

Anexo IE

Guia de Métricas da Superintendência de Tecnologia da Informação – STI Versão 1.0

Brasília/DF – Outubro de 2019

Sumário

1. Introdução	4
2. Objetivo	6
3. Contagem de Pontos de Função:	7
3.1. Determinar propósito, tipo e escopo da contagem e fronteira da aplicação	8
3.2. Identificação de processo elementar	8
3.3. Identificar Funções de Dados e Funções Transacionais	9
3.3.1. Funções do tipo dado:	9
3.3.2. Funções do tipo transação:	10
3.3.3. Contribuição dos tipos funcionais:	10
3.4. Calcular tamanho funcional	11
3.5. Requisitos não funcionais	12
4. Cálculo de pontos de função	14
4.1. Iniciativa de desenvolvimento	14
4.2. Iniciativa de melhoria	14
4.2.1. Observações sobre alterações:	16
4.2.2. Outros tipos de funções alteradas:	17
4.3. Iniciativa de migração de dados – PF_CONVERSÃO	18
4.4. Manutenção corretiva	18
4.5. Mudança de plataforma	20
4.5.1. Mudança de plataforma – Linguagem de programação:	20
4.5.2. Mudança de plataforma – Banco de dados:	21
4.6. Atualização de versão	22
4.6.1. Atualização de versão – Linguagem de programação:	22
4.6.2. Atualização de versão – Browser:	23
4.6.3. Atualização de versão – Banco de dados:	23
4.7. Manutenção em interface	24
4.8. Adaptação em funcionalidades sem alteração de requisitos funcionais	25
4.9. Apuração especial	26
4.9.1. Apuração especial – Base de dados:	27
4.9.2. Apuração especial – Geração de relatórios:	28
4.9.3. Apuração especial – Reexecução:	29

4.10.	Atualização de dados	29
4.11.	Desenvolvimento, manutenção e publicação de páginas estáticