e WEBSERVICE na Tabela 6: Tabela de Produtividade por Linguagem e Tipos de Projetos. Reorganização da numeração das tabelas.	Neila Azevedo		
22/06/2015	7.0	Inclusão das linguagens CKAN e Joomla na "Tabela 6: Tabela de Produtividade por Linguagem e Tipos de Projetos". Inclusão do Tópico "4.22-Projeto de Automação de Testes". Alteração do item "4.15 Pontos de Função de Testes não Funcionais" para incluir exemplos para testes de portabilidade. Inclusão do último parágrafo no item "4.21 Desenvolvimento e Manutenção de Componentes Internos Reusáveis" sobre serviços de monitoramento. Alteração do item "4.16 Projeto de Teste Integrado" para retirada do texto "Trata-se de uma demanda de teste em requisitos não funcionais."	Neila Azevedo	Luciano Buzzacaro, Rosana Disconzi, Nádía Costa, Edilson Santos.	

Sumário

1. INTRODUÇÃO	5
2. OBJETIVO	6
3. ESTIMATIVAS DE PROJETOS DE SOFTWARE	7
3.1 CONTAGEM ESTIMATIVA DE PONTOS DE FUNÇÃO (CEPF)	9
3.2 ESTIMATIVA DE ESFORÇO DE PROJETOS DE SOFTWARE	15
<i>3.2.1 Distribuição de Esforço por Fase do Projeto</i>	<i>19</i>
3.3 ESTIMATIVA DE PRAZO DE PROJETOS DE SOFTWARE	21
<i>3.3.1 Alocação de Equipe ao Projeto</i>	<i>23</i>
3.4. MÉTODO PARA ESTIMATIVA DE CUSTO	24
3.5 ESTIMATIVA DE RECURSOS COMPUTACIONAIS	25
4. CONTAGEM DE PONTOS DE FUNÇÃO DE PROJETOS	26
4.1 PROJETO DE DESENVOLVIMENTO	26
4.2 PROJETO DE MIGRAÇÃO DE DADOS	27
4.3 PROJETO DE MELHORIA	28
4.4 MANUTENÇÃO CORRETIVA	31
4.5 REDESENVOLVIMENTO DE PROJETOS EM OUTRA PLATAFORMA	32
4.6 ATUALIZAÇÃO DE PLATAFORMA	33
4.7 MANUTENÇÃO EM INTERFACE	33
4.8 MANUTENÇÃO ADAPTATIVA EM REQUISITOS NÃO FUNCIONAIS	34
4.9 APURAÇÃO ESPECIAL	36
<i>4.9.1. Apuração Especial – Base de Dados</i>	<i>37</i>
<i>4.9.2. Apuração Especial – Geração de Relatórios</i>	<i>38</i>
<i>4.9.3. Apuração Especial – Reexecução</i>	<i>38</i>
4.10 ATUALIZAÇÃO DE DADOS E PF DOMÍNIO	39
4.11 MANUTENÇÃO EM PÁGINAS ESTÁTICAS DE INTRANET, INTERNET OU PORTAL	40
4.12 MANUTENÇÃO DE DOCUMENTAÇÃO DE SISTEMAS LEGADOS	40
4.13 VERIFICAÇÃO DE ERROS	41
4.14 PONTOS DE FUNÇÃO DE TESTES FUNCIONAIS	41
4.15 PONTOS DE FUNÇÃO DE TESTES NÃO FUNCIONAIS	44
4.16 PROJETO DE TESTE INTEGRADO	45
4.17 PONTOS DE FUNÇÃO DE MASSA DE TESTE PARA HOMOLOGAÇÃO	46
4.18 ROTEIRO PARA HOMOLOGAÇÃO	46
4.19 GERENCIAMENTO DE RISCOS DE SEGURANÇA	47
4.20 DESENVOLVIMENTO E MANUTENÇÃO DE HELP	47
4.21 DESENVOLVIMENTO E MANUTENÇÃO DE COMPONENTES INTERNOS REUSÁVEIS	48
4.22 PROJETO DE AUTOMAÇÃO DE TESTES	50
5. ATIVIDADES SEM CONTAGEM DE PONTOS DE FUNÇÃO	50
5.1 ATIVIDADES DE PROJETOS DE DW SEM CONTAGEM DE PONTOS DE FUNÇÃO	54
6. CONSIDERAÇÕES PARA CONTAGEM DE PONTOS DE FUNÇÃO	56
6.1 CONSIDERAÇÕES SOBRE MUDANÇA DE REQUISITOS	56
6.2 CONSIDERAÇÕES SOBRE MUDANÇAS DE REQUISITOS NA FASE DE REQUISITOS	59
6.3 CONSIDERAÇÕES SOBRE MUDANÇAS DE REQUISITOS NA FASE DE HOMOLOGAÇÃO -CLIENTE RFB	60
6.4 CONSIDERAÇÕES SOBRE PROJETOS CANCELADOS	60
6.5 CONSIDERAÇÕES SOBRE REDUÇÃO DE CRONOGRAMA	60
6.6 FATOR DE CRITICIDADE DE SOLICITAÇÃO DE SERVIÇO	61
7. CONTAGEM DE PONTOS DE FUNÇÃO COM MÚLTIPLAS MÍDIAS E DADOS DE CÓDIGO	62
7.1 CENÁRIO 1: MESMOS DADOS APRESENTADOS EM TELA E IMPRESSOS	63

7.2	CENÁRIO 2: MESMOS DADOS DE SAÍDA COMO DADOS EM ARQUIVO E RELATÓRIO IMPRESSO	64
7.3	CENÁRIO 3: MESMOS DADOS DE ENTRADA BATCH E ON-LINE	64
7.4	CENÁRIO 4: MÚLTIPLOS CANAIS DE ENTREGA DA MESMA FUNCIONALIDADE	65
7.5	CENÁRIO 5: RELATÓRIOS EM MÚLTIPLOS FORMATOS	65
7.6	CENÁRIO 6: FUNCIONALIDADES FORNECIDAS VIA APLICAÇÃO E WEBSERVICE	65
7.7	CENÁRIO 7: PLATAFORMA MOBILE – ANDROID E IOS	66
7.8	DIMENSIONAMENTO DE DADOS DE CÓDIGO	66
8	GUIA DE CONTAGEM DE PONTOS DE FUNÇÃO PARA PROJETOS DE DATA WAREHOUSE	68
8.1	DEFINIÇÃO DE TIPO DE CONTAGEM	68
8.2	IDENTIFICAÇÃO DE PROPÓSITO, ESCOPO E FRONTEIRA	68
8.3	IDENTIFICAÇÃO DE PROCESSOS ELEMENTARES	70
8.4	ENTRADAS EXTERNAS EM PROJETOS DE DATA WAREHOUSE	70
8.5	CONSULTAS EM PROJETOS DE DATA WAREHOUSE	71
8.6	IDENTIFICAÇÃO DE FUNÇÕES DE DADOS EM PROJETOS DE DATAWAREHOUSE	72
8.6.1	<i>Conceito de Mudança Estrutural em ALI/AIE para projetos de Melhoria</i>	74
8.7	TABELAS DE VISUALIZAÇÃO – GERAÇÃO DE CUBOS OU CONTEXTO DE ANÁLISE OU UNIVERSO	75
8.8	FUNCIONALIDADES DE CONTROLE DO DATA WAREHOUSE	75
8.9	DEMANDAS TÍPICAS DE MANUTENÇÃO EVOLUTIVA EM DW	76
8.9.1	<i>Criação de Métricas (fórmulas)</i>	76
8.9.2	<i>Alteração de Campos em tabelas Fato e Dimensão</i>	76
8.9.3	<i>Criação, Configuração e Disponibilização de um Filtro</i>	77
8.10	ALTERAÇÃO DE DADOS DE DIMENSÕES ESTÁTICAS	77
8.11	CONTAGEM DE METADADOS: DESCRIÇÃO DE ATRIBUTOS, MÉTRICAS E PASTAS	77
8.12	REORGANIZAÇÃO DA BANCADA (REPOSICIONAMENTO DE ITENS)	78
8.13	EVOLUÇÃO DE PÁGINAS ESTÁTICAS EM DATA WAREHOUSE	78
8.14	MASSA DE DADOS PARA HOMOLOGAÇÃO EM DW	78
8.15	CONSIDERAÇÕES SOBRE ESTIMATIVAS DE DATA WAREHOUSE	79
9	CONTAGEM DE PONTOS DE FUNÇÃO DE PORTAIS ZOPE/PLONE	80
10	ORIENTAÇÕES PARA SISTEMAS RFB	82
10.1	DEFINIÇÃO FRONTEIRAS DE SISTEMAS RFB	84
11	CONCLUSÃO	85
	REFERÊNCIAS BIBLIOGRÁFICAS	86

1. Introdução

O SERPRO tem utilizado a métrica de Pontos de Função (PF) nas estimativas e dimensionamento de tamanho funcional de projetos de software, devido aos diversos benefícios de utilização da métrica, tais como: possibilitar as estimativas de prazo, esforço e equipe alocada nas fases iniciais do processo de software; apoiar a gestão do desenvolvimento sendo um dado padrão para a aferição de indicadores de produtividade. Além disso, cabe ressaltar o uso métrica nos contratos com os clientes em aderência às recomendações dos Acórdãos do Tribunal de Contas da União (TCU).

O Manual de Práticas de Contagem de Pontos de Função (CPM 4.3) [IFPUG, 2010], publicado pelo *International Function Point Users Group* (IFPUG), define as regras de contagem de Pontos de Função. É importante ressaltar que a métrica de Pontos de Função foi concebida como uma medida de tamanho funcional para projetos de desenvolvimento e de melhoria (manutenção evolutiva) de software. No entanto, os projetos de software não estão limitados a projetos de desenvolvimento e de melhoria. Assim, torna-se essencial a definição de métricas para dimensionar o tamanho de projetos de manutenção de uma maneira objetiva para que estes projetos possam ser gerenciados e faturados com base em uma métrica.

Além disso, a contagem de Pontos de Função é baseada no projeto lógico da aplicação (logical design) e nas fases iniciais do ciclo de vida do software, o documento de requisitos para a estimativa e elaboração do plano do projeto é um documento inicial de requisitos, por exemplo: Documento de Visão, Formalização Simples de Requisitos, Ata de Reunião ou algum outro tipo de especificação inicial. Assim, torna-se importante o estabelecimento de métodos para estimar o tamanho funcional dos projetos de software nas fases iniciais do ciclo de vida. Outro ponto a ser destacado é a importância da definição de métodos para geração de estimativas de prazo, esforço, equipe alocada, preço e recursos computacionais dos projetos de software da empresa, visando melhorar o gerenciamento dos projetos.

É importante ressaltar o Manual de Práticas de Contagem (CPM) é um documento que se destina a mensurar o tamanho funcional de projetos de software, não tendo por objetivo principal suportar contratos de fábrica de software. Desta forma, torna-se necessário a criação de guias de contagem complementares.

2. Objetivo

Este documento tem como propósito apresentar um roteiro de Contagem de Pontos de Função aderente ao Manual de Práticas de Contagem (CPM 4.3) e ao Roteiro de Métricas do SISP. O Roteiro de Métricas do SERPRO tem como objetivo definir os tipos de projetos de manutenção e uma sistemática para dimensionar o tamanho de tais projetos, com base na métrica Pontos de Função. Além da contagem de Pontos de Função, este roteiro apresenta um processo de aderente ao modelo CMMI.

A métrica considerada nesse Roteiro é a de **Pontos de Função Não Ajustados**.

Este documento encontra-se organizado da seguinte maneira: O Capítulo 1 apresenta a motivação para a elaboração do documento; O Capítulo 2 mostra os objetivos aos quais este documento se propõe e a organização deste documento; O Capítulo 3 define o processo de estimativas de projetos de software integrado ao processo, indicando o momento de realização das estimativas e as análises a serem realizadas; O Capítulo 4 apresenta diretrizes para a estimativa e a contagem de Pontos de Função de todos os tipos de projetos de manutenção de software; O Capítulo 5 descreve as atividades associadas ao processo de desenvolvimento de software sem contagem de Pontos de Função; O Capítulo 6 apresenta algumas considerações importantes sobre utilização de métricas para dimensionar as mudanças de requisitos e redução de cronograma; O Capítulo 7 define um Guia para contagem de Múltiplas Mídias; O Capítulo 8 define um Guia para Contagem de Projetos de Data Warehouse; O Capítulo 9 apresenta diretrizes para contagem de Pontos de Função de Portais utilizando a plataforma Zope/Plone; O Capítulo 10 apresenta orientações para contagem e documentação de contagem de Pontos de Função; Finalmente, o Capítulo 11 conclui o documento apresentando sugestões para trabalhos futuros.

3. Estimativas de Projetos de Software

Este Capítulo tem como propósito descrever um processo de estimativas de projetos de software aderente ao CMMI. Nesse contexto, são apresentados: o método Contagem Estimativa de Pontos de Função (CEPF) para estimar o tamanho dos projetos de software em PF, o modelo simplificado de estimativas para estimar o esforço dos projetos em homem_hora (HH), a fórmula de Capers Jones para estimar os prazos dos projetos.

A **Figura 1** ilustra um processo de Estimativas de Projetos de Software aderente à área de processo de Planejamento do Projeto do nível 2 do CMMI. Este processo é descrito em linhas gerais nos parágrafos seguintes.

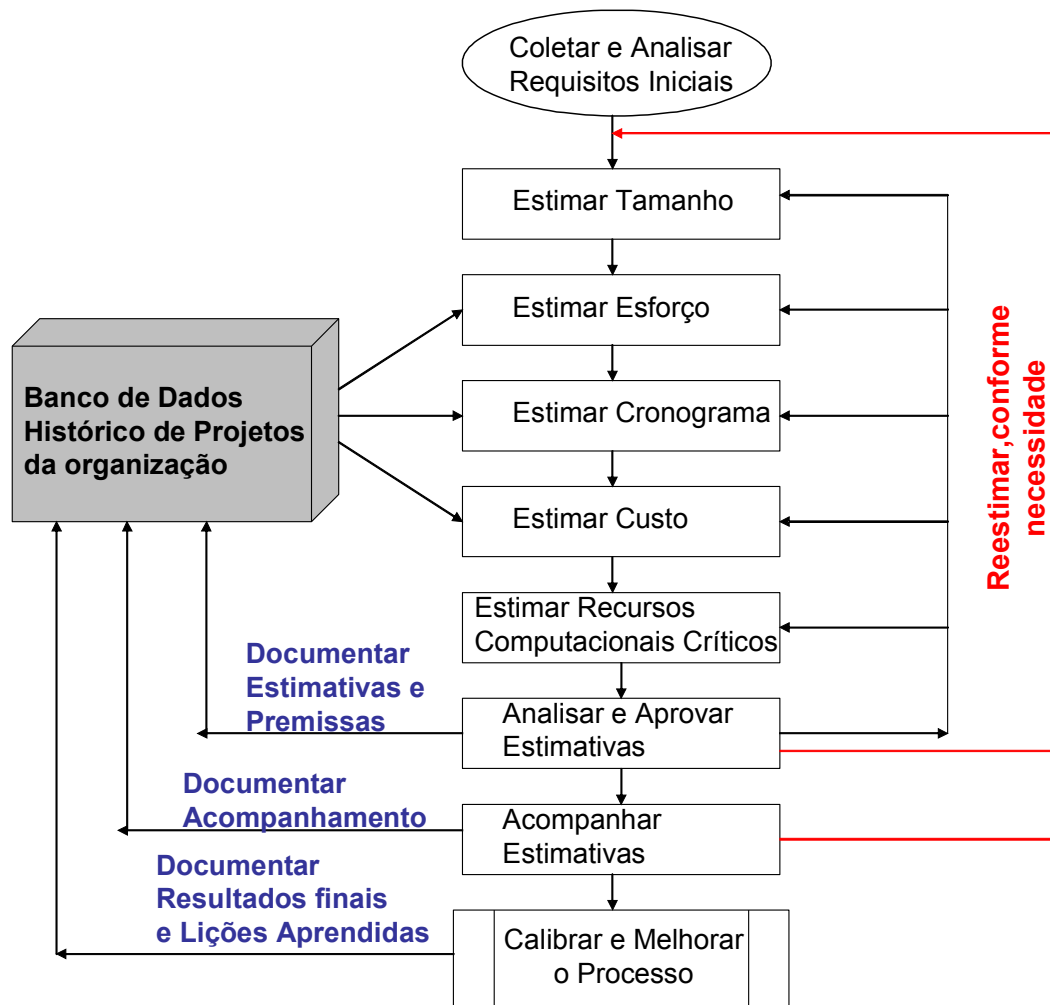


Figura 1: Processo de Estimativas de Projetos de Software [Hazan, 2008]

O principal insumo (artefato de entrada) para um processo de estimativas é o documento de requisitos. Como as estimativas devem ser realizadas no início do processo de desenvolvimento de software, então, o artefato utilizado é um documento inicial de requisitos, por exemplo: Documento de Visão, Ata de Reunião. O estimador deve analisar os requisitos para garantir a qualidade e então estimar o tamanho do projeto de software. O próximo passo é a derivação das estimativas de esforço, prazo (cronograma), custo (orçamento) com base na estimativa de tamanho e nos dados históricos de projetos concluídos da empresa, assim como o estabelecimento da estimativa de recursos computacionais críticos e dos recursos da equipe a ser alocada ao projeto. Neste ponto, as principais estimativas foram geradas e precisam ser documentadas. As premissas e suposições utilizadas na geração das estimativas, dentre outras: complexidade do projeto, plataforma de desenvolvimento, tipo do projeto, percentual de evolução de requisitos (Scope Creep), também devem ser documentadas [Hazan, 2008].

Nessa etapa é importante destacar os seguintes conceitos na área de estimativas: Uma **Estimativa** é obtida por meio de uma atividade técnica, utilizando métodos de estimativas. Não deve sofrer interferências políticas. A **Meta** é um desejo, em função de necessidades de negócio, estabelecida politicamente. Um **Compromisso** é um acordo da gerência com as equipes técnicas para alcançar uma meta [Parthasarathy,2007]. Em um cenário ideal os resultados da estimativa atendem as metas de negócio. Quando este cenário não é real, é fundamental a redução de escopo do projeto, de modo que a meta se adapte aos resultados da estimativa.

A realização das estimativas por um analista de métricas que não atue na equipe do projeto constitui uma prática recomendada. O analista de métricas deve analisar também a consistência da documentação utilizada na estimativa. No decorrer do processo de desenvolvimento, as estimativas devem ser acompanhadas conforme o refinamento dos requisitos. O projeto deve ser reestimado após a fase de requisitos, quando for gerada a especificação de casos de uso, e sempre ocorrerem mudanças significativas nos requisitos funcionais ou não funcionais. Quando o projeto é concluído, deve-se aferir e documentar o tamanho, prazo, custo, esforço e recursos realizados, assim como outros atributos relevantes do projeto, visando a coleta de dados para a melhoria do processo de estimativas. As lições aprendidas também devem ser documentadas [Hazan, 2008].

Portanto, as estimativas e contagens de Pontos de Função devem ser realizadas nos seguintes marcos do processo de desenvolvimento de software, a saber:

- **Estimativa inicial:** realizada após o fechamento do escopo do projeto. Geralmente, é baseada em um documento inicial de requisitos, por exemplo Documento de Visão. Constitui uma boa prática a previsão de evolução de requisitos, especialmente em projetos de desenvolvimento de médio ou grande porte. Recomenda-se a utilização do percentual de 35% para evolução de requisitos.
- **Contagem de Pontos de Função de Referência:** realizada após o aceite dos requisitos. Geralmente, leva em consideração a Especificação dos Casos de Uso e Regras de Negócio da aplicação.
- **Contagem de Pontos de Função Final:** realizada após a homologação da aplicação. Esta contagem leva em consideração as funcionalidades efetivamente entregues para o usuário pela aplicação.
- **Contagem Pontos de Função Retrabalho:** realizada sempre ocorrer mudança de requisitos em qualquer fase do processo de desenvolvimento. Esta contagem leva em consideração o Relatório de Análise de Impacto. Caso as mudanças sejam significativas, maiores que a evolução de requisitos (scope creep) prevista na estimativa inicial, o prazo do projeto deve ser reestimado. Toda mudança de requisito deve passar por uma análise de impacto do SERPRO e ser aprovada pelo cliente.

Para fins de faturamento e aferição dos indicadores de produtividade do projeto, deve-se considerar a Contagem de Pontos de Função Final e as Contagens de Pontos de Função de Retrabalho.

As seções seguintes apresentam os métodos de estimativas de tamanho prazo, custo e esforço utilizados nos projetos de software do SERPRO.

3.1 Contagem Estimativa de Pontos de Função (CEPF)

Antes de definir o método de estimativas – Contagem Estimativa de Pontos de Função (CEPF), é importante destacar que “estimar significa utilizar o mínimo de tempo e esforço para se obter um valor aproximado dos Pontos de Função do projeto de software investigado” [Meli, 1999]. Desta forma, é recomendável sempre fazer uma distinção entre

os termos e conceitos: Contagem de Pontos de Função e Estimativa de Pontos de Função.

- **Contagem de Pontos de Função:** significa medir o tamanho do software por meio do uso das regras de contagem do IFPUG [IFPUG, 2010];
- **Estimativa de Pontos de Função:** significa fornecer uma avaliação aproximada do tamanho de um software utilizando métodos diferentes da Contagem de Pontos de Função do IFPUG.

O método CEPF visa aferir o tamanho em PF de maneira simplificada, com base no conhecimento dos requisitos iniciais do projeto. Inicialmente, os requisitos funcionais iniciais documentados nas propostas comerciais, nos documentos de visão, formalização simples de requisitos ou em qualquer especificação inicial do sistema do usuário são mapeados nos tipos funcionais da Análise de Pontos de Função: Arquivo Lógico Interno (ALI), Arquivo de Interface Externa (AIE), Entrada Externa (EE), Consulta Externa (CE) e Saída Externa (SE) (Figura 2). Posteriormente, os Pontos de Função são associados a cada função identificada, baseando-se nas tabelas de complexidade e de contribuição funcional do CPM.

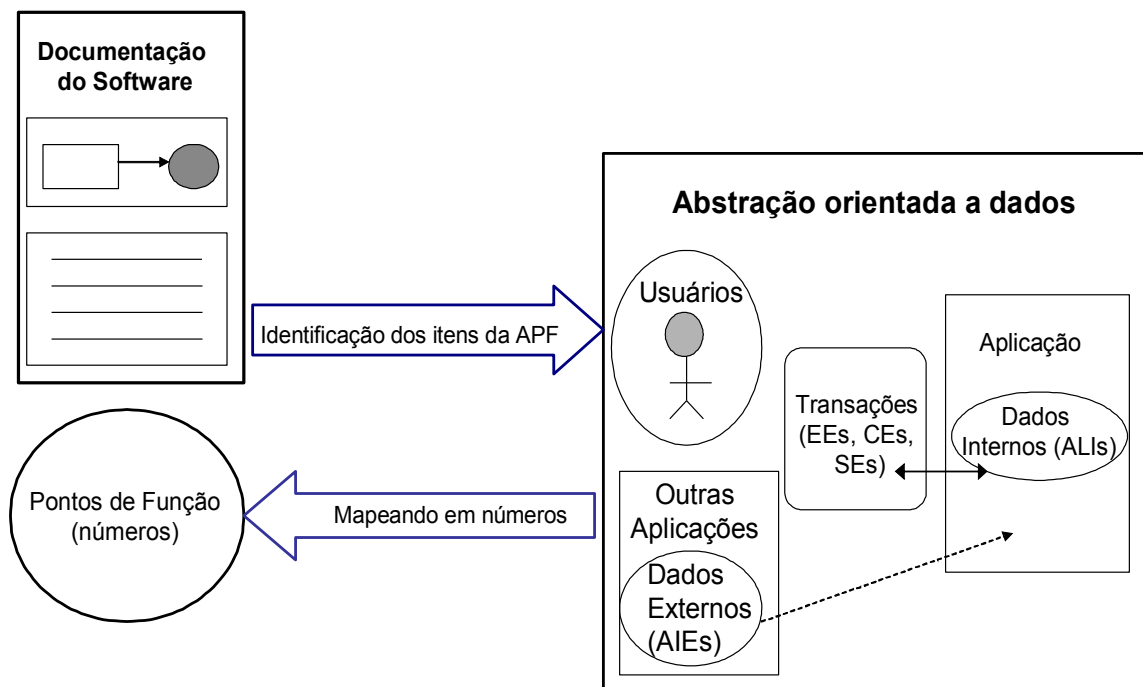


Figura 2: Modelo Lógico da Análise de Pontos de Função

O estimador deve realizar uma leitura no documento inicial de requisitos, buscando informações relevantes para a identificação de processos elementares. O processo elementar é definido como a menor unidade de atividade significativa para o usuário. O processo elementar deve ser uma transação completa em si mesma, independente e deixar a aplicação em um estado consistente [IFPUG, 2010]. Em outras palavras, os processos elementares são funções transacionais independentes, isto é, funções sequenciais pertencem a um mesmo processo elementar e funções independentes constituem processos elementares diferentes.

Uma vez identificado o processo elementar, o estimador deve buscar o entendimento deste para classificá-lo em Entrada Externa, Consulta Externa ou Saída Externa. Adicionalmente, o estimador deve descobrir os dados associados ao processo elementar, visando a determinação da complexidade funcional da função identificada. Caso não seja possível a identificação da complexidade da funcionalidade em questão, recomenda-se a utilização da complexidade Média. Na análise do processo elementar também são identificados, os grupos de dados lógicos da aplicação, que são classificados como Arquivos Lógicos Internos ou Arquivos de Interface Externa. Caso não seja possível a identificação da complexidade da função de dados em questão, recomenda-se a utilização da complexidade Baixa. É importante ressaltar que se o estimador identificar mais de um Registro Lógico no Arquivo Lógico Interno, recomenda-se utilizar a complexidade Média.

A seguir são apresentadas dicas para ajudar no mapeamento dos requisitos funcionais da aplicação nos tipos funcionais da APF. As necessidades e funcionalidades especificadas para o projeto, contidas no documento inicial de requisitos, devem ser enquadradas em uma das seguintes tabelas:

Tabela 1 - Contagem dos Arquivos Lógicos Internos (ALIs): Banco de Dados Lógico da Aplicação (tabelas e arquivos mantidos pela aplicação).

Considerações: Identifique os grupos de dados lógicos de aplicação nos modelos de dados ou diagrama de classes ou a partir dos requisitos funcionais, descritos nos documentos de requisitos (Documento de Visão, Relação de Casos de Uso, etc.). Não considere arquivos físicos, arquivos de índices, arquivos de trabalho e tabelas de relacionamento sem atributos próprios (tabelas que existem para quebrar o relacionamento nxm e apenas transportam as chaves estrangeiras). As entidades fracas

também não são consideradas um ALI. Se possível, tente descobrir os atributos lógicos, campos reconhecidos pelo usuário, e subgrupos de dados existentes para obter a complexidade funcional, segundo as regras de contagem do CPM. Caso não seja possível, a experiência tem mostrado que a maioria dos ALIs dos sistemas são de complexidade **Baixa**.

Nº ALIs Baixa:	X 7 PF
Nº ALIs Média:	X 10 PF
Nº ALIs Alta:	X 15 PF
Total PF da Tabela 1:	

Tabela 1: Identificação dos Arquivos Lógicos Internos da Aplicação

Tabela 2 - Contagem de Arquivos de Interface Externa (AIEs): Banco de Dados de outras Aplicações, **apenas referenciados** pela aplicação que está sendo estimada (tabelas e arquivos mantidos por outra aplicação).

Considerações: Identifique os grupos de dados lógicos de outras aplicações referenciados pela aplicação que está sendo estimada. Frequentemente, o referenciamento de dados ocorre para a validação de informações em cadastros ou consultas. Algumas vezes, relatórios ou consultas referenciam dados externos de outras aplicações, também considerados AIEs. Não são considerados arquivos físicos, arquivos de índice, arquivos de trabalho, tabelas de relacionamento sem atributos próprios e entidades fracas. Geralmente, os AIEs dos sistemas possuem a classificação de complexidade **Baixa**. Porque, são considerados para a determinação da complexidade funcional do AIE apenas os atributos referenciados pela aplicação que está sendo contada.

Nº AIEs Baixa:	X 5 PF
Nº AIEs Média:	X 7PF
Nº AIEs Alta:	X 10 PF
Total PF da Tabela 2:	

Tabela 2: Identificação dos Arquivos de Interface Externa da Aplicação

Tabela 3 - Contagem de Entradas Externas (EEs): Funcionalidades que mantêm os Arquivos Lógicos Internos (ALIs) ou alteram o comportamento da aplicação.

Considerações: Identifique as funcionalidades de manutenção de dados. Conte separadamente a inclusão, alteração e exclusão de dados, isto é, cada função independente de inclusão ou alteração ou exclusão deve ser contada separadamente. A aplicação possui funções de entrada de dados que alteram o comportamento dela, por exemplo: processamentos *batch*, ou processamento de informações de controle? Caso positivo, estas funções também devem ser identificadas como Entradas Externas. Se você não possui conhecimento sobre o processo elementar (funcionalidade analisada), considere a Entrada Externa identificada com complexidade **Média**.

Nº EEs Baixa:	X 3 PF
Nº EEs Média:	X 4 PF
Nº EEs Alta:	X 6 PF
Total PF da Tabela 3:	

Tabela 3: Identificação das Entradas Externas da Aplicação

Tabela 4 - Contagem de Consultas Externas (CEs): funcionalidades que apresentam informações para o usuário **sem** a utilização de cálculos ou algoritmos. São os processos elementares do tipo “lê - imprime”, “lê - apresenta dados”, incluindo consultas, relatórios, geração de arquivos pdf, xls, *downloads*, entre outros.

Considerações: Você está desenvolvendo uma função para apresentar informações para o usuário: uma consulta, relatório, *browse*, *listbox*, *download*, geração de um arquivo, geração de arquivo pdf, xls? Esta função **não** possui cálculos ou algoritmos para derivação dos dados referenciados nem altera um Arquivo Lógico Interno, nem muda o comportamento do sistema? Caso positivo, estas funções devem ser identificadas como Consultas Externas. Se você não possui conhecimento sobre o processo elementar (funcionalidade analisada), considere as Consultas Externas com complexidade **Média**.

Nº CEs Baixa:	X 3 PF
Nº CEs Média:	X 4 PF
Nº CEs Alta:	X 6 PF
Total PF da Tabela 4:	

Tabela 4: Identificação das Consultas Externas da Aplicação

Tabela 5 - Contagem de Saídas Externas (SEs): Funcionalidades que apresentam informações para o usuário **com** utilização de cálculos ou algoritmos para derivação de dados ou atualização de Arquivos Lógicos Internos ou mudança de comportamento da aplicação. São as consultas ou relatórios com totalização de dados, relatórios estatísticos, gráficos, geração de arquivos (xls, pdf, etc.) com atualização *log*, *downloads* com cálculo de percentual, entre outros.

Considerações: Você está desenvolvendo uma funcionalidade para apresentar informações para o usuário: uma consulta ou relatório com totalização de dados, etiquetas de código de barras, gráficos, relatórios estatísticos, *download* com percentual calculado, geração de arquivo pdf, xls contendo dados calculados ou com atualização de *log*? Caso positivo, estas funções devem ser identificadas como Saídas Externas. Observe que esta função *deve* ter cálculos ou algoritmos para processar os dados referenciados nos arquivos lógicos ou atualizar campos (normalmente indicadores) nos arquivos ou mudar o comportamento da aplicação ou criar dados derivados. Caso não haja conhecimento sobre o processo elementar (funcionalidade) analisado, considere a Saída Externa com complexidade **Média**.

Nº SEs Baixa:	X 4 PF
Nº SEs Média:	X 5 PF
Nº SEs Alta:	X 7 PF
Total PF da Tabela 5:	

Tabela 5: Identificação das Saídas Externas da Aplicação

A Estimativa de tamanho do projeto em PFs deve ser gerada totalizando-se os PFs obtidos nas **Tabelas 1, 2, 3, 4, e 5**.

A fórmula de contagem ou de estimativa de Pontos de Função para Projetos de Desenvolvimento é a seguinte:

$$\text{PF_Desenvolvimento} = \text{PF_Incluído} + \text{PF_Conversão}$$

Observação 1: PF_Conversão: Pontos de Função associados às funcionalidades de conversão de dados dos projetos. Exemplos de funções de conversão incluem: migração

ou carga inicial de dados para popular as novas tabelas criadas no sistema e relatórios associados à migração de dados.

Observação 2: Em projetos de redesenvolvimento de sistemas em outra plataforma com mudança do tipo do Sistema Gerenciador de Banco de Dados (por exemplo: ADABAS – hierárquico para Oracle - relacional), a migração de dados deve ser tratada como um novo projeto de desenvolvimento. Desta forma, não serão contadas as funções de conversão de dados. Observe que serão dois projetos de desenvolvimento, o desenvolvimento do sistema propriamente dito e o desenvolvimento do projeto de migração de dados.

Observação 3: Em projetos para o cliente RFB, na estimativa inicial além do tamanho funcional, devem ser estimadas as horas de especificação do projeto. Esta informação deve ser preenchida na planilha na aba de atividades sem Contagem de Pontos de Função. Sugere-se que a estimativa do homem_hora de especificação seja realizado da seguinte maneira: Multiplicar a contagem de Pontos de Função pela produtividade padrão de 12 hh/PF e aplicar um fator de 20%. Por exemplo, suponha um projeto com 100 PFs estimados, a estimativa de homem_hora de especificação é a seguinte: Especificação = $100 \text{ PF} \times 12 \text{ horas/PF} \times 20\% = 240 \text{ horas}$.

3.2 Estimativa de Esforço de Projetos de Software

Uma vez que o tamanho do projeto está estimado em Pontos de Função, o próximo passo é estimar o esforço de desenvolvimento projeto, bem como sua distribuição pelas fases do ciclo de vida do desenvolvimento do software. A Engenharia de Software possui vários modelos para estimar esforço de projetos de software, baseados em Pontos de Função, sendo o Modelo Simplificado de Estimativas [Vazquez, 2010] e o Modelo COCOMO II [Boehm, 2000] os mais utilizados. No SERPRO é adotado o modelo Simplificado de Estimativas. Futuramente, pretende-se utilizar também o COCOMO II. A implantação do COCOMO II exige um esforço para a calibração do banco de dados históricos de projetos concluídos da empresa e o apoio de consultoria especializada.

O modelo simplificado de estimativas consiste em obter um índice de produtividade em horas/PF para o projeto em questão, e então multiplicar o tamanho em PF do Projeto pelo índice de produtividade, conforme a fórmula [Vazquez, 2010]:

Esforço (horas) = Tamanho (PF) x Índice de Produtividade (HH/PF)

É fundamental obter os índices de produtividade dos vários tipos de projetos do SERPRO, considerando, dentre outros: plataforma tecnológica, complexidade do domínio, experiência da equipe alocada, tamanho do projeto, tipo de manutenção, reuso de componentes. Assim, com base em análise de dados históricos de projetos do SERPRO, Benchmarking e análise de literatura específica, foi definida uma Tabela de Produtividade do SERPRO (Tabela 6) para ser utilizada nas estimativas de esforço da empresa. Caso o projeto seja desenvolvido utilizando várias linguagens de programação, considere a linguagem predominante, ou seja a linguagem com maior percentual de funcionalidades desenvolvidas.

É importante ressaltar que algumas fases contidas em projetos de software devem ser consideradas separadamente, incluindo o esforço, o custo e o prazo associados. No ciclo de vida do software, são consideradas as fases de requisitos, arquitetura, implementação, testes, homologação e implantação. A fase de negócios e demais atividades de capacitação e de consultoria devem ser tratadas a parte.

Em alguns contratos, por exemplo o contrato com o cliente RFB, a fase especificação de Requisitos é considerada fora do ciclo de vida do software, apenas para a finalidade de faturamento. Desta forma, apenas para efeitos de faturamento, deve ser informado para a SUNAC o esforço em homem hora da especificação de requisitos – até a assinatura do Anexo 2 e a contagem de Pontos de Função do projeto. Para aferição dos indicadores de produtividade da empresa, são consideradas todas as fases do ciclo de vida, independentemente da forma de faturamento da URC.

A Tabela de Produtividade (Tabela 6) é uma referência para projetos típicos. Os projetos com características específicas de alta complexidade ou baixa complexidade, equipe iniciante, etc. devem ter sua produtividade analisada separadamente. Para estes projetos sugere-se utilizar também outros modelos de estimativas para apoiar a análise de dados. Esta Tabela deve ser atualizada periodicamente com base no *feedback* das equipes de desenvolvimento da empresa e análises de dados dos indicadores de produtividade do SERPRO.

Plataforma de Desenvolvimento	Produtividade (horas/PF)		
	Baixa	Média	Alta
ACCESS	10	8	6
ASP	16	11	6
ASPNET	11	8	5
ASSEMBLER	18	12	8
BASIC	18	12	8
C	24	18	12
C#	17	12	7
C++	18	12	8
CKAN	18	14	8
CLIPPER	12	8	6
COBOL	14	10	6
COMPONENTE – CÓDIGO PROPRIETÁRIO	26	19	12
COMPONENTE – CÓDIGO ABERTO	26	19	12
CSP	16	10	8
Dardo /Netuno	18	14	12
DATA DISCOVERY	16	12	6
DELPHI	12	8	6
Dot Net (.Net)	14	12	10
DW OUTRAS (apenas OLAP) – Microstrategy, Power Center e etc	14	10	5
DW OUTRAS (Extração e OLAP) - Microstrategy, Power Center e etc	16	12	6
DW PENTAHO (Apenas OLAP)	16	10	6
DW PENTAHO (Extração e OLAP)	18	12	8
EXCEL	6	5	4
FORMS/REPORTS/ORACLE	16	12	6
HTML	10	8	4
JAVA	14	10	6
Java AndroMDA	15	10	5
Java Demoiselle V 1.0	16	11	6
Java Demoiselle V 2.0	19	13	7
Java Flex	15	12	8
Java Script	16	12	8

Plataforma de Desenvolvimento	Produtividade (horas/PF)		
	Baixa	Média	Alta
Java Web não Distribuída	16	11	6
JCUPIM	41	28	15
Joomla	18	14	8
LASER XEROX	30	20	16
LIFERAY	16	12	8
LIGHTBASE	18	12	6
LOTUS NOTES	8	6	4
LTD	18	13	8
MIDDLEWARE	26	19	12
MOBILE - ANDROID	18	14	12
MOBILE - ANDROID E IOS	18	14	12
MOBILE - HTML 5 e JQUERY MOBILE	18	14	12
MOBILE - IOS	18	14	12
MOBILE - PHONEGAP	18	14	12
MOBILE – WINDOWS PHONE	18	14	12
NATURAL (Batch e On-Line)	12	10	8
Oracle Designer 2000	12	6	4
PENTAHO (Projetos PENTAHO Não BI)	7	6	5
PHP	15	10	5
PL/SQL de demais SGBDs	11	9	7
PROJETOS DE GEOPROCESSAMENTO	27	19	11
PROJETOS DE GEORREFERENCIAMENTO	27	19	11
PROJETOS DE WORKFLOW	24	18	16
PYTHON	18	14	8
RUBY ON RAILS	18	14	8
UNIX SHELL SCRIPTS	18	14	8
VB-SCRIPT	16	14	8
VISUAL BASIC /Crystal Reports	12	8	6
VISUAL C++	16	14	7
VISUAL GEN	10	8	6

Plataforma de Desenvolvimento	Produtividade (horas/PF)		
	Baixa	Média	Alta
VISUAL INTERDEV	24	14	8
WEBSERVICE	26	19	12
ZOPE PLONE	17	11	5

Tabela 6: Tabela de Produtividade por Linguagem e Tipos de Projetos [SERPRO, 2014]

As linguagens marcadas em Vermelho são aquelas que a produtividade foi definida por analogia e/ou pesquisa de mercado. Ou seja, não foram identificados um quantitativo suficiente de projetos concluídos no SERPRO para uma análise estatística.

A linguagem “COMPONENTE – CÓDIGO ABERTO” deve ser definida para projetos de desenvolvimento ou manutenção de componentes, tais como middlewares ou webservices, que serão implementados em plataformas abertas nas linguagens HTML, Java, Java AndroMDA, Java Demoiselle V 1.0, Java Demoiselle V 2.0, Java Web não Distribuída, Pentaho, PHP, Python e Zope Plone. Para projetos de desenvolvimento ou manutenção de componentes que utilizem outras linguagens, utilizar “COMPONENTE – CÓDIGO PROPRIETÁRIO”.

Em geral, projetos típicos devem ser estimados com a produtividade média. Os projetos de manutenções que a equipe possua conhecimento do sistema podem ser estimados com a produtividade alta. Os projetos com requisitos não funcionais complexos podem ser estimados com a produtividade baixa.

A fórmula utilizada para o cálculo de esforço total de um projeto (EP) é a seguinte [SERPRO, 2008]:

$$EP = QHC + QHA + (QPF \times EPF)$$

Onde:

QHC = Quantidade de Horas Perfil Consultor

QHA = Quantidade de Horas Perfil Analista

QPF = Tamanho do Projeto em PF

EPF = Esforço para implementar um PF na plataforma em questão

3.2.1 Distribuição de Esforço por Fase do Projeto

O próximo passo é a definição da distribuição de esforço pelas fases do projeto, visando definir o valor agregado ao projeto após cada fase do ciclo de vida.

Os contratos estabelecidos com os clientes determinam o processo de desenvolvimento a ser seguido com percentual de esforço por fases. Assim, se existirem cláusulas contratuais tratando o esforço, deve-se seguir o contrato.

A Tabela 7 apresenta a Distribuição de Esforço do Roteiro de Métricas do SISP, a ser considerada em contratos com o Ministério do Planejamento e adotada pelo Roteiro SERPRO.

A Tabela 8 apresenta a Distribuição de Esforço do Roteiro de Métricas da RFB, a ser considerada em contratos com o cliente RFB. O Contrato RFB considera a fase de especificação de requisitos (até a assinatura do Anexo 2) como um projeto a parte faturado em homem_hora. Desta forma, a Engenharia de Requisitos está fora do processo de desenvolvimento de sistemas para o cliente RFB.

Fases do Processo de Desenvolvimento de Software	Percentual de esforço (%)
Engenharia de Requisitos	25%
Design, Arquitetura	10%
Implementação	40%
Testes	15%
Homologação	5%
Implantação	5%

Tabela 7: Distribuição de Esforço de Projeto – Roteiro SISP

Fases do Processo de Desenvolvimento de Software	Percentual de esforço (%)
Gestão de Requisitos	5%
Design, Arquitetura	15%
Implementação	50%
Testes	15%
Homologação	10%
Implantação	5%

Tabela 8: Distribuição de Esforço de Projeto sem a fase de especificação – Cliente RFB

Para o cliente STN o percentual de esforço das fases é definido em contrato e aplicado nas contagens via SIGOP (Sistema da SUNAF).

A estimativa de prazo é abordada na seção seguinte.

3.3 Estimativa de Prazo de Projetos de Software

As estimativas de prazo não são lineares com o tamanho do projeto. O melhor tempo de desenvolvimento, no qual há um a melhor relação custo x benefício de alocação de recursos e menor prazo de desenvolvimento, dado o tamanho de um projeto específico, tem o uso recomendado por esse roteiro. Jones [Jones, 2007] propõe uma fórmula para o cálculo do melhor tempo de desenvolvimento, denominado T_d e de Região Impossível (RI) de desenvolvimento, onde a adição de mais recursos ao projeto não implicará em redução no prazo (Figura 3).

Note que a curva (Figura 3) mostra que quanto menor o prazo almejado para a conclusão do projeto, maior será o esforço requerido e conseqüentemente maior o custo do projeto. O aumento do esforço para reduzir o prazo acontece através da realização de horas-extras e da inclusão de pessoal adicional. No entanto, a redução de prazo tem um limite, como demonstra a região impossível da Figura 3.

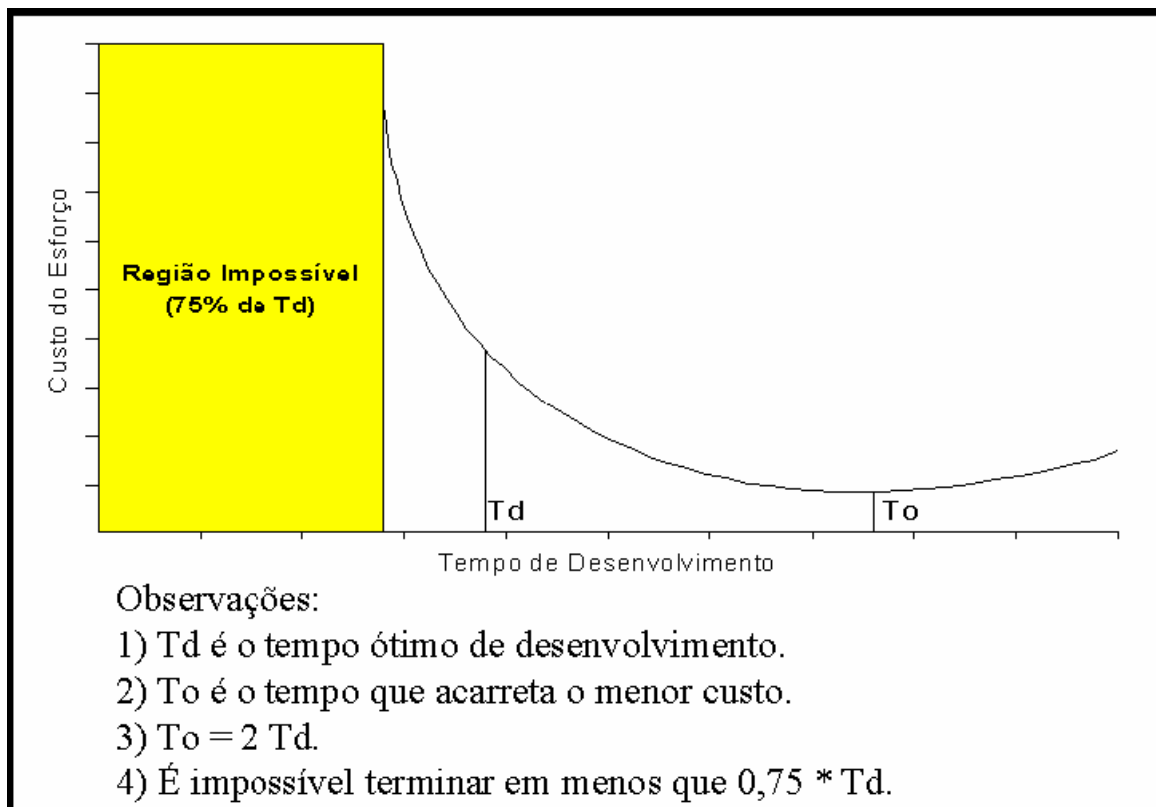


Figura 3: Relação entre a Estimativa de Prazo e de Esforço

O método utilizado para estimar o prazo dos projetos (Td) é baseado na fórmula de Capers Jones [Jones, 2007]. Posteriormente, pretende-se implantar o modelo COCOMO II para obtenção de mais de uma estimativa de prazo para o projeto. A fórmula de Capers Jones estima o prazo, baseando-se no tamanho do projeto em Pontos de Função, da seguinte maneira:

$$Td = V^t$$

Onde:

Td: prazo de desenvolvimento em meses

V: tamanho do projeto em Pontos de Função

t: o expoente t é definido de acordo com a Tabela 9

Tipo de Sistema	Expoente t
Sistema Comum – Mainframe (desenvolvimento de sistema com alto grau de reuso ou manutenção evolutiva)	0,32 a 0,33
Sistema Comum – WEB ou Cliente Servidor	0,34 a 0,35
Sistema OO (se o projeto OO não for novidade para equipe, não tiver o desenvolvimento de componentes reusáveis, considerar sistema comum)	0,36
Sistema Cliente/Servidor (com alta complexidade arquitetural e integração com outros sistemas)	0,37
Sistemas Gerenciais complexos com muitas integrações, Datawarehousing, Geoprocessamento, Workflow	0,39
Software Básico, Frameworks, Sistemas Comerciais, Projetos de Desenvolvimento ou Manutenção de Componentes	0,40

Tabela 9: Expoente t por tipo de Projeto

É importante destacar que o método só funciona para projetos a partir de 100 PFs. O prazo calculado considera todo o ciclo de vida do projeto.

Caso o cliente precise receber o projeto em um prazo menor que o Td calculado, recomenda-se propor um processo de desenvolvimento incremental, priorizando

funcionalidades em cada iteração de acordo com a necessidade dele. Caso, ainda assim, a estimativa não atenda às necessidades do cliente, então pode-se reduzir o Td em até 25%, observando-se a Região Impossível. No entanto, quanto mais perto da Região Impossível, o esforço e o custo do projeto aumentam de maneira exponencial. De um modo geral, a redução de prazo de 10 % implica no aumento de esforço de 20% (projetos urgentes); a redução de prazo de 20% implica no aumento de esforço de 50% (projetos críticos); a redução de prazo de 25% implica em um aumento de esforço de 70% (projetos de alta criticidade). Esse esforço deve ser considerado no custo do projeto em questão.

Não é recomendada a redução de prazo, devido ao alto risco. Deve-se buscar priorizar funcionalidades trabalhando com o processo incremental.

Caso o projeto seja menor que 100 PF, obtenha o prazo por meio de WBS, com base em dados históricos de outras manutenções do projeto.

Além disso, é importante observar a existência de cláusulas contratuais associadas à definição de prazo para projetos muito pequenos (menores que 100PF). O Roteiro de Métricas do SISP apresenta a seguinte distribuição (Tabela 10).

Tamanho do Projeto	Prazo máximo (em dias úteis)	
	Projetos Complexidade Baixa	Projetos Complexidade Média
Até 10 PF	9 dias	15 dias
De 11 PF a 20 PF	18 dias	30 dias
De 21 PF a 30 PF	27 dias	45 dias
De 31 PF a 40 PF	36 dias	60 dias
De 41PF a 50 PF	45 dias	75 dias
De 51 PF a 60 PF	54 dias	90 dias
De 61 PF a 70 PF	63 dias	105 dias
De 71 PF a 85 PF	70 dias	110 dias
De 86 PF a 99 PF	79 dias	110 dias

Tabela 10: Estimativa de Prazo de Projetos menores que 100 PF

Na seção seguinte é abordada a questão da distribuição de esforço e alocação de pessoas ao projeto em questão.

3.3.1 Alocação de Equipe ao Projeto

Na alocação de equipe, deve ser considerada a estimativa de prazo e a de esforço. A fórmula utilizada é a seguinte:

$$\text{Equipe} = \text{Esforço (HH)} / (21 \times \text{prod. diária} \times \text{Prazo})$$

Onde:

prazo = Td em meses

Prod. Diária = 6h/dia, 7h/dia e 8h/dia

O SGI e o PSDS sugerem a utilização de 8 horas/dia. Recomenda-se considerar 7 horas/dia em pareceres técnicos – Referência COCOMO.

21 = dias úteis contidos em 1 mês

O tamanho da equipe é obtido em quantidade de recursos para o desenvolvimento do projeto, deve-se considerar percentuais de alocação. Por exemplo, suponha uma Equipe de 2,2 recursos. Esta equipe pode conter 5 pessoas, sendo que 4 pessoas com 50% de alocação e um líder de projeto com 20% de alocação ao projeto.

3.4. Método para Estimativa de Custo

A estimativa de custo do projeto deve levar em consideração o custo da mão de obra, considerando o esforço e o custo da hora de todos os profissionais envolvidos no desenvolvimento da solução de software. Além do custo da mão de obra, devem ser considerados outros custos, tais como: treinamento, consultoria, viagens, licenças de software, custos indiretos etc. Também devem ser considerados os custos dos recursos computacionais descritos na seção seguinte.

Sugere-se a seguinte fórmula para calcular o custo relativo a mão de obra para o desenvolvimento da solução (CP – Custo do Projeto).

$$\text{CP} = (\text{QHC} \times \text{VPC}) + (\text{QHA} \times \text{VPA}) + (\text{QPF} \times \text{EPF} \times \text{VPA}) + \text{Outros Custos}$$

Onde:

QHC = Quantidade de Horas Perfil Consultor

VPC = Valor da Hora do Perfil Consultor

QHA = Quantidade de Horas Perfil Analista

VPA = Valor da Hora do Perfil Analista

QPF = Tamanho do Projeto em PF

EPF = Esforço para implementar um Ponto de Função na plataforma em questão

Caso a equipe de desenvolvimento esteja atuando em um contrato de preço fixo para um PF do projeto, então pode-se considerar o seguinte:

$$\mathbf{CP = (QHC \times VPC) + (QHA \times VPA) + (QPF \times VPF)}$$

Onde:

VPF = Valor do PF para o projeto em questão

É importante destacar que como o esforço para a construção do PF é variável, o preço do PF também é variável de acordo com os requisitos não funcionais do projeto. O preço para uma determinada demanda será obtido a partir da quantidade de Pontos de Função, sem considerar o esforço, multiplicado pelo valor unitário do item na tabela de Serviços Padrão do Sistema de Orçamento Técnico do SERPRO, para o Ambiente/Linguagem correspondente. O preço do desenvolvimento de software deve ser calculado seguindo a fórmula abaixo:

$$\mathbf{Preço = Tamanho (PF) \times Valor unitário do PF correspondente (R\$)}$$

Onde:

Tamanho (PF): Quantidade de PFs contados para o projeto de desenvolvimento da solução.

Valor Unitário do PF correspondente: Identificado de acordo com a Tabela de Serviço Padrão.

Deve-se realçar que o preço de uma solução de software também deve levar em consideração as atividades associadas ao processo de desenvolvimento de soluções que não possuem Pontos de Função associados, por exemplo: atividades de negócios, capacitação, etc.

3.5 Estimativa de Recursos Computacionais

A Estimativa de Recursos Computacionais também deve ser considerada, esta constitui um componente importante para as estimativas de custos dos projetos. Um recurso computacional é um hardware que se precisa adquirir; ou que se possui, mas precisa-se configurar. Exemplos de recursos computacionais incluem, dentre outros: espaço

em disco para o sistema entrar em produção, um servidor específico para teste ou homologação do sistema.

Devem ser registradas as seguintes informações associadas aos recursos computacionais críticos:

•**Nome do Recurso Computacional:** [considere exclusivamente hardware: micro, periférico, expansão de memória, área em disco, banda de rede, etc]

•**Descrição:**

•**Responsável pela Disponibilização:**

•**Data Limite:**

•**Parâmetros:** [características do recurso: quantidade, perfil, configuração, etc]

•**Tipo do Recurso:** [D: recurso para ambiente de Desenvolvimento; P: recurso para ambiente de Produção; H: recurso para ambiente de Homologação]

•**Custo (Opcional):** [Preencher este campo quando for possível a definição dos custos envolvidos com o recurso computacional. Não considerar custos de processamento ou custos operacionais de produção. Este custo irá compor o custo do projeto]

Caso o projeto a ser desenvolvido não possua nenhum recurso computacional crítico, deve ser registrado no documento de estimativas que o projeto não possui nenhum recurso computacional crítico.

4. Contagem de Pontos de Função de Projetos

Esta seção tem como propósito apresentar a contagem de Pontos de Função de projetos de desenvolvimento e descrever os diversos tipos de projetos de manutenção e mostrar métricas baseadas em Pontos de Função para dimensionar tais projetos, visto que o manual de práticas de contagem – CPM não contempla projetos de manutenção (*maintenance*) apenas o de Melhoria (*enhancement*).

Quanto à documentação de projetos de manutenção pequenos (menores que 100 PF) [Jones, 2007], deve-se documentar os requisitos da demanda em questão de forma detalhada e atualizar a documentação da aplicação impactada pela demanda, visando apoiar a contagem de Pontos de Função da demanda. É importante também documentar as estimativas e a contagem de Pontos de Função.

4.1 Projeto de Desenvolvimento

Um Projeto de Desenvolvimento tem objetivo construir e entregar a primeira versão de uma aplicação de software. A contagem de Pontos de função de um projeto de desenvolvimento é definida pelo CPM de acordo com a fórmula abaixo:

$$\text{PF_Desenvolvimento} = \text{PF_Incluído} + \text{PF_Conversão}$$

Observação 1: PF_Conversão: Pontos de Função associados às funcionalidades de conversão de dados dos projetos. As funções de migração e conversão de dados são processos elementares contidos em um projeto de desenvolvimento necessários para a sua implantação, que têm por objetivo: migração de dados oriundos de outros sistemas ou tabelas, com ou sem transformação; carga inicial de dados para popular as novas tabelas ou novos campos em tabelas já existentes; atualização de dados legados para manter consistência com o projeto de melhoria; relatórios de exceção, erros, conversão ou de controle necessários para garantir a integridade dos dados que estão sendo convertidos.

Observação 2: Em projetos de redesenvolvimento de sistemas em outra plataforma, podemos ter os seguintes tipos de migração de dados:

- Conversão de Dados: Os requisitos de carga inicial de dados nas tabelas da outra plataforma fazem parte do projeto de desenvolvimento. Geralmente, esta demanda ocorre quando não há mudanças no Sistema Gerenciador de Banco de Dados.
- Apuração Especial: A carga de dados é realizada em uma demanda a parte de Apuração Especial do tipo Banco de Dados. Neste caso, conta-se o desenvolvimento do sistema propriamente dito como Projeto de Desenvolvimento e a migração de dados como Apuração Especial Banco de Dados. Neste caso também não há mudança no Sistema Gerenciador de Banco de Dados
- Projeto de Migração de Dados: a migração de dados é tratada como um novo projeto de desenvolvimento. Desta forma, não serão contadas as funções de conversão de dados no Projeto de Desenvolvimento. Observe que serão dois projetos de desenvolvimento, o desenvolvimento do sistema propriamente dito e o desenvolvimento do projeto de migração de dados. Nestes casos há mudança no Sistema Gerenciador de Banco de

Dados, por exemplo, de ADABAS para ORACLE. A Contagem de PF do Projeto de Migração de Dados é descrita na seção seguinte.

4.2 Projeto de Migração de Dados

Este Roteiro recomenda a supressão do PF_CONVERSÃO das fórmulas de contagem de Pontos de Função de Desenvolvimento e de Melhoria, em casos de mudança plataforma com alteração do Sistema Gerenciador de Banco de Dados, e o tratamento das funções de migração de dados como um projeto separado de migração de dados.

Os projetos de migração de dados devem ser contados com um novo projeto de desenvolvimento de um sistema, contemplando minimamente: os ALIs mantidos pela migração, as Entradas Externas considerando as cargas de dados nos ALIs. Os AIEs de outras fronteiras usados na validação de dados durante as cargas e caso seja solicitado pelo usuário devem ser contados, caso exista o requisito de validação de dados. Cabe ressaltar que os dados provenientes de outras fronteiras lidos e carregados nos ALIs devem ser contados APENAS como Tipos de Dados das Entradas Externas, ou seja não devem ser contados como AIEs. Os relatórios gerenciais das cargas, caso solicitado pelo usuário, devem ser contados como Saídas Externas, geralmente possuem dados calculados. Todas as contagens de PF devem ser realizadas com base nas funcionalidades requisitadas e recebidas pelo usuário.

4.3 Projeto de Melhoria

O Projeto de Melhoria, também denominado de projeto de melhoria funcional, ou manutenção evolutiva, está associado às mudanças em requisitos funcionais da aplicação, ou seja, a inclusão de novas funcionalidades, alteração ou exclusão de funcionalidades em aplicações implantadas.

Segundo o padrão IEEE Std 1229 [IEEE 1229], esta manutenção seria um tipo de manutenção adaptativa, definida como: modificação de um produto de software concluído após a entrega para mantê-lo funcionando adequadamente em um ambiente com mudanças. O projeto de melhoria é considerado um tipo de projeto de manutenção

adaptativa com mudanças em requisitos funcionais da aplicação, ou seja com funcionalidades incluídas, alteradas ou excluídas na aplicação, segundo o CPM 4.3.

Este documento separa o projeto de melhoria, quando as mudanças são associadas aos requisitos funcionais e a manutenção adaptativa quando as mudanças estão associadas aos requisitos não funcionais da aplicação.

Um projeto de melhoria consiste em demandas de criação de novas funcionalidades (grupos de dados ou processos elementares), demandas de exclusão de funcionalidades (grupos de dados ou processos elementares) e demandas de alteração de funcionalidades (grupos de dados ou processos elementares) em aplicações implantadas em produção.

Uma função de dados (Arquivo Lógico Interno ou Arquivo de Interface Externa) é considerada alterada, quando a alteração contemplar mudanças de tipos de dados, inclusão ou exclusão de tipos de dados. Ou mudança de tamanho (número de posições) ou tipo de campo (por exemplo: mudança de numérico ou alfanumérico), sendo que esta ocorre por mudança de regra de negócio do usuário.

Uma função transacional (Entrada Externa, Consulta Externa e Saída Externa) é considerada alterada, quando a alteração contemplar:

- Mudança de tipos de dados em uma função existente;
- Mudança de arquivos referenciados;
- Mudança de lógica de processamento, segundo as ações das lógicas e processamento do CPM 4.3:

A **Lógica de Processamento** é definida 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;

O Roteiro de Métricas, em aderência aos Acórdãos do TCU, utiliza a seguinte fórmula:

$$\text{PF_MELHORIA} = \text{PF_INCLUIDO} + (0,75 \times \text{PF_ALTERADO}) + (0,40 \times \text{PF_EXCLUIDO}) + \text{PF_CONVERSÃO}$$

Definições:

PF_INCLUÍDO = Pontos de Função associados às novas funcionalidades que farão parte da aplicação.

PF_ALTERADO = Pontos de Função associados às funcionalidades existentes na aplicação que serão alteradas no projeto de manutenção.

PF_EXCLUÍDO = Pontos de Função associados às funcionalidades existentes na aplicação que serão excluídas no projeto de manutenção.

PF_CONVERSÃO = Pontos de Função associados às funcionalidades de conversão de dados dos projetos de melhoria. As funções de migração e conversão de dados são processos elementares contidos em um projeto de desenvolvimento necessários para a sua implantação, que têm por objetivo: migração de dados oriundos de outros sistemas ou tabelas, com ou sem transformação; carga inicial de dados para popular as novas tabelas ou novos campos em tabelas já existentes; atualização de dados legados para manter consistência com o projeto de melhoria; relatórios de exceção, erros, conversão ou de controle necessários para garantir a integridade dos dados que estão sendo convertidos. Exemplos de funções de conversão incluem: carga inicial de dados para popular as novas tabelas criadas e relatórios associados à migração de dados.

Observação: Em caso de projetos não desenvolvidos pelo SERPRO, sem documentação atualizada, as funções alteradas serão contadas como PF_INCLUÍDO. Em caso de projetos legados desenvolvidos pelo SERPRO, sem documentação atualizada, as funções alteradas também serão contadas como PF_INCLUÍDO.

O Roteiro de Métricas do SISP versão 2.0 utiliza a seguinte fórmula

$$PF_MELHORIA_SISP = PF_INCLUIDO + (0,40 \times PF_EXCLUIDO) + (FI \times PF_ALTERADO) + PF_CONVERSÃO$$

FI = Fator de Impacto pode variar de 50% a 90% de acordo com o seguinte:

- Funcionalidade de sistema desenvolvida ou mantida pelo SERPRO: FI = 50%
- Funcionalidade de sistema não desenvolvida ou mantida pelo SERPRO e sem necessidade de redocumentação da funcionalidade: FI = 75%
- Funcionalidade de sistema não desenvolvida ou mantida pelo SERPRO com necessidade de redocumentação da funcionalidade : FI =90%.

O Roteiro de Métricas da STN considera o Fator de Impacto de 85% para as funcionalidades de conversão de dados, 30% para as funcionalidades excluídas e 75% para as funcionalidades alteradas.

$$PF_MELHORIA_STN = PF_INCLUIDO + (PF_ALTERADO \times 0,75) + (0,30 \times PF_EXCLUIDO) + (0,85 \times PF_CONVERSÃO)$$

Para funcionalidades alteradas não desenvolvidas ou mantidas pelo SERPRO não há utilização de Fator de Impacto

4.4 Manutenção Corretiva

Mesmo com a execução de atividades de garantia da qualidade, o cliente pode identificar defeitos na aplicação entregue. A manutenção corretiva altera o software para correção de defeitos. Encontra-se nesta categoria, as demandas de correção de erros (bugs) em funcionalidades em sistemas em produção.

É importante destacar que as demandas de manutenção corretiva frequentemente precisam ser atendidas com urgência. Assim, o grau de criticidade do projeto poderá trazer impacto nas estimativas de custo e esforço. O padrão IEEE [IEEE,1998] define um tipo de manutenção corretiva, denominado de Manutenção Emergencial como “manutenção corretiva não programada executada para manter o sistema em estado operacional”.

Quando o sistema não for desenvolvido pelo SERPRO ou a funcionalidade em questão estiver fora do prazo de garantia estabelecido em cláusula contratual, esta manutenção deverá ser cobrada do cliente.

As manutenções corretivas faturadas, com ônus para o cliente, devem ser dimensionadas como PF_Alterado de um projeto de melhoria. Estas demandas serão classificadas como projetos de melhoria.

As manutenções não faturadas, ônus SERPRO, deverão ser dimensionadas para apoiar as estimativas de esforço e a aferição de indicadores, de acordo com a seguinte fórmula.

$$\text{PF_CORRETIVA} = \text{PF_ALTERADO} \times 0,75$$

4.5 Redesenvolvimento de Projetos em outra Plataforma

São considerados nesta categoria projetos que precisam ser migrados para outra plataforma. Os projetos com mudança linguagem de desenvolvimento, por exemplo, um sistema em COBOL que precisa ser redesenvolvido em JAVA, serão considerados como novos projetos de desenvolvimento.

$$\text{PF_REDESENVOLVIMENTO} = \text{PF_INCLUÍDO} + \text{PF_CONVERSÃO}$$

Em caso de mudança de plataforma - Banco de Dados, demandas de redesenvolvimento de sistemas para executar em um outro Sistema Gerenciador de Banco de Dados, deve-se verificar se mudança é de Banco de Dados Hierárquico para Relacional ou de Relacional para Relacional.

Em casos de **mudança de banco hierárquico para relacional**, deve-se considerar como um novo projeto de desenvolvimento, seguindo a fórmula abaixo:

$$\text{PF_REDESENVOLVIMENTO_BD_HIERÁRQUICO} = \text{PF_INCLUÍDO}$$

Em projetos de redesenvolvimento de sistemas em outra plataforma, com mudança de Banco de Dados hierárquico para relacional, a migração de dados deve ser tratada como um novo projeto de desenvolvimento. Desta forma, não serão contadas as funções de conversão de dados. Observe que serão dois projetos de desenvolvimento, o desenvolvimento do sistema propriamente dito e o desenvolvimento do projeto de migração de dados. A contagem de PF de projetos de migração de dados está descrita na Seção 4.2.

Em casos de **mudança de banco relacional para relacional**, então deve ser utilizada a seguinte fórmula:

PF_REDESENVOLVIMENTO_BD_RELACIONAL = (PF_ALTERADO X 0,50) + PF_CONVERSÃO

Nos Projetos de redesenvolvimento de Banco de Dados Relacional para outro Relacional, recomenda-se tratar o PF_CONVERSÃO dentro do mesmo projeto.

4.6 Atualização de Plataforma

São consideradas nesta categoria, demandas para uma aplicação existente ou parte de uma aplicação existente executar em versões mais atuais de *browsers* (ex: versão atual do *Internet Explorer*, *Mozilla*, *Firefox*,...) ou de linguagens de programação (ex: versão mais atual do *JAVA* ou do Banco de Dados). Também são consideradas nesta categoria aplicações *Web* desenvolvidas para executar em *Internet Explorer* que precisam executar também em *browser* em software livre e mudanças de arquitetura.

Tipicamente, as demandas de mudança de arquitetura podem ser dimensionadas como Manutenção Adaptativa em requisitos não funcionais ou Manutenção Preventiva ou Manutenção de Componentes. As funcionalidades apenas testadas devem ser contadas como PF Testes.

As demandas de atualização de versão de linguagem de programação, browser ou banco de dados podem ser contadas como manutenção de componentes, em casos de alteração de bibliotecas de configuração e PF Testes para as funcionalidades apenas testadas. Nos casos de alteração de funcionalidades estas devem ser dimensionadas considerando 30% do tamanho da funcionalidade alterada, de acordo com a fórmula abaixo:

$$\text{PF_Atualização_Versão} = 0,30 \times \text{PF_ALTERADO}$$

O Roteiro do cliente STN considera o faturamento do mesmo em homem_hora. Em muitos casos, estas demandas podem contadas como manutenção adaptativa em requisitos não funcionais – considerando as funcionalidades efetivamente alteradas e

PF_Testes para as funcionalidades apenas testadas. Em alguns casos, estas demandas poderão ser consideradas manutenção em componentes.

4.7 Manutenção em Interface

São consideradas manutenções em interface ou cosméticas, conforme denominado na literatura, as demandas associadas às alterações de interface, por exemplo, fonte de letra, cores de telas, logotipos, mudança de botões, inclusão de botões ou link na tela para chamada de funcionalidades existentes (navegação), mudança de posição de campos ou texto na tela ou em relatórios, manutenção em texto estático em telas ou relatórios.

Também se enquadram neste tipo de manutenção as mudanças de texto em mensagens específicas de uma funcionalidade, tais como erro, validação, aviso, alerta, confirmação de cadastro ou conclusão de processamento; Mudança em texto estático de e-mail enviado para o usuário em uma funcionalidade de cadastro. A demanda deve ser contada como manutenção em interface na funcionalidade de cadastro; Alteração de título de um relatório; Alteração de labels de uma tela de consulta.

Nestes casos, a aferição do tamanho em Pontos de Função da funcionalidade ou das funcionalidades alteradas considera 20% do PF_Alterado de uma função transacional de mais baixa complexidade (3 PF), independentemente da complexidade da funcionalidade alterada.

PF_INTERFACE = 0,6 x Quantidade de Processos Elementares Alterados

Caso ressaltar que a documentação da funcionalidade deve ser atualizada. Seguem exemplos de atualização de documentação:

- Sistemas onde as mensagens estão documentadas utilizando o artefato EMS (Especificação de Mensagens do Sistema), definido no PSDS, sendo necessário alterar este artefato de documentação.
- Sistemas que possuem protótipos e que portanto qualquer alteração cosmética no sistema em produção também deve ser realizada no protótipo.

Esta modalidade não contempla a redocumentação da aplicação.

4.8 Manutenção Adaptativa em Requisitos Não Funcionais e Manutenção Preventiva

Seguindo os conceitos da IEEE, existem vários tipos de Manutenção Adaptativa. Quando há mudança em requisitos funcionais, estes projetos são denominados de projetos de Melhoria, descritos na seção 4.3. Esta seção visa apresentar demandas de manutenções adaptativas e manutenções preventivas associadas às mudanças em requisitos não funcionais da aplicação, ou seja, alteração em Funcionalidades motivadas por Requisitos Não Funcionais.

São consideradas nesta categoria as demandas de manutenção adaptativa e de manutenção preventiva associadas à adequação de funcionalidades às mudanças de regras de negócio ou de Legislação ou requisitos de usabilidade **que não se enquadram nas funções alteradas do Projeto de Melhoria**, seguindo as regras de contagem do CPM. Observe que tais solicitações envolvem aspectos não funcionais, sem alteração em requisitos funcionais.

Seguem alguns exemplos de Manutenção Adaptativa.

- Aumentar a quantidade de linhas por página em um relatório;
- Incluir paginação em relatórios;
- Permitir exclusão múltipla em uma funcionalidade que antes só possibilitava a exclusão de um item;
- Adaptação da funcionalidade para possibilitar a chamada por um Webservice ou para outro tipo de integração com outros sistemas;
- 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, por exemplo, alteração em sub-rotinas chamadas por este software.

Seguem alguns Exemplos de Manutenção Preventiva.

- Manutenções Preventivas nos sistemas para evitar futuros problemas de performance, sobrecarga, ou ainda, mudanças para reestruturar programas ou dados a fim de melhorar a facilidade de manutenção ou outros atributos do software instalado.
- Mudanças no software ou hardware executadas para prevenir defeitos futuros ou falhas antes que eles se manifestem.

Nestes casos, a aferição do tamanho em Pontos de Função da funcionalidade ou das funcionalidades alteradas considera 75% do PF_Alterado, segundo a fórmula abaixo:

$$\text{PF_ADAPTATIVA_RNF} = \text{PF_ALTERADO} \times 0,75$$

$$\text{PF_PREVENTIVA} = \text{PF_ALTERADO} \times 0,75$$

É importante ressaltar que PF é uma métrica de tamanho funcional. Desta forma, deve-se ter cuidado com essa métrica definida por analogia. Em alguns casos, a mudança em requisitos não funcionais pode gerar um esforço muito grande em relação a essa métrica, por exemplo, em casos de mudança de arquitetura. Então, deve-se buscar a remuneração em homem_hora com justificativa por meio de Parecer Técnico emitido pela Área de Métricas do SERPRO em acordo com o cliente. No SGI, estas demandas de componentes de integração, middleware, dentre outras, terão contagem de Pontos de Função e devem ser classificadas na linguagem de desenvolvimento – Projeto de Componentes para a aferição da estimativa de esforço e do indicador de produtividade.

Os Roteiros de Métricas dos clientes do SERPRO tratam as manutenções preventivas como manutenções adaptativas em requisitos não funcionais.

O Roteiro SISP trata as manutenções adaptativas e preventivas como da seguinte maneira:

$$\text{PF_ADAPTATIVA_SISP} = \text{PF_ALTERADO} \times \text{FI}$$

FI = Fator de Impacto pode variar de 50% a 90% de acordo com o seguinte:

- Funcionalidade de sistema desenvolvida ou mantida pelo SERPRO: FI = 50%
- Funcionalidade de sistema não desenvolvida ou mantida pelo SERPRO e sem necessidade de redocumentação da funcionalidade: FI = 75%

- Funcionalidade de sistema não desenvolvida ou mantida pelo SERPRO com necessidade de redocumentação da funcionalidade: FI =90%.

4.9 Apuração Especial

São funcionalidades executadas apenas uma vez para: corrigir problemas de dados incorretos na base dados das aplicações ou atualizar dados em bases de dados de aplicações, detalhado no item 4.9.1; gerar um relatório específico ou arquivo para o usuário por meio de recuperação de informações nas bases da aplicação, detalhado no item 4.9.2. A seção 4.9.3 considera os casos de reexecução da apuração especial.

Algumas demandas de Apuração Especial não estão associadas à especificamente uma única fronteira. Estas demandas de Apuração Especial envolvendo várias fronteiras devem ser tratadas como um novo projeto de desenvolvimento. Por exemplo, uma demanda de geração de um relatório de contribuinte com dados calculados envolvendo leitura de dados em vários sistemas do IRPF. Esta demanda deve ser contada da seguinte maneira: os dados lidos devem ser contados como Arquivos de Interface Externa e o relatório com dados calculados deve ser contado como uma Saída Externa.

4.9.1. Apuração Especial – Base de Dados

Este tipo de apuração especial é um projeto que inclui a geração de procedimentos para atualização da base de dados. Deve-se destacar que estas funções são executadas apenas uma vez, não fazendo parte da aplicação, visando a correção de dados incorretos na base de dados da aplicação ou atualização em função de modificação da estrutura de dados, por exemplo, inclusão do indicador de matriz – sim ou não para um CNPJ. Nestes casos, a contagem de Pontos de Função das funcionalidades desenvolvidas. Geralmente, estas funcionalidades são classificadas como Entradas Externas. Nesse caso, como artefato de homologação da demanda, deve ser gerado um relatório para validação do usuário.

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.

Seguem as fórmulas de cálculo:

a) Atualização de Dados sem Consulta Prévia

$$PF_AESP_BD = PF_INCLUÍDO$$

b) Consulta Prévia com Atualização de Dados

Em alguns casos de Apuração Especial – Base de Dados, o usuário solicita uma consulta prévia das informações. Deve-se ressaltar que esta Consulta deve ser realizada antes da construção da funcionalidade, não é homologação. A Consulta Prévia geralmente é solicitada para a avaliação da viabilidade de implementar a Apuração Especial - Base de Dados. De fato, é uma prática interessante para evitar informações errôneas na base de produção dos sistemas. Esta Consulta Prévia, classificada como Consulta Externa ou Saída Externa deve ser dimensionada, considerando-se o tamanho da funcionalidade em questão, conforme a fórmula abaixo:

$$PF_CONSULTA_PRÉVIA = PF_INCLUÍDO \times 0,60$$

Observação: Caso não seja solicitada a atualização de dados Pós Consulta Prévia, então esta deve contada como Apuração Especial – Geração de Relatórios.

4.9.2. Apuração Especial – Geração de Relatórios

Este tipo de apuração especial é um projeto que inclui a geração de relatórios em uma ou mais mídias para o usuário. Em alguns casos, são solicitadas extrações de dados e envio dos dados para outros sistemas. Caso neste envio de dados sejam requisitadas atualizações no sistema de origem, então estas funções são Saídas Externas, devido à atualização do Arquivo Lógico Interno.

Deve-se destacar que estas funções são executadas apenas uma vez, não fazendo parte da aplicação. Nestes casos, considera-se contagem de Pontos de Função das funcionalidades desenvolvidas. Frequentemente, estas funcionalidades são classificadas como Saídas Externas. Também podem ser classificadas como Consultas Externas, caso não possuam cálculos ou criação de dados derivados.

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.

$$\text{PF_AESP_RELATÓRIO} = \text{PF_INCLUÍDO}$$

4.9.3. Apuração Especial – Reexecução

Em alguns casos, há interesse do contratante em executar uma apuração especial mais de uma vez. Nesses casos, o contratante deve solicitar formalmente à contratada o armazenamento do script executado. Desta forma, se for solicitada a reexecução de uma apuração especial, esta deve ser dimensionada com a aplicação de um fator considerado 10% na contagem de Pontos de Função da apuração especial em questão, da seguinte maneira:

$$\text{PF_AESP_REEXECUÇÃO} = \text{PF_INCLUÍDO} \times 0,10$$

Para o cliente RFB esta demanda é faturada em homem_hora, demandas de apuração com esforço inferior a 50 HH.

4.10 Atualização de Dados e PF Domínio

Em alguns casos, as demandas de correção de problemas em base de dados estão associadas a atualizações em um único registro. Por exemplo, atualização do nome de um Fornecedor cadastrado erroneamente. Estas demandas são realizadas de forma manual.

Nesse casos, a aferição do tamanho, em Pontos de Função, deve considerar 10% do PF_ALTERADO de Entrada Externa, os Tipos de Dados da Entrada Externa são os campos atualizados.

$$\text{PF_ATUALIZAÇÃO DE DADOS} = \text{PF_ALTERADO} \times 0,10$$

Este tipo de demanda não encontra-se previsto nos Roteiros do SERPRO com clientes. Esta prática de atualização manual de dados não está aderente às melhores práticas de desenvolvimento de sistemas, visto que é perdido todo o versionamento das atualizações da base de dados.

No entanto, esse tipo de demanda pode ser utilizado quando o cliente solicita inclusão de dados em uma Tabela de Domínio do tipo Code Data. Por exemplo, suponha

a atualização de uma Tabela de Situação de Projetos (Andamento, Concluído, Cancelado) para inclusão da Situação “Suspenso”. Então, esta demanda será classificada como Atualização de Dados e contada como uma EE: Atualizar Situação de Projetos (AR:1 Situação do Projeto TD: 1 Situação do Projeto) – Baixa – 3 PF.

4.11 Manutenção em Páginas Estáticas de *Intranet, Internet ou Portal*

Nesta seção são tratadas manutenções específicas em páginas estáticas de Portais, Intranets ou Websites ou Menus ou tela principal de aplicações cliente servidor. A demanda consiste na alteração dessas estruturas não contadas como funções transacionais pelo CPM, tais como: alteração de página de estilo, criação de página html, atualização de menu estático, atualização de texto ou banner estáticos em páginas html existentes. Em caso de alteração Banner, menu ou outro componente que se repete em várias páginas, considerando-se que não ocorra na mudança no código da página, tratar a demanda como PF_Componentes. Estas demandas são tratadas como desenvolvimento de consultas. Cada página é contada como uma consulta. Considera-se 20% dos Pontos de Função de uma Consulta Externa de complexidade baixa, segundo a fórmula abaixo:

$$\text{PF_Páginas} = 0,6 \times \text{Quantidade de Páginas Alteradas ou Incluídas}$$

4.12 Manutenção de Documentação de Sistemas Legados

Nesta seção são tratadas demandas de documentação ou atualização de documentação de sistemas legados. Observe que o desenvolvedor deve realizar uma Engenharia Reversa da aplicação para gerar a documentação. Para este tipo de projeto, caso a demanda seja apenas a documentação de requisitos, devem ser considerados 25% dos Pontos de Função da aplicação em questão, conforme a fórmula abaixo:

$$\text{PF_DOCUMENTAÇÃO} = \text{PF_NÃO_AJUSTADO} \times 0,25$$

Caso a demanda seja a geração de artefatos de documentação de todas as fases do processo de desenvolvimento, deve-se considerar um percentual mais alto de 30% a

50%, dependendo dos artefatos a serem gerados. As premissas utilizadas devem ser acordadas entre o SERPRO e o cliente e documentadas no documento de estimativas do projeto.

4.13 Verificação de Erros

São consideradas verificações de erro ou análise e solução de problemas as demandas referentes a todo comportamento anormal ou indevido apontado pelo cliente nos sistemas aplicativos. Neste caso, a equipe de desenvolvimento do SERPRO se mobilizará para encontrar a(s) causa(s) do problema ocorrido. Se for constatado erro de sistema, a demanda será atendida como manutenção corretiva.

Entretanto, uma vez não constatado o problema apontado pelo cliente ou o mesmo for decorrente de regras de negócio implementadas ou utilização incorreta das funcionalidades, será realizada a aferição do tamanho em Pontos de Função das funcionalidades verificadas, e será considerado 20% do tamanho funcional das funcionalidades analisadas, segundo a fórmula abaixo:

$$PF = PF_NÃO_AJUSTADO \times 0,20$$

É importante ressaltar que a demanda de verificação de erros deve ser associada a uma funcionalidade específica. Os casos de sistema fora do ar por conta de problemas em rede ou banco de dados devem ser tratados como serviços de suporte e não de Fábrica de Software. Esses serviços de suporte não fazem parte do escopo desse Roteiro de Métricas.

Para os clientes RFB e STN, a faturamento dessas demandas será dimensionado em homem_hora. Para o SGI este tipo de projeto é classificado como projeto de não software. Assim, esta contagem se aplica apenas para demandas do cliente MP.

4.14 Pontos de Função de Testes Funcionais

Muitas vezes, em projetos de manutenção o conjunto de funções de dados e funções transacionais a serem testadas é maior do que a quantidade de funções a serem implementadas, i.e., além das funcionalidades que são afetadas diretamente pelo projeto

de manutenção, outras precisam ser testadas [NESMA, 2009]. O tamanho das funções a serem testadas deve ser aferido em Pontos de Função de Teste Funcional (PFT). Não considerar as funcionalidades incluídas, alteradas ou excluídas do projeto de manutenção na contagem de Pontos de Função de Teste Funcional.

A contagem de PFT deve considerar o seguinte [NESMA, 2009]:

- Determinar o tamanho em Pontos de Função de cada função de dados ou transacional envolvida no teste.
- Calcular o tamanho em Pontos de Função de todas as funções de dados ou transacionais envolvidas no teste.

A contagem do Ponto de Função de Testes Funcionais (PFT) deve ser feita de acordo com a fórmula abaixo:

$$\text{PFT} = \text{PF_Funcionalidades_Apenas_Testadas} \times 0,15$$

É importante ressaltar que as funções testadas consideradas no PFT podem ser identificadas pela equipe de desenvolvimento ou requisitadas pelo cliente, devendo ser documentadas. Observe que estas funções farão parte do escopo do projeto de manutenção, sendo consideradas para efeito de estimativa de esforço e faturamento junto ao cliente.

Observação: Em alguns projetos de desenvolvimento ou de manutenção podem ocorrer demandas de testes integrados em outras fronteiras. Nestes casos, há demandas de teste de funcionalidades de outras aplicações. Desta forma, será contado Ponto de Função de Testes para estas funcionalidades de outras fronteiras.

4.15 Pontos de Função de Testes Não Funcionais

Em alguns projetos podem ocorrer demandas de testes não funcionais. O teste não funcional tem como propósito testar os atributos de um componente ou sistema que não se relacionam à funcionalidade, por exemplo: confiabilidade, eficiência, usabilidade, manutenibilidade e portabilidade¹. Os principais tipos de testes não funcionais realizados nos sistemas da RFB, dentre outros, são os seguintes: Testes de performance – para

¹ Fonte: Standard glossary of terms used in Software Testing Version 1.1 (September/2005) - International Software Testing Qualification Board; Editor : Erik van Veenendaal (The Netherlands)

verificar o tempo de resposta da aplicação; Testes de stress – para verificar a quantidade de usuários simultâneos suportados pela aplicação; Teste de integração – para verificar se os componentes, juntos, executam conforme está descrito nas especificações. Estes testes não são focados em “o quê” os componentes fazem, mas se eles "se comunicam" conforme especificado no desenho do sistema; Teste de segurança – para verificar o atendimento dos requisitos não funcionais de segurança; Teste de portabilidade – para verificar o comportamento da aplicação em vários ambientes, como por exemplo, testar um PGD em vários ambientes, testar uma aplicação web em diferentes browsers e testar a mesma aplicação em diferentes tipos de dispositivo móvel. As atividades de execução de testes não funcionais devem ser realizadas por especialistas em testes e dimensionadas de acordo com a seguinte fórmula:

$$\text{PF_Testes_Não_Funcionais} = \text{PF_Funcionalidades_Testadas} \times 0,25$$

Devem ser consideradas apenas as funções transacionais impactadas pela demanda do teste não funcional.

É importante ressaltar que PF é uma métrica de tamanho funcional. Desta forma, deve-se ter cuidado com essa métrica definida por analogia. Em alguns casos, o teste em requisitos não funcionais pode gerar um esforço muito grande em relação a essa métrica. Então, deve-se buscar a remuneração em homem_hora com justificativa por meio de Parecer Técnico emitido pela Área de Métricas do SERPRO em acordo com o cliente.

Deverão ser gerados documentos que evidenciem a realização dos testes não funcionais, caso o cliente solicite comprovação da realização destes testes.

4.16 Projeto de Teste Integrado

Nesta seção são tratadas as demandas específicas do cliente de realização de Testes Integrados, baseados em processos de negócio. Estas demandas consistem no teste do comportamento de funções de dados e transacionais de forma integrada com outros sistemas. Por exemplo, demanda de teste integrado do Macroprocesso de Crédito Tributário.

A contagem de PF destas demandas consiste no seguinte:

- Calcular o tamanho em Pontos de Função de todas as funções de dados ou transacionais envolvidas no teste.

$$PFI = PF_Funcionalidades_Apenas_Testadas \times 0,25$$

4.17 Pontos de Função de Massa de Teste para Homologação

Em alguns projetos podem ocorrer demandas de geração de massa de testes para homologação. Estas atividades consistem em carga de dados no Banco de Dados do Ambiente de Homologação. A realização destas atividades é similar à Conversão de Dados. Desta forma, o dimensionamento deve ser realizado da seguinte maneira: Contar uma Entrada Externa para cada Arquivo Lógico Interno ou Arquivo de Interface Externa carregado, desconsiderando os Registros Lógicos. A EE terá sempre apenas um Arquivo Referenciado (o Arquivo Lógico carregado) e os Tipos de Dados (TDs) serão os campos das tabelas carregados. Segue a fórmula de cálculo:

$$PF_Massa_Testes = PF_Incluído_Carga_Dados$$

Este Roteiro não contempla retrabalho para carga de Massa de Testes para Homologação. Após realização da carga de dados no Banco de Dados do Ambiente de Homologação, caso o cliente solicite mudanças nos requisitos do projeto, tornando necessário implementar e executar novas cargas, elas deverão ser consideradas na planilha de contagem, justificando no campo "Observações" da aba "Sumário" o fato de existirem duas cargas para o mesmo ALI na contagem.

4.18 Roteiro para Homologação

A elaboração de Roteiro para Homologação consiste na definição de um guia demonstrando os passos para utilizar as funcionalidades disponibilizadas na homologação do sistema. O produto gerado por esta atividade é um Roteiro para Homologação. O dimensionamento desta atividade será realizado considerando 2% da contagem de Pontos de Função das funções transacionais impactadas, de acordo com a seguinte fórmula:

$$PF_Roteiro_Homologação = PF_funcionalidades_impactadas \times 0,02$$

4.19 Gerenciamento de Riscos de Segurança

O desenvolvimento ou manutenção, geralmente em inclusão de novos módulos, de sistemas com complexidade alta pode tornar necessária a avaliação dos riscos de segurança antes da entrada do mesmo em produção. O produto gerado por esta atividade para o cliente é um Relatório de Riscos. O esforço de realização dessa atividade será remunerado com base na contagem de Pontos de Função de Referência do projeto de desenvolvimento ou do projeto de melhoria. O dimensionamento desta atividade será realizado considerando 2% da contagem de Pontos de Função de Referência, de acordo com a seguinte fórmula:

$$\text{PF_Riscos} = \text{PF_Referência} \times 0,02$$

4.20 Desenvolvimento e Manutenção de Help

Em alguns projetos há demandas específicas de desenvolvimento e/ou manutenção de help de funções (help de campo, help de tela ou relatório e help do sistema). O conteúdo do help deve ser elaborado pelo cliente. O dimensionamento destas demandas deve ser realizado de acordo com a seguinte fórmula:

$$\text{PF_HELP} = \text{PF_funcionalidades_impactadas} \times 0,05$$

Não considerar contagem de função de dados, nem massa de testes para homologação.

Caso a demanda seja apenas de Help de sistema, consistindo na disponibilização de texto enviado pelo usuário em um link de Help. Então, a demanda será contada como Manutenção de Componentes. Sendo considerando uma Consulta Externa de complexidade baixa.

Em casos de projetos de manutenção, considerando-se uma alteração apenas no Help de uma funcionalidade, então este help será contado como publicação de página estática. Observe que se a funcionalidade e o help forem alterados a demanda será contada como projeto de melhoria e Ponto de Função de Help. Se apenas a funcionalidade for alterada, sem impacto no Help, não será contado Ponto de Função de Help. Se o Help for um arquivo, a atualização de dados do Arquivo de Help será tratada como um Projeto de Atualização de Dados de Domínio.

Em casos de demandas de Elaboração de Manual de Usuário, a contagem de PF da elaboração do Manual de Usuário será realizada como PF_Help. Ou seja, deve se

contar todas as funcionalidades documentadas no Manual do Usuário com o Fator de Impacto de 5%. Na planilha de contagem, classificar a demanda como desenvolvimento ou melhoria e a classificação de função como Help.

4.21 Desenvolvimento e Manutenção de Componentes Internos Reusáveis

Em alguns casos são demandadas desenvolvimento e manutenções em componentes internos de uma aplicação, sendo estes reusados por várias funcionalidades da aplicação. Por exemplo, suponha uma mudança em uma rotina de validação de um CPF, mudança de rotina de validação NIRE do sistema CNPJ, usada em várias funcionalidades de cadastro. Se considerarmos o método de contagem de projetos de melhoria do CPM, seriam contadas todas as funcionalidades impactadas por esta mudança.

Seguem alguns exemplos de Manutenção de Componentes:

– Alteração de valores de elementos internos de configuração que afetem o comportamento ou a apresentação do sistema de forma geral, tais como arquivos com mensagens de erro, arquivos de configuração de sistema, páginas de estilo, arquivos de internacionalização.

– Mudança em item de um menu de um sistema Web que aparece em todas as telas da aplicação. A contagem pode ser realizada considerando o componente “Apresentar Menu”.

No entanto, este Roteiro propõe que o componente, o qual deverá ser implementado e testado, seja considerado um processo elementar independente e contado como uma funcionalidade. Cabe ressaltar, que a função será tratada como funcionalidade, assim um componente de apresentação de dados deve ser contado como uma Consulta Externa, o Arquivo Referenciado será o grupo lógico de dados apresentado, mesmo que este seja físico. Um componente que apresente dados e tenha um algoritmo de cálculo será tratado como Saída Externa. Um componente que a função seja atualizar um arquivo de configuração interna será classificado como Entrada Externa., o Arquivo Referenciado será o arquivo físico de configuração atualizado.

Além disso, as funcionalidades da aplicação que necessitem de teste devem ser requisitadas pela contratante e dimensionadas por meio da métrica Pontos de Função de Testes.

Segue a fórmula de cálculo de Manutenção de Componentes:

$$\text{PF_COMPONENTE} = \text{PF_Função_Incluída} \text{ ou } \text{PF_Função_Alterada}$$

O PF Componente também pode ser contado para funções em projetos de desenvolvimento ou de manutenção baseadas apenas em requisitos não funcionais, ou seja, sem contagem de Pontos de Função, por exemplo desenvolvimento de instalador para PGD, desenvolvimento de desinstalador para PGD, disponibilização de Help de Sistema.

As demandas de componentes específicos, tais como: middleware, webservices, instaladores de software devem ser tratadas como uma fronteira a parte, seguindo as orientações do IFPUG, visto que consistem uma demanda específica. As funcionalidades do componente, funções de dados e transacionais, devem ser dimensionadas em Pontos de Função. Além disso, devem ser contados os testes não funcionais associados ao desenvolvimento ou manutenção do componente.

Os serviços de monitoramento, como por exemplo, webservices ou rotinas que verifiquem a disponibilidade de servidores de aplicação ou impressão, de conexão com bancos de dados e etc, deverão ser contados como componentes. Devem ser classificados como Saída Externa, considerando que a informação de disponibilidade ou erro é um dado derivado, que não se encontra armazenado em arquivo. Caso seja necessário desenvolver rotinas independentes para monitorar disponibilidade de recursos diferentes, tais como servidores e portas distintas, estas funções devem ser contadas como processos elementares independentes.

4.22 Projeto de Automação de Testes

Nesta seção são tratados projetos somente com demandas de automação de testes de sistemas legados. Para este tipo de projeto, recomenda-se que seja considerado 25% dos Pontos de Função das funções transacionais cujos testes serão automatizados, conforme fórmula abaixo:

$$PF_AUTOMAÇÃO = PF_NÃO_AJUSTADO \times 0,25$$

Este percentual considera uma produtividade de 12HH/PF e poderá ser alterado conforme histórico de produtividade da equipe de desenvolvimento para este tipo de projeto e ferramenta utilizada para automação.

5. Atividades Sem Contagem de Pontos de Função

Deve-se ressaltar que o processo de desenvolvimento de soluções possui várias atividades que devem ser consideradas como um projeto separado, levando-se em conta as horas realizadas, dentre outras:

- Treinamentos em Tecnologia, Metodologias, Métricas, etc.: encontram-se nesta categoria as demandas de treinamentos em linguagens de programação, ferramentas de gestão, processos, modelos da qualidade, métricas, etc. Estes serviços são executados por consultores do SERPRO, especialistas no assunto em questão. Assim, devem ser consideradas as horas de consultoria para preparação e execução do curso e o custo do deslocamento do instrutor, se for o caso.
- Desenvolvimento de Cursos para EaD: encontram-se nesta categoria as demandas de desenvolvimento de um curso na modalidade de Ensino a Distância (EaD). Estas demandas devem ser remuneradas em horas.
- Mapeamento de Processos de Negócio: encontram-se nesta categoria as demandas de elaboração de documentação contendo o mapeamento de processos de negócio de uma organização ou de parte de uma organização. Estes serviços são executados por consultores do SERPRO, especialistas em BPM (*Business Process Modeling*).
- Elaboração de Plano Diretor de Tecnologia da Informação (PDTI): encontram-se nesta categoria demandas para elaboração de PDTIs para clientes. Estes serviços são executados por consultores do SERPRO, especialistas nas atividades associadas à elaboração de um PDTI.
- Definição de Processo de Desenvolvimento de Soluções: encontram-se nesta categoria demandas para definição de Processos de Software, aderentes às melhores práticas do

CMMI e IN04. Estes serviços são executados por consultores do SERPRO, especialistas nas atividades de processos de software e na customização da ferramenta para criação do site do processo.

- Administração de Dados: este serviço requer uma equipe de ADs com um número de profissionais definido junto ao Cliente, dedicada para atender as demandas associadas à definição e manutenção do modelo de dados de negócio do cliente. Esta equipe fica disponível em horário comercial para atendimento das demandas. Assim, estes serviços não possuem contagem de PF associada. **É importante ressaltar que as atividades de banco de dados associadas ao projeto de desenvolvimento ou de manutenção, por exemplo, preparação de ambiente (testes, homologação, implantação), desempenhadas pelos DBAs da equipe de desenvolvimento, já estão consideradas dentro do projeto de software, não cabendo cobrança adicional.**

- Análise de Solução: Serviço de apoio destinado à análise de regras de negócio implementadas em soluções de TI. Estas demandas não possuem contagem de PF associada.

- Consultoria: Serviço de apoio destinado à análise de regras de negócio a serem implementadas em soluções de TI realizado por consultores especialistas do SERPRO. As demais modalidades de consultoria também podem ser enquadradas neste item, por exemplo, Consultoria em Métricas. Estas demandas não possuem contagem de PF associada.

Outras atividades contidas em um processo de software devem ser gerenciadas dentro do projeto de desenvolvimento ou de manutenção, no entanto o esforço deve ser considerado separadamente da estimativa de esforço derivado da contagem de Pontos de Função. **Estas atividades também devem ser precificadas a parte.** São elas:

- Treinamento para Implantação: são demandas de treinamentos sobre utilização do sistema a ser implantado para os gestores de solução do cliente e usuários. O esforço deste serviço deve ser considerado separadamente da estimativa de esforço derivada da contagem de PF. O preço deste serviço deve ser calculado, levando-se em conta o preço da hora dos consultores do SERPRO que estarão realizando atividades de preparação de treinamento e de instrutoria. Em alguns casos, pode ocorrer também o deslocamento do instrutor, que também deve ser cobrado do cliente. Deve-se ressaltar que este

treinamento para implantação pode ser definido na modalidade de EaD, sendo tratado como um projeto de treinamento a parte. O esforço deste é considerado dentro do projeto de EaD que não faz parte do projeto de desenvolvimento ou manutenção em questão.

- Especificação de Negócio: esta atividade é a primeira atividade a ser executada em uma demanda de projeto de desenvolvimento e/ou de manutenção. O objetivo desta atividade é gerar a Especificação da demanda. O principal produto gerado nesta atividade é o artefato: Documento de Visão do Projeto (DV), que deve ser validado pelo cliente, por meio da assinatura do termo de aceite. Além do DV podem ser gerados outros documentos, tais como: atas de reunião, documento de requisitos não funcionais e glossário da especificação de negócio. O esforço desta atividade deve ser considerado separadamente da estimativa de esforço derivada da contagem de PF.

É importante ressaltar que esta atividade é de responsabilidade dos Analistas de Negócios da empresa contratante, de acordo com a Instrução Normativa (IN 04). Portanto. No entanto, por se tratar de empresa pública, o SERPRO pode ser contratado por alguns clientes para atividades de negócio. Essas atividades antecedem a fase de requisitos – primeira fase do processo de software e devem ser faturadas em horas de consultoria.

Além das atividades descritas acima, o contrato SERPRO –RFB considera as seguintes atividades em homem_hora:

- **Especificação de Requisitos**: esta atividade é a primeira atividade a ser executada em uma demanda de projeto de desenvolvimento e/ou de manutenção. O objetivo desta atividade é gerar a Especificação da demanda. O esforço de realização dessa atividade será remunerado em homem_hora.
 - **Análise de Viabilidade**: Serviço de apoio destinado à análise de arquitetura, processos e regras de negócio implementados em soluções de TI. Estas demandas não possuem contagem de PF associada.
- **Apoio à Especificação**: Serviço de apoio destinado à análise de regras de negócio a serem implementadas em soluções de TI. Estas demandas não possuem contagem de PF associada.
 - **Administração e Modelagem de Dados**: este serviço requer uma equipe de ADs com um número de profissionais definido junto ao Cliente, dedicada para atender as

demandas associadas à definição e manutenção do modelo de dados de negócio do cliente. Estes serviços não possuem contagem de PF associada. A execução destas atividades deve sempre estar contemplado no esforço de especificação. Essas atividades serão remuneradas em homem_hora e devem ser executadas antes da assinatura do Anexo II. É importante ressaltar que as atividades de banco de dados associadas ao projeto de desenvolvimento ou de manutenção, por exemplo, preparação de ambiente (testes, homologação, implantação), desempenhadas pelos DBAs da equipe de desenvolvimento, já estão consideradas dentro da contagem de PF do projeto de software, não cabendo cobrança adicional.

- **Componentes Não Funcionais:** A métrica Pontos de Função tem como propósito dimensionar o tamanho de projetos de software, considerando apenas os requisitos funcionais. No entanto, há demandas em que os requisitos não funcionais possuem alta complexidade, sem funcionalidade associada, as quais devem ser tratadas como uma demanda a parte e remunerados em homem_hora. Caso seja possível dimensionar as funcionalidades do componente, este deve ser dimensionado em Pontos de Função, sendo tratado como uma fronteira a parte, seguindo as orientações do IFPUG. Além disso, devem ser contados os testes não funcionais associados ao desenvolvimento ou manutenção do componente.
- **Suporte à Homologação:** a preparação e a disponibilização do ambiente de homologação (infra estrutura) e a documentação da atividade de homologação - plano de homologação, assim como a correção de eventuais erros estão contempladas no processo de software, remunerado em Pontos de Função. Além disso, a elaboração do Roteiro de Homologação também será tratada como uma atividade remunerada em Pontos de Função. Cabe ressaltar que as atividades de suporte à homologação presencial ou à distância, tais como: esclarecimento de dúvidas sobre os requisitos do sistema, participação de analistas do SERPRO para suporte ao evento de homologação e apoio ao cliente RFB na inclusão de casos no Mantis, podem ser tratados como uma demanda a parte e devem ser remunerados em homem_hora.
- **Treinamento para Implantação:** são demandas de treinamentos presencial ou à distância sobre utilização do sistema a ser implantado para os gestores de solução do cliente e usuários. O esforço deste serviço é considerado separadamente da

estimativa de esforço derivada da contagem de PF. Esta atividade deve ser uma demanda a parte sendo remunerada em homem_hora.

O Roteiro RFB apresenta outras atividades faturadas em homem_hora associadas aos projetos de DataWarehousing.

5.1 Atividades de Projetos de DW sem Contagem de Pontos de Função

Em aderência ao contrato RFB, anexo III.4, as seguintes atividades associadas a Serviços OLAP são remuneradas em homem_hora. Estas atividades estão estimadas em 10.000 horas/mês, de acordo com o item 5.3.

Seguem os serviços, maiores detalhes podem ser obtidos no item 6.1 do anexo III.4 do contrato RFB:

- **Atuar em eventos de Processo OLAP:** atividades de atuação em eventos relativos à qualidade do Processo OLAP, participação em reuniões para levantamento de informações de novos temas de DW e evolução de temas existentes, para garantia de integração dos dados em nível corporativo. Estas atividades devem ocorrer por solicitação da COTEC.
- **Sistematizar Processos:** atividades de definição, desenvolvimento, manutenção e disseminação de aplicações, ferramentas de apoio e padrões a serem seguidos.
- **Divulgar e Garantir o Uso de Padrões:** atividades de divulgação e auditoria do uso do processo OLAP, das normas e padrões vigentes para a construção do DW corporativo da RFB.
- **Manter Dimensões Globais:** atividades de identificação e criação de estrutura para armazenamento de dados das Dimensões Globais do DW Corporativo da RFB.
- **Elaborar Especificação de Requisitos Dimensionais:** atividades de modelagem e descrição das tabelas Fatos, Dimensão e os objetos, apresentando a Constelação, Lista de Estrelas, os Diagramas e a Descrição das tabelas.
- **Apurar a Origem dos Dados:** atividades de apuração e documentação de origem dos dados para as entidades do Processo OLAP.

- **Documentar Artefatos:** atividades de documentação do processo OLAP (tabelas Fato, atributos, métricas, relacionamentos, domínios, lista de valores válidos, regras de transformação), Origem de Dados e Especificação de Requisitos Dimensionais, conforme metodologia de trabalho definida pela RFB.
- **Verificar, Revisar e Validar Modelo Lógico de Dados:** atividades de revisão do Modelo de Dados, verificando a adequação aos padrões definidos, completeza e correção.
- **Elaborar o projeto de ETL:** atividades de documentação do processo de ETL e atualização dos artefatos produzidos neste serviço.
- **Manter Metadados de Produção do DW Receita:** atividades de modelagem construção e manutenção da Base de Metadados com informações necessárias à gerência dos processos de extração, transformação e carga.
- **Apoiar o processo de desenvolvimento de temas do DW:** atividades de apoio ao desenvolvimento de temas em todas as fases de trabalho.
- **Suporte às equipes temáticas:** atividades de suporte e consultoria às diversas equipes temáticas tanto durante o desenvolvimento dos temas de DW quanto à manutenção dos temas.
- **Suporte aos usuários dos temas de DW:** atividades de suporte e consultoria aos usuários dos temas de DW durante a utilização dos mesmos, no que tange à construção de relatórios, utilização de objetos e as diversas aplicações que compõem o serviço do Processo OLAP.
- **Apoiar Homologação, Implantação e Treinamento de temas:** atividades de apoio a homologação, implantação e treinamento de temas, considerando testes referentes à implementação do modelo dimensional de dados.

6. Considerações para Contagem de Pontos de Função

Esta seção apresenta considerações especiais sobre o dimensionamento em Pontos de Função de mudança de requisitos, projetos cancelados e redução de cronograma. E sugere métricas para o dimensionamento de atividades de negócio.

6.1 Considerações sobre Mudança de Requisitos

Em projetos de desenvolvimento e manutenção de software é bastante comum as mudanças de requisitos no decorrer do projeto, conforme o usuário e o desenvolvedor adquirem mais conhecimento sobre o negócio [Sommerville, 2007]. O CPM denomina este fenômeno de Scope Creep. Como os requisitos não podem ser congelados, então temos que gerenciá-los de forma efetiva. Nas estimativas iniciais de tamanho de projetos de desenvolvimento, após a fase de especificação, considerando-se o documento de visão inicial do projeto, recomenda-se utilizar um percentual para evolução de requisitos de 30% a 40%. Nas estimativas, após a fase de requisitos, utilizando-se como insumo as especificações de casos de uso, deve-se considerar um percentual de 20% a 30%. Por exemplo, suponha que após a análise do Documento de Visão de um Projeto, aplicando-se a CEF, foi obtido o tamanho de 200 PFs, então o tamanho estimado do projeto considerado é de 270 PFs ($200 + 35\%$), utilizando-se a premissa de evolução de requisitos em 35%. Esta premissa deve ser documentada.

Uma mudança de requisito gera retrabalho da equipe de desenvolvimento, aumentando assim o esforço e o custo do projeto. Por exemplo, suponha um relatório de clientes em que no final da fase de implementação foi solicitado a exibição de uma nova informação. A equipe de desenvolvimento terá um retrabalho de várias fases do ciclo de vida. Para tratar o dimensionamento das mudanças de requisitos torna-se importante definir a distribuição de esforço pelas fases do projeto, visando definir o valor agregado ao projeto após cada fase do ciclo de vida.

A Tabela 11 apresenta o percentual de esforço tratado no Roteiro de Métricas do SISP do contrato SERPRO –MP e adotado pelo Roteiro SERPRO.

A Tabela 12 apresenta o percentual de esforço para projetos de desenvolvimento, com a fase de especificação de requisitos fora do ciclo de vida do software, considerando o contrato RFB.

Fases do Processo de Desenvolvimento de Software	Percentual de esforço (%)
Engenharia de Requisitos	25%
Design, Arquitetura	10%
Implementação	40%
Testes	15%
Homologação	5%
Implantação	5%

Tabela 11: Distribuição de Esforço de Projeto – Roteiro SISP

Fases do Processo de Desenvolvimento de Software	Percentual de esforço (%)
Gestão de Requisitos	5%
Design, Arquitetura	15%
Implementação	50%
Testes	15%
Homologação	10%
Implantação	5%

Tabela 12: Distribuição de Esforço de Projeto – Roteiro RFB

Para o cliente STN, o percentual das fases não será aplicado na contagem de PF_Retrabalho. Este percentual é definido no contrato SERPRO STN e será aplicado pela SUNAF via sistema SIGOP, conforme definições da SUNAF com o cliente. A fórmula de contagem de PF_Retrabalho para o cliente STN considera o Fator de Impacto de 75% para funcionalidades alteradas. A planilha de Contagem de Pontos de Função da STN automatiza os cálculos.

Os parágrafos seguintes explicam a sistemática de contagem de PF Retrabalho adotada pelo Roteiro SERPRO.

Quando houver mudança de requisitos o SERPRO informará o progresso da construção dos requisitos impactados pela mudança, informando a fase na qual se encontram os requisitos em questão, seu percentual de execução e justificativa do trabalho realizado, de forma a comprovar o percentual informado.

O cálculo de PF do requisito original deve considerar o tamanho funcional do requisito, o Fator de Impacto do Tipo de retrabalho e o percentual de conclusão .

Em caso de mudanças em funcionalidades ou adaptativas em requisitos não funcionais deve ser aplicado um Fator de Impacto de 75% na contagem de PF Retrabalho.

Por exemplo, suponha uma mudança de requisitos no Sistema de Gerenciamento de Projetos, onde o Caso de Uso 2: Relatório de Atividades, precisará de um novo filtro.

Segue a contagem de PF do requisito em questão:

[Caso de Uso 2] Relatório de Atividades – SE – Média – 5 PF

Em uma análise de impacto da mudança foi constatado que este requisito encontra-se na fase de implementação com 20% da mesma concluída.

Desta forma, o tamanho desta mudança deve ser calculado da seguinte maneira:

Macro Atividades	Esforço da Fase
Gestão de Requisitos	25%
Design, Arquitetura	10%
Implementação	8%

Total: 43%

Como a mudança ocorreu na fase de implementação e o percentual de execução foi de 20%, então foi considerado o esforço de $20\% \times 40\% = 8\%$ para a fase de implementação.

O tamanho da mudança de requisitos é de 1,61 PF ($5 \text{ PF} \times 43\% \times 75\%$).

O requisito alterado será considerado na planilha de contagem de PF do projeto em questão, supondo que este será entregue ao cliente sem passar por novas alterações. Desta forma, a contagem de Retrabalho é de 1,61 PF e o PF do requisito entregue ao cliente é de 5 PF. Portanto, o tamanho total deste requisito considerando a mudança é de 6,61 PF.

Supondo uma demanda de inclusão de paginação no Relatório de Atividades, mencionado no exemplo anterior. A contagem de PF Retrabalho seria: $\text{PF_Retrabalho} = 5 \times 43\% \times 75\% = 1,61 \text{ PF}$.

Em caso de mudanças cosméticas em requisitos, na contagem de PF Retrabalho a funcionalidade deve ser contada com 0,6 PF (20% do PF de uma funcionalidade com

complexidade mais baixa). Além disso, deve ser considerado o percentual da fase. Supondo uma demanda de alteração do título do Relatório de Atividades, mencionado no exemplo anterior. A contagem de PF Retrabalho seria: $PF_Retrabalho = 0,6 \times 43\% = 0,26$ PF.

Em caso de exclusão de requisitos, a contagem de PF de retrabalho deve considerar a atividade de construção do requisito em questão e o processo de retirada do requisito da baseline do projeto. A exclusão do requisito deve considerar o percentual de PF_Excluído 40%. Desta forma, a contagem PF_Retrabalho de requisito excluído deve aplicar o fator de 140% na contagem de PF da funcionalidade em questão, multiplicado pelo percentual da fase. Desta forma, suponha a demanda de exclusão do Relatório de Atividades. A contagem de PF Retrabalho seria: $PF_Retrabalho = 5 \times 140\% \times 43\% = 3,01$ PF.

6.2 Considerações sobre Mudança de Requisitos na Fase de Requisitos

No Roteiro RFB, a fase de Engenharia de Requisitos é tratada a parte como Especificação, sendo faturada em homem_hora. No entanto, os indicadores de produtividade do SERPRO, assim como os demais clientes, consideram a fase de requisitos como parte do processo de desenvolvimento de software.

Desta forma, o retrabalho em requisitos terá contagem de PF internamente para o SERPRO e faturada para os clientes que trabalham com o Roteiro SISP. As mudanças de requisitos na fase de requisitos serão dimensionadas em PF, considerando o percentual da fase de até 25%, dependendo da completeza da fase associada à funcionalidade alterada.

Por exemplo, suponha o requisito:

[Caso de Uso 2] Relatório de Atividades – SE – Média – 5 PF

Este requisito teve alteração no final da fase de requisitos, assim será considerado o percentual de completeza de 20%.

Então, o $PF_Retrabalho$ será $5 \text{ PF} \times 0,20 = 1 \text{ PF}$.

Caso o cliente seja a RFB, esta contagem não será encaminhada para o cliente, no entanto, esta será adicionada ao tamanho final do projeto e tamanho dos demais retrabalhos para a atualização do tamanho total do projeto no SGI.

6.3 Mudança de Requisitos na Fase de Homologação – Cliente RFB

Esta seção apresenta orientações associadas à contagem de PF retrabalho para mudanças de requisitos na fase de Homologação para o cliente RFB.

Na homologação de sistemas, em alguns casos, podem ser solicitados pelo cliente muitas mudanças de requisitos, por exemplo: inserir mais uma informação na consulta de declarações, após a conclusão da implementação, posteriormente, implementar ordenação dinâmica, e ainda acrescentar um novo requisito: gerar a consulta de declarações em formato .pdf.

Para o cliente RFB, as alterações de requisitos em eventos de homologação serão limitadas.

Para cada funcionalidade, por evento de homologação, poderá ocorrer um único pedido de alteração de requisitos por tipo (mudanças em funcionalidades ou em requisitos não funcionais, alterações cosméticas e exclusão de requisitos).

O atendimento de quaisquer outros relatos classificados como “Sugestão” (melhorias, novos requisitos, alteração de requisitos, etc) deverá ser objeto de demandas específicas posteriores.

Observe que no exemplo acima, as demandas: incluir ordenação dinâmica na consulta declaração e gerar a consulta de declarações em formato .pdf devem ser tratadas em demandas específicas posteriores. Cabe ressaltar que se o cliente fizer um pedido único solicitando a inclusão da nova informação e ordenação dinâmica na Consulta Declarações, esta deve ser contada apenas uma vez com retrabalho do tipo melhoria.

6.4 Considerações sobre Projetos Cancelados

Em alguns casos, devido às mudanças no ambiente do cliente, uma demanda ou parte de um projeto de desenvolvimento ou manutenção pode ser cancelado pelo cliente. Nestes casos, o tamanho funcional das funcionalidades canceladas será aferido por meio da contagem de Pontos de Função das funcionalidades canceladas e um Fator de Impacto.

O Fator de Impacto é definido com base no percentual de esforço alocado a construção da funcionalidade em questão, observando as tabelas de distribuição de esforço contidas na Seção 6.1 ou alguma diretriz específica de distribuição de esforço do contrato em questão. O Fator de Impacto deve ser aplicado na contagem de Pontos de Função das funcionalidades em questão. É importante ressaltar que em um processo de desenvolvimento incremental uma funcionalidade pode por exemplo estar em fase de requisitos e de testes, porque o plano de testes é construído na fase de requisitos. O Progresso das atividades executadas em cada funcionalidade do projeto deve ser obtido por meio do acompanhamento do plano do projeto.

6.5 Considerações sobre Redução de Cronograma

As estimativas de prazo não são lineares com o tamanho do projeto, assim pretende-se pesquisar mais sobre o melhor tempo de desenvolvimento (onde há uma melhor relação custo x benefício de alocação de recursos e menor prazo de desenvolvimento), dado o tamanho de um projeto específico. Jones [Jones, 2007] propõe uma fórmula para o cálculo do melhor tempo de desenvolvimento.

Alguns projetos, devido à legislação e a outros fatores externos ao SERPRO, possuem um prazo imposto pelo cliente. Se este prazo for igual ou superior ao prazo calculado pela Fórmula de *Capers Jones* (expoente t) ou em caso de projetos pequenos (menores que 100 PF) a um prazo calculado considerando o trabalho da equipe de 8 horas/dia nos dias úteis, então este é tratado como um projeto normal.

No entanto, se o projeto tiver um prazo imposto pelo cliente inferior ao prazo calculado, então se deve considerar o seguinte:

- Redução de prazo de 10%: aumento de esforço de 20% (projetos urgentes)
- Redução de prazo de 20%: aumento de esforço de 50% (projetos críticos)
- Redução de prazo de 25%: aumento de esforço de 70% (projetos de alta criticidade)

Deve-se ressaltar que não é possível uma redução de prazo maior que 25 %, devido aos cálculos de Região Impossível e ainda conforme nos aproximamos da região impossível, o esforço e o custo do projeto aumentam de maneira exponencial. Como os riscos da redução de cronograma também são altos, não é recomendada a redução de

cronograma. Deve-se tentar priorizar funcionalidades trabalhando com o ciclo de vida incremental.

Como o contrato é baseado em Pontos de Função, este aumento de esforço será refletido na contagem de PF. Assim, um aumento de esforço de 20% implica em aumento de 20% na contagem de PF.

6.6 Fator de Criticidade de Solicitação de Serviço

Em função da criticidade e da necessidade de alocação de recursos extras para atendimento da demanda no prazo estipulado pelo cliente, será adotado um Fator de Criticidade de 1,35 (um vírgula trinta e cinco), que deverá ser multiplicado pelo tamanho funcional da demanda considerada crítica, de modo a remunerar adequadamente o aumento do esforço de atendimento. **Este fator é considerado para demandas que devem ser atendidas em finais de semana, feriados e fora do horário comercial. Entende-se como horário comercial o horário de 08:00 às 18:00 h, horário de Brasília.**

O Fator Criticidade deve ser aplicado para projetos menores que 100 PF, onde não se aplica a fórmula de Capers Jones. Para os projetos com o prazo estimado por meio da fórmula de Capers Jones, utilizaremos o Fator Redução de Cronograma, descrito na seção anterior.

7. Contagem de Pontos de Função com Múltiplas Mídias e Dados de Código

Esta seção tem como propósito apresentar as diretrizes de Contagem de Pontos de Função utilizadas no SERPRO em relação ao tema Múltiplas Mídias. Esta abordagem é reconhecida pelo IFPUG. As definições apresentadas têm como base o artigo “*Considerations for Counting with Multiple Media*” Release 1.0 publicado pelo IFPUG [IFPUG, 2009]. Também é abordado neste capítulo algumas diretrizes para mensuração de Dados de Código e Code Data.

Considerando-se a contagem de PF de funcionalidades entregues em mais de uma mídia, a aplicação das regras de contagem de Pontos de Função 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 abordagens *single instance* e *multiple instance* para a aplicação das regras definidas no CPM. A seguir são descritos os termos comuns definidos pelo IFPUG [IFPUG, 2009]:

- **Canal:** também refere-se a mídia. Múltiplos canais é sinônimo de múltiplas mídias.
- **Mídia:** descreve a maneira que os dados ou informações se movimentam para dentro e para fora de uma fronteira de 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 a consulta de extrato bancário via terminal do banco.

- **Multi-Mídia:** 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 o vídeo.
- **Abordagem *Single Instance*:** esta abordagem não reconhece que a mídia utilizada na entrega da função transacional é uma característica de diferenciação na identificação da unicidade da função transacional. Se duas funções entregam a mesma funcionalidade usando mídias diferentes, elas são consideradas a mesma funcionalidade em uma contagem de Pontos de Função.
- **Abordagem *Multiple Instance*:** esta abordagem especifica que o tamanho funcional é obtido no contexto de objetivo da contagem, permitindo uma função de negócio ser reconhecida no contexto das mídias que são requisitadas para a funcionalidade ser entregue. A abordagem *multiple instance* reconhece que a mídia para entrega constitui uma característica de diferenciação na identificação da unicidade da função transacional.

Os cenários descritos nas seções seguintes não representam uma lista completa de situações de múltiplas mídias. O entendimento destes exemplos facilitará o entendimento de outros cenários envolvendo múltiplas mídias. Este Roteiro deve ser atualizado considerando a publicação de novas diretrizes do IFPUG e novos cenários que emergirão nas contagens de PFs dos projetos dos clientes do SERPRO.

7.1 Cenário 1: Mesmos dados apresentados em tela e impressos

Neste cenário, uma aplicação apresenta uma informação em uma consulta em tela. A mesma informação pode ser impressa caso requisitado pelo usuário na tela em questão.

Nesses casos, utiliza-se a abordagem *single instance*, considerando que dados idênticos sendo apresentados em tela e relatório impresso devem ser contados como uma única função. Caso as lógicas de processamento da consulta em tela e do relatório em papel sejam distintas, o processo elementar não é único e portanto a funcionalidade será contada duas vezes.

7.2 Cenário 2: Mesmos dados de saída como dados em arquivo e relatório impresso

Uma aplicação grava dados em um arquivo de saída e imprime um relatório com informações idênticas as gravadas no arquivo.

Nesses casos, utiliza-se a abordagem *single instance* considerando que os dados impressos e os dados apresentados no arquivo de saída sejam idênticos e que a ferramenta de desenvolvimento apoie a geração dessas múltiplas saídas. Assim, apenas uma funcionalidade será incluída na contagem de Pontos de Função. Caso as lógicas de processamento da geração do arquivo de saída e do relatório em papel sejam distintas, o processo elementar não é único e portanto a funcionalidade será contada duas vezes. E ainda, se a geração das múltiplas saídas não seguirem o padrão da ferramenta de desenvolvimento e tiverem que ser customizadas para o cliente, então será utilizada a abordagem *multiple instance*. Caso seja necessária a geração de mais de um tipo de arquivo, cada tipo de arquivo gerado será considerada uma funcionalidade distinta.

Observe que a abordagem *multiple instance* considera que dados idênticos estão sendo entregues em mais de um tipo de mídia e a contagem de PF incluirá todas as instâncias de tipos de mídia. Neste cenário, duas funções são contadas – geração arquivo e apresentação dos dados impressos.

7.3 Cenário 3: Mesmos dados de entrada batch e on-line

Uma informação pode ser carregada na aplicação por meio de dois métodos: arquivo batch e entrada on-line. O processamento do arquivo batch executa validações durante o processamento. O processamento on-line também executa validações das informações.

Deve-se utilizar a abordagem *multiple instance* que conta duas funcionalidades: a entrada de dados batch e a entrada de dados on-line. Geralmente, a lógica de processamento utilizada nas validações em modo batch é diferente da lógica de processamento das validações nas entradas de dados on-line. Portanto, o SERPRO contará duas funcionalidades.

7.4 Cenário 4: Múltiplos canais de entrega da mesma funcionalidade

Uma funcionalidade deve ser disponibilizada em múltiplos canais, por exemplo consulta de dados em página Web e consulta de dados no telefone celular.

A abordagem *single instance* conta apenas uma funcionalidade. Deve-se utilizar a abordagem *multiple instance* que conta duas funcionalidades: a consulta de dados na Web e a consulta de dados via celular. Considera-se que a funcionalidade é desenvolvida duas vezes para os dois canais. Algumas vezes, são até projetos de desenvolvimento distintos, um projeto relativo ao sistema Web e outro para o sistema via celular. Portanto, serão contadas duas funcionalidades.

7.5 Cenário 5: Relatórios em Múltiplos Formatos

Um relatório deve ser entregue em diferentes formatos, por exemplo em um arquivo *html* e um formato de valores separados por vírgula.

Nestes casos, conforme sugerido na abordagem *multiple instance*, considera-se a ferramenta utilizada na geração dos relatórios. Se a equipe de desenvolvimento precisar desenvolver o relatório nos dois formatos na ferramenta em questão, serão contadas duas funcionalidades. Porque, a lógica de processamento de análise de condições para verificar quais são aplicáveis é identificada. No entanto, se a ferramenta de desenvolvimento suportar um gerador de relatórios que o usuário visualize o relatório em tela e o gerador permita ao usuário imprimir o relatório, salvar em *html* ou salvar no formato de valores separados por vírgula, então o SERPRO contará apenas uma vez, observando que a funcionalidade será da ferramenta e não da aplicação.

7.6 Cenário 6: Funcionalidades Fornecidas via Aplicação e Webservice

Algumas funcionalidades são fornecidas pela aplicação via interface com o usuário final e também via Web Service. Estas funcionalidades devem ser contadas duas vezes, considerando a abordagem *multiple instance*, visto que há esforço para o desenvolvimento das duas funcionalidades, assim como são entregues duas funcionalidades para o usuário.

7.7 Cenário 7: Plataforma MOBILE - Android e IOS

Para a construção de um determinado sistema foi solicitado que fossem entregues, em um único projeto, versões para plataformas distintas (ex: *Android* e *IOS*).

Apesar de ambas serem direcionadas a dispositivos móveis (*mobile*), as plataformas *Android* e *IOS* são completamente diferentes, necessitando que uma mesma funcionalidade seja construída duas vezes. Desta forma, deve-se utilizar a abordagem *multiple instance*, que conta transações distintas. Analisando a construção deste produto em projetos separados (um para o desenvolvimento *Android* e um para o *IOS*), cada projeto teria sua contagem realizada considerando apenas a funcionalidade entregue pelo projeto.

7.8 Dimensionamento de Dados de Código

As Tabelas com atributos de Código e Descrição devem ser analisadas com muito cuidado. Caso estas tabelas não sejam mantidas pela aplicação, o que geralmente acontece, estas são classificadas como Code Data (Dados de Código) e portanto não são contadas. Os Code Data são implementações de requisitos técnicos, requisitos não funcionais para melhorar a manutenibilidade da aplicação, não sendo contemplados em uma Contagem de Pontos de Função.

Entretanto, as Tabelas com atributos de código e descrição que armazenam dados de negócio, ou seja possuem manutenção de dados por processos elementares da aplicação, serão tratadas como Entidades de Negócio e contadas como Arquivos Lógicos Internos.

Desta forma, as Tabelas com atributos de código e descrição serão tratadas da seguinte maneira:

- **Dados de Código: Tabelas Código:** Descrição ou qualquer outra entidade com dados estáticos, sem funcionalidades de manutenção requisitadas pela RFB. A manutenção de tais entidades de dados é por meio de apuração especial. Estas Tabelas não são contadas.

Roteiro de Métricas SERPRO

- **Dados de Negócio: Tabelas Código:** Descrição ou qualquer outra entidade com dados dinâmicos, com funcionalidades de manutenção requisitadas pela RFB, por meio de Especificações de Casos de Uso aprovadas. Estas Tabelas são contadas como Arquivo Lógico Interno ou Registro Lógico, observando as regras de contagem de Funções de dados do CPM. As funcionalidades de manutenção e de consulta a estes dados, descritas no Documento de Requisitos aprovado, também serão contadas de acordo com as regras de contagem de Funções transacionais do CPM.

8. Guia de Contagem de Pontos de Função para Projetos de Data Warehouse

Esta seção tem como propósito apresentar um roteiro contagem de Pontos de Função para projetos de desenvolvimento e manutenção de Data Warehouse (DW) aderente ao Manual de Práticas de Contagem (CPM 4.3) e definir os tipos de projetos de manutenção e uma sistemática para dimensionar o tamanho de tais projetos, utilizando a métrica Pontos de Função Não Ajustados.

8.1. Definição de Tipo de Contagem

O CPM menciona três tipos de contagem: Contagem de Desenvolvimento, Contagem de Melhoria e Contagem de Aplicação. Este último tipo tem como objetivo obter o tamanho funcional em Pontos de Função de aplicações implantadas.

Neste Roteiro, considerando o contexto de contagem de Projetos de Data Warehouse para estimativas ou contratos, consideram-se os seguintes tipos de Contagem de Pontos de Função:

- Projeto de Desenvolvimento: mede a funcionalidade entregue ao usuário na primeira instalação do software, quando o projeto estiver completo. Este tipo de projeto considera as novas funcionalidades e as funcionalidades de conversão de dados.
- Projeto de Manutenção: 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. Nesta categoria encontram-se os Projetos de Melhoria, Projetos de Manutenção Corretiva, Projetos de Manutenção Cosméticas – Adaptação de Interface, os Projetos de Manutenção Adaptativa e as Apurações Especiais.

8.2. Identificação de Propósito, Escopo e Fronteira

Uma contagem de Pontos de Função se inicia pela definição do propósito da contagem. O propósito de uma contagem de Pontos de Função é fornecer uma resposta a um problema de negócio. Neste contexto, o propósito da contagem é a aferição do tamanho funcional das funcionalidades desenvolvidas em um Projeto de desenvolvimento de Data Warehouse ou das funcionalidades impactadas por um projeto de manutenção de um Data Warehouse.

O Escopo da Contagem define quais são as funcionalidades objeto de determinada contagem. O escopo da contagem é determinado pelo propósito da contagem, e identifica os sistemas, as aplicações ou seus componentes que serão dimensionados. Um escopo de contagem pode conter mais de uma aplicação. No entanto, a contagem de Pontos de Função é realizada separadamente considerando cada fronteira de aplicação. No contexto de Contagem de PF de Data Warehouse, o escopo da contagem abrange: o Projeto de Desenvolvimento ou Manutenção do Data Warehouse e o Projeto de Melhoria nas aplicações que irão fornecer os dados para o Data Warehouse, ou seja, o DW é uma fronteira única e as aplicações de origem são outras fronteiras.

A fronteira da aplicação indica o limite lógico entre o sistema que está sendo medido e o usuário. É uma interface conceitual entre a aplicação (interno) e o mundo do usuário (externo). Depende da visão do usuário da aplicação. Uma visão de usuário representa uma descrição formal das necessidades de negócio do usuário na linguagem do usuário. No caso de projetos de Data Warehouse, o projeto é considerado uma fronteira e as aplicações de origem, de onde são extraídos os dados do DW, são consideradas outras fronteiras distintas.

Deve-se ressaltar que a Contagem de Pontos de Função está associada a um projeto, considerando todo o ciclo de vida de desenvolvimento ou de manutenção do projeto em questão. No entanto, em projetos de Data Warehouse existem atividades de negócios que antecedem o processo de desenvolvimento. Estas atividades são denominadas de “Atividades de Modelagem de Negócio para Solução de Apoio à Decisão” e estão fora do escopo da contagem. Estas atividades de definição do modelo conceitual dos dados preliminar e descrição das funcionalidades de negócio, documentadas em um Documento de Visão, requerem atuação de consultores de negócio especialistas do SERPRO. Portanto, essas atividades são consideradas como Consultoria e a métrica utilizada é a de Homem_Hora. A precificação destas é realizada por meio do insumo Consultoria descrito na Política de Preços do SERPRO. As estimativas de tamanho e prazo dos projetos são realizadas após a definição dos processos de negócio da aplicação.

8.3. Identificação de Processos Elementares

Um processo elementar é a menor unidade de atividade identificada pelo usuário. O processo elementar deve ser auto-contido e deixar a aplicação em um estado consistente. Os processos elementares são classificados em Entrada Externa (EE), Consulta Externa (CE) e Saída Externa (SE).

8.4. Entradas Externas em Projetos de Data Warehouse

Entrada Externa é um processo elementar que processa dados ou informações de controles vindos de fora da fronteira da aplicação. A sua principal intenção é manter um Arquivo Lógico Interno (ALI) e/ou alterar o comportamento do sistema.

Em Projetos de Data Warehouse geralmente existem funcionalidades de cargas de dados nas tabelas do DW. Estas tabelas são denominadas de tabelas fato e tabelas dimensão em um modelo multidimensional em um diagrama estrela. As funcionalidades de carga de dados são classificadas como Entradas Externas.

Uma situação a considerar trata 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 de Pontos de Função. No entanto, é importante ressaltar que em alguns casos, o usuário deseja realizar consultas e emitir relatórios diretamente os dados da DSA. Nesses casos, as funcionalidades da DSA serão consideradas na contagem de Pontos de Função. Os dados da DSA serão contados como Arquivos Lógicos Internos. As cargas de dados serão contadas como Entradas Externas e as consultas e relatórios serão contados como Consultas Externas ou Saídas Externas.

As funcionalidades associadas às cargas de dados inicial (full) e incremental (delta) em Tabelas Fato, Dimensão ou Agregação (Estrela de Performance) são contadas como Entradas Externas.

8.5. Consultas em Projetos de Data Warehouse

Existem dois conceitos definidos no CPM para Consultas:

A Saída Externa é um processo elementar que envia dados para fora da fronteira da aplicação, e sua principal intenção é apresentar informação ao usuário por meio de lógica de processamento adicional à recuperação de dados ou informações de controle. Sua lógica de processamento deve conter no mínimo uma fórmula matemática ou cálculo, ou ainda criar dado derivado. Esse processo pode manter um ou vários arquivos lógicos ou alterar o comportamento do sistema.

A Consulta Externa é um processo elementar que envia dados para fora da fronteira da aplicação, e sua principal intenção é apresentar informação ao usuário por meio da recuperação de dados ou informações de controle. Sua lógica de processamento não envolve fórmula matemática, nem cálculo, não cria dado derivado, nenhum arquivo lógico é mantido durante o processo e o comportamento do sistema também não é alterado.

Freqüentemente, em projetos 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 em um projeto de melhoria. Observe que estas funcionalidades não fazem parte da fronteira da aplicação de DW. No entanto, fazem parte do escopo da contagem do projeto de DW.

Em alguns casos, o DW acessa diretamente o Banco de Dados das aplicações de origem, por meio de ferramentas. Observe que nestes casos 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.

Em Aplicações de Data Warehouse existem requisitos para geração de relatórios (relatório ou consulta de bancada) ou gráficos (painéis, dashboards) usando as ferramentas. Cada relatório ou gráfico requisitado pelo usuário e implementado pela equipe de desenvolvimento será contado como Consulta ou Saída Externa, devendo-se ainda verificar se atendem os critérios de determinação da unicidade do CPM.

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.

A geração do Contexto de Análise também deve ser contada como Saída Externa.

8.6. Identificação de Funções de Dados em Projetos de Datawarehouse

As funções de dados estão associadas aos requisitos de dados internos e externos. As Funções de dados são classificadas em Arquivos Lógicos Internos (ALI) ou Arquivos de Interface Externa (AIE).

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

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

Em um modelo de dados multi dimensional, Esquema Estrela, são reconhecidos dois tipos de entidades: Tabelas Fato e Tabelas Dimensão.

Deve-se observar a quantidade de níveis hierárquicos na Dimensão e contar um registro lógico para cada nível hierárquico. Por exemplo, suponha que a dimensão setor sempre está relacionada à dimensão departamento, ou seja não se associa diretamente à fato empregado. Desta forma, a dimensão setor é uma hierarquia da dimensão Departamento e deve ser contada como um Registro Lógico do ALI: Departamento/Setor.

Caso não existam níveis hierárquicos nem subgrupos de dados dentro da dimensão, considere apenas um Registro Lógico para a Dimensão.

Conta-se uma Entrada Externa para cada tipo de carga de informações em uma tabela Dimensão. Deve-se ressaltar que a carga inicial de dados nas tabelas Dimensão também deve ser contada separadamente como uma Entrada Externa. Caso exista 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. Nesses casos, 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 Arquivos Lógicos Internos. Deve ser contada uma Entrada Externa para a carga de dados na Tabela Fato, e, de modo análogo ao da Dimensão, a carga inicial de dados é contada separadamente como uma Entrada Externa.

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, observando o critério de unicidade das Entradas Externas do Manual de Práticas de Contagem (CPM). Desta forma, se as Entradas Externas tiverem os mesmos Arquivos Referenciados, os mesmos Tipos de Dados e a mesma lógica de processamento, então deve-se contar apenas uma vez.

Caso exista leitura de dados de outras aplicações para validação de informações durante as cargas de dados, essas tabelas que são Arquivos Lógicos Internos de outras aplicações e são apenas lidas pelo DW, serão contadas como Arquivos de Interface Externa.

As Dimensões de outros DWs referenciadas na geração do Contexto de Análise também são consideradas como Arquivos de Interface Externa.

Algumas vezes, o usuário requer a combinação de tabelas Fatos gerando uma outra Tabela Fato ou uma estrutura de agregação ou estrela de *performance*, visando

apoiar a geração de consultas. Em alguns casos, 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 Datawarehouse da RFB também pode ser encontrada uma estrutura de agregação de Tabelas Dimensão denominada Placa. A Placa deve ser contada como um Arquivo Lógico Interno. A carga de dados é contada como uma Entrada Externa.

Caso exista uma extração de dados específica para as estruturas de agregação, seja ela do tipo estrela de performance ou placa. A extração de dados deve ser contada como Consulta Externa ou Saída Externa.

Em alguns casos, 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. Nestes casos 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.

8.6.1 Conceito de Mudança Estrutural em ALI/AIE para projetos de Melhoria

Sempre que houver uma mudança estrutural em um ALI/AIE este deverá ser contado como função de dados alterada em um projeto de melhoria.

Mudança estrutural é toda inclusão ou exclusão de campo de um arquivo lógico, ou alteração de suas características (ex. alteração do tamanho do campo e alteração de tipo de campo – numérico para alfanumérico). Simples alterações de valores válidos em um ALI não serão consideradas mudanças estruturais.

Exemplo: Se um campo aceitava os valores 1,2 e 3 e passa a aceitar também o 4, isto não será considerado uma mudança estrutural. Portanto, não é contado como um projeto de melhoria. Estes casos serão contados como manutenções adaptativas em requisitos não funcionais. Caso essas mudanças estruturais reflitam em mudança de lógica de

processamento, tal como mudança em validações nas funcionalidades, estas funcionalidades que sofrerem impactos serão consideradas PF_Alterado em um projeto de melhoria. Caso necessário, podem ser realizadas apurações especiais para atualização da base de dados.

8.7 Tabelas de Visualização – Geração de Cubos ou Contexto de Análise ou Universo

Esse tipo de tabela, normalmente, é utilizada para consumo por outras aplicações ou pelo próprio Datamart (temas para a RFB). 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 Tabela Fato e cada Tabela Dimensão, identificada como Arquivo Lógico Interno, e os tipos de dados serão os atributos de todos os Arquivos Referenciados (Tabela Fato e Dimensão) e as métricas associadas. As Dimensões de outros DWs referenciadas na geração do Contexto de Análise, contadas como Arquivos de Interface Externa, também devem ser consideradas como Arquivos Referenciados da SE: Geração de Contexto de Análise.

Os projetos de manutenção evolutiva que possuem como requisitos alteração de métricas existentes ou criação de novas métricas em uma Tabela Fato deve ser contada a funcionalidade de Geração de Contexto de Análise como PF_Alterado.

8.8. 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, por exemplo guardar 60 meses de dados históricos. Esta função de limpeza é contada como uma Entrada Externa.

Os dados utilizados para gerenciar o DW podem ser, por exemplo: 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.

8.9 Demandas Típicas de Manutenção Evolutiva em DW

Esta seção apresenta demandas típicas de projetos de Manutenção Evolutiva para DataWarehouse.

8.9.1 Criação de Métricas (fórmulas)

As métricas (fórmulas) são atributos lógicos associados às Tabelas Fatos ou Tabela Dimensão, e são criadas com a geração do contexto de análise da Tabela Fato. Assim, caso o usuário solicite a criação de uma nova métrica, a contagem de PF será a seguinte:

SE: Geração do Contexto de Análise da Tabela – Fato

Arquivos Referenciados: Tabela Fato e suas tabelas dimensões.

Tipos de Dados: todos os campos da Tabela Fato, Dimensão e Métricas.

8.9.2 Alteração de Campos em tabelas Fato e Dimensão

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

CE/SE: Extração de Dados do Sistema de Origem

ALI: Tabela Fato

EE: Atualização de Dados da Tabela Fato

EE: Carga de Dados na Tabela Fato

SE: Geração de Contexto de Análise

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

CE/SE: Extração de Dados do Sistema de Origem

ALI: Tabela Dimensão

EE: Atualização de Dados da Tabela Dimensão

EE: Carga de Dados na Tabela Dimensão

SE: Geração de Contexto de Análise

Devem ser contadas todas as gerações de contexto de análise que referenciem a tabela Dimensão alterada.

As funções acima serão classificadas como PF_ALTERADO.

8.9.3 Criação, Configuração e Disponibilização de um Filtro

Quando for solicitada a criação de um filtro de segurança para um DW, não serão recontados os relatórios de bancada, a demanda completa será contada como 3 Pontos de Função.

PF_Filtro de Segurança = **3 PONTOS DE FUNÇÃO por demanda**
(limitada a 10 filtros)

Quando for solicitada a criação de um filtro de relatório para um DW, não serão recontados os relatórios de bancada, a demanda completa será contada como 3 Pontos de Função.

PF_Filtro de Relatório = **3 PONTOS DE FUNÇÃO por demanda**

8.10 Alteração de Dados de Dimensões Estáticas

A inclusão ou alteração de dados nas dimensões estáticas, em projetos de manutenção, serão contadas da seguinte forma:

PF_Dimensão_Estática= 0,3 PF x Qtd Dimensões Alteradas

8.11 Contagem de Metadados: Descrição de Atributos, Métricas e Pastas

As demandas para descrever atributos, métricas e pastas relacionadas a uma tabela fato ou atributos e pastas associados a uma tabela Dimensão serão contadas como PF_METADADOS, onde PF_METADADOS é igual à 0,2 x contagem da tabela Fato ou Dimensão. Ainda que uma dimensão tenha mais de um registro lógico, a contagem desse ALI será feita apenas uma vez.

$$PF_Metadados = 0,2 \times \text{Contagem PF do ALI}$$

Caso uma dimensão seja dados de código, conta-se 0,3 pontos de função.

$$PF_Metadados_Dimensão_Estática = 0,3 \text{ PF por Dimensão}$$

8.12 Reorganização da bancada (reposicionamento de itens)

A demanda reorganização pode estar associada à qualquer métrica, atributo ou filtro. As demandas para reorganização de objetos da bancada serão contadas como 0,6 pontos de função.

$$PF_Reorganização_de_Bancada = 0,6 \text{ PF} \times \text{Qtd de Itens Reorganizados}$$

8.13 Evolução de Páginas Estáticas em Data Warehouse

A contagem de páginas estáticas será contada conforme a seção manutenção de páginas estáticas em Internet, Intranet e Portal.

8.14 Massa de Dados para Homologação em DW

Em alguns projetos de Datawarehouse podem ocorrer demandas de geração de massa de testes para homologação. Estas atividades consistem em carga de dados no Banco de Dados do Ambiente de Homologação. A contagem de Pontos de Função deve ser realizada da seguinte maneira: Contar uma Entrada Externa para cada Tabela Fato, Dimensão ou Agregação carregada. A EE terá sempre apenas um Arquivo Referenciado (o Arquivo Lógico carregado) e os Tipos de Dados (TDs) serão os campos das tabelas carregados. Segue a fórmula de cálculo:

$$PF_Massa_Testes = PF_Incluído_Carga_Dados$$

Este Roteiro não contempla retrabalho para carga de Massa de Dados para Homologação. Após realização da carga de dados no Banco de Dados do Ambiente de Homologação, caso o cliente solicite mudanças nos requisitos do projeto, tornando

necessário implementar e executar novas cargas, elas deverão ser consideradas na planilha de contagem, justificando no campo "Observações" da aba "Sumário" o fato de existirem duas cargas para o mesmo ALI na contagem.

8.15. Considerações sobre Estimativas de Data Warehouse

De posse do documento de visão do projeto, devem ser contadas as Tabelas Fato e Tabelas Dimensão. Caso não seja 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 informação. Para cada Estrela, deve-se considerar uma Saída Externa Complexa, considerando a geração do Contexto de Análise. Geralmente, a geração de Contexto de Análise é uma Saída Externa de complexidade Alta. Caso, os relatórios estejam definidos nesta fase, estes devem ser contados como Saídas Externas médias. Caso contrário, não serão contados. Caso seja possível identificar os Tipos de Dados e Arquivos Referenciados das Entradas Externas e Saídas Externas, deve-se avaliar a complexidade funcional com base nas informações identificadas.

O método utilizado para estimar o prazo dos projetos (T_d) é baseado na fórmula de Capers Jones [Jones, 2007]. A fórmula de Capers Jones estima o prazo, baseando-se no tamanho do projeto em Pontos de Função, da seguinte maneira:

$$T_d = V^t$$

Onde:

T_d: prazo de desenvolvimento em meses

V: tamanho do projeto em Pontos de Função

t: o expoente t para projetos de **Data Warehouse** é **0,39**, devido a estes projetos serem considerados Sistemas Gerenciais

Esta fórmula é utilizada para projetos com tamanho a partir de 100 PF. Projetos de manutenção de Data Warehouse com menos de 100 PF terão seu prazo calculado com base no cronograma de atividades do projeto.

9. Contagem de Pontos de Função de Portais Zope/Plone

A ferramenta Zope Plone de desenvolvimento de Portais possui uma produtividade relativamente alta. Esta produtividade é refletida na estimativa de esforço das demandas e na política de preços do SERPRO. No entanto, a contagem de Pontos de Função leva em consideração o tamanho funcional das funcionalidades requisitadas e recebidas pelo usuário.

Seguem as orientações para contagem de Pontos de Função de Portais Zope/Plone:

- Funcionalidades desenvolvidas ou mantidas pela SERPRO serão contadas de acordo com as diretrizes do CPM 4.3.
- Funcionalidades da ferramenta Zope/Plone **APENAS** disponibilizadas para o usuário não serão contadas. O Documento de Visão deverá registrar que estas funcionalidades do Zope/Plone serão apenas disponibilizadas. Não serão construídos casos de uso para estas funcionalidades. Elas podem ser registradas no documento de requisitos não funcionais. Trata-se de um requisito técnico.
- As funcionalidades da ferramenta Zope/Plone customizadas para o usuário serão contadas de acordo com as diretrizes do CPM 4.3. Observe que estas funcionalidades são requisitadas e recebidas pelo usuário. As facilidades da ferramenta Zope/Plone não são tratadas em uma contagem de Pontos de Função. Estas funcionalidades devem ser documentadas na Especificação de Casos de Uso.
- Os Tipos de Dados das funções transacionais são apenas aqueles que atravessam a fronteira da aplicação. Desta forma, uma atualização de dados de Notícia que referencie os arquivos lógicos Notícias e Log de Auditoria, deve ser contada como uma Entrada Externa com dois Arquivos Referenciados. No entanto, os Tipos de Dados são apenas os dados de notícias atravessam a fronteira da aplicação. Os dados de Log não devem ser contados.
- A implementação de funcionalidades de portais Zope/Plone geralmente acessam o arquivo de usuários. No entanto, na visão do usuário, os dados de usuário já foram passados para a aplicação no momento do Controle de

Roteiro de Métricas SERPRO

Acesso. A referência aos dados de usuário , por exemplo, pela funcionalidade Incluir Banner é uma implementação de um requisito técnico de segurança. Desta forma, o ALI usuário não pode ser contado como Arquivo Referenciado na funcionalidade incluir Banner. Os tipos de dados de usuário que não atravessam a fronteira da aplicação também não devem ser contados.

- Em requisitos de manutenção de cadastros, por exemplo manter notícias, deve-se observar as funcionalidades entregues para todos os perfis de acesso. Por exemplo, a consulta lista de documentos para download é idêntica para o perfil de acesso gestor e usuário comum. Então, contar apenas uma vez. No entanto, a consulta de notícias do usuário gestor, apresenta dados referentes ao histórico de publicação de notícias, tais como: autor, revisor, publicador, data da revisão, e a consulta de notícia do usuário não apresenta tais informações. Assim, devem ser contadas duas funcionalidades distintas: CE/SE: Consulta de Notícias – Usuário; CE/SE: Consulta de Notícia – Gestor.

10. Orientações para Sistemas RFB

Esta seção tem como propósito apresentar orientações específicas para contagem de Pontos de Função de Sistemas da RFB. Seguem algumas diretrizes a serem consideradas nas contagem de PF:

- **Perfis de Acesso e Funcionalidades:** alguns sistemas da RFB possuem vários tipos de usuários com perfis de acesso distintos. Para estes perfis de acesso são disponibilizadas funcionalidades distintas ou não. Na contagem de Pontos de Função deve-se contar todas as funcionalidades que a aplicação disponibiliza para os perfis de acesso de usuários. No entanto, deve-se atentar para não contar duas vezes funcionalidades iguais com base nos critérios de unicidade do manual de práticas de contagem (CPM), a saber: A funcionalidade é única se ela tiver arquivos referenciados distintos, ou tipos de dados distintos ou lógica de processamento distinta. Por exemplo: uma função alterar senha disponibilizada para vários perfis deve ser contada apenas uma vez, supondo que esta se comporte da mesma maneira para todos os perfis. Já a funcionalidade alterar aluno de um sistema de treinamento não é igual para o perfil aluno e o perfil professor. O perfil aluno pode alterar seus dados cadastrais e o perfil professor pode alterar a informação de notas das provas. Desta forma, alterar aluno – perfil aluno e alterar professor- perfil professor devem ser contadas duas vezes.

- **Entidades Lógicas x Entidades Físicas:** Em sistemas da RFB, uma entidade física pode corresponder a várias entidades lógicas, especialmente em sistemas que usam o Banco de Dados ADABAS. A contagem de Arquivos Lógicos leva em consideração os arquivos lógicos. Desta forma, uma tabela física pode ser contada como vários Arquivos Lógicos Internos. Por outro lado, um Arquivo Físico pode ser contado como parte de um Arquivo Lógico Interno – Registro Lógico (implementação de entidades fracas) ou até mesmo não ser contado (implementação de Code Data).

- **Contagem de PF de Tabelas de Histórico:** Geralmente, as tabelas de histórico são contadas como Registros Lógicos de um Arquivo Lógico Interno, por serem consideradas entidades fracas dos mesmos. No entanto, no SIEF assim como em outros sistemas da RFB, os registros das tabelas de histórico continuam existindo mesmo quando o registro associado da entidade principal é excluído. Desta forma, o histórico é uma entidade independente e deve ser contado como um Arquivo Lógico Interno separado.

- **Identificação de Fronteiras:** A definição da fronteira da contagem deve ser realizada antes da contagem de Pontos de Função propriamente dita. Cabe destacar que cada fronteira de aplicação deve ter sua contagem de Pontos de Função documentada em uma planilha de contagem distinta. **Não deve ser utilizada a mesma planilha de contagem para várias fronteiras de aplicações.**

- **Contagem PF Retrabalho:** A contagem de Pontos de Função de Retrabalho deve ser realizada na planilha de contagem de PF de retrabalho separadamente da contagem de PF do projeto. Esta contagem estará relacionada à uma demanda de mudança de requisitos documentada em um Relatório de Análise de Impacto (RAI). O tamanho final do projeto será a contagem de PF final do projeto (contagem PF de encerramento) adicionada às contagens de PF Retrabalho.

- **Atualização ou Criação de Componentes Internos Reusáveis:** As funções associadas à rotinas, arquivos de configuração ou páginas de estilo com reuso interno pelo sistema sendo contado podem ter contagem de Pontos de Função de acordo com a seção “Desenvolvimento e Manutenção de Componentes Internos Reusáveis”.

Em caso de componentes com requisitos não funcionais complexos, estes devem ser tratados como uma fronteira separada e uma demanda distinta. Esta demanda deve ser classificada como Projeto de Desenvolvimento de Componente e pode ser faturada em homem_hora com negociação. Para o SGI, a demanda será considerada do tipo de linguagem Componentes, apenas em casos de exceção.

- **Documentação de Contagem de Pontos de Função:** As funcionalidades identificadas devem ser descritas, incluindo a rastreabilidade para o requisito de origem. Também podem ser descritas: justificativa de contagem e observações caso necessário. Para os Arquivos de Interface Externa deve ser identificada a fronteira da aplicação de origem. Por exemplo, AIE: Municípios (TOM). Os Arquivos Referenciados e Registros Lógicos devem ser descritos na planilha de contagem. Os Tipos de Dados devem ser descritos apenas para justificar a complexidade. Por exemplo, para um Relatório com 4 Arquivos Referenciados, só precisarão ser descritos 6 tipos de dados para justificar a complexidade alta. Caso haja disponibilidade do estimador, podem ser descritos todos os Tipo de Dados da funcionalidade.

- **Contagem de Pontos de Função de GRS:** A contagem de Pontos de Função de Gerenciamento de Riscos de Segurança é calculada com 2% da Contagem de PF de

Referência. Cabe ressaltar que esta contagem deve ser contemplada também na contagem de PF final do projeto.

- **Contagem de Pontos de Função de Manutenção:** Em algumas demandas podem ocorrer dois ou mais tipos de manutenção em uma mesma funcionalidade. Por exemplo, acrescentar um campo em uma consulta e alterar o texto estático do cabeçalho da consulta e melhorar o tempo de resposta da consulta em questão. Observe que foram solicitadas três tipos de manutenção em uma mesma funcionalidade: Melhoria – acrescentar campo; Manutenção em Interface – atualização texto estático da tela; Manutenção adaptativa em requisitos não funcionais – melhoria de performance. Como estas manutenções foram pedidas em uma mesma demanda, a funcionalidade poderá ser contada APENAS UMA VEZ em uma demanda. Desta forma, o tipo de demanda considerado será aquele que possui a maior contagem de Pontos de Função. Neste caso será classificado como Melhoria.

10.1 Definição Fronteiras de Sistemas RFB

Em geral, os novos exercícios dos PGDs: DMED, DIRF, DIRPF, DBF devem ser considerados um projeto de manutenção. No entanto, quando há reconstrução deve ser considerado um novo projeto de desenvolvimento. Para o PGD DBF não há uma versão por Exercício/Ano Calendário. Uma mesma versão pode perdurar por vários Exercícios/AC. Em caso de desenvolvimento de uma nova versão, este deve ser tratado como um novo projeto de desenvolvimento. As alterações em uma versão são consideradas como projeto de manutenção.

11. Conclusão

Este documento apresentou um guia para o dimensionamento de tamanho de todos os tipos de projetos de software desenvolvidos pelo SERPRO. O tamanho funcional é aferido com base na métrica de Pontos de Função Não Ajustados como unidade de medida, conforme recomendado nos Acórdãos do Tribunal de Contas da União (TCU).

Além disso, foram apresentadas diretrizes para facilitar a utilização do CPM, visando melhorar o entendimento das contagens de Pontos de Função realizadas. Os projetos do cliente STN devem ser contados com base no Roteiro STN. Este Roteiro, disponível no site do Escritório de Métricas, apresenta orientações de contagem de PF para sistemas do cliente STN.

As planilhas de contagem de PF também encontram-se disponíveis no site do Escritório de Métricas.

Como trabalho futuro recomenda-se a revisão e atualização deste roteiro sempre que se verificar inconsistência entre alguma definição do IFPUG, seja publicada em versões futuras do CPM ou em *White Paper*, ou quando for detectado um novo tipo de serviço associado ao desenvolvimento de software não previsto neste trabalho. Este Roteiro também precisa ser revisado sempre que ocorrerem atualizações no Roteiro de Métricas do SISP e demais Roteiros de Contratos com clientes.

Futuramente, sugere-se pesquisar outros modelos de estimativa de esforço e prazo, por exemplo o COCOMO II, visando a comparação das estimativas de prazo e esforço por mais de um método. É recomendada a evolução da ferramenta Pontua para a automatização das contagens e armazenamento de dados históricos.

Referências Bibliográficas

- [Boehm, 2000] BOEHM, B.W. **Software Cost Estimation With COCOMO II**. Prentice Hall, New Jersey, 2000.
- [Hazan, 2008] HAZAN, C. **Análise de Pontos de Função: Uma Aplicação nas Estimativas de Tamanho de Projetos de Software**. Engenharia de Software Magazine, Edição 2, Devmedia, pp.25-31.
- [IEEE,1998] IEEE Computer Society. **IEEE Standard for Software Maintenance**. IEEE Std 1219, 1998.
- [Meli, 1999] MELI, R.; SANTILLO, L. **Function Point Estimation Methods: A Comparative Overview**. Proceedings of FESMA 99, Amsterdam, Netherlands, October 1999, pp. 271-286.
- [IEEE,1998] IEEE Computer Society. **IEEE Standard for Software Maintenance**. IEEE Std 1219, 1998.
- [IFPUG,2009] IFPUG. **Considerations for Counting with Multiple Media**. Release 1.0, September, 2009.
- [IFPUG,2010] IFPUG. **Counting Practices Manual**. Version 4.3, January, 2010.
- [Jones, 2007] JONES, C. **Estimating Software Costs**. Second Edition, Mc Graw Hill, 2007.
- [NESMA, 2009] NESMA. **Function Point Analysis for Software Enhancement Guidelines**. Version 2.2.1, 2009
- [Parthasarathy,2007] PARTHASARATHY, M. A. **Practical Software Estimation: function point methods for insourced and outsourced projects**. Addison Wesley, New York, 2007.
- [Roetzheim, 2005] ROETZHEIM, W. **Estimating and Managing Project Scope for New Development**. CrossTalk, Vol. April, 2005.
- [Sommerville, 2007] SOMMERVILLE, I. **Software Engineering**. Pearson Education Limited, 8th Edition, 2007.
- [Vazquez, 2010] VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. **Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software**. 9ª Edição. Editora Érica, São Paulo.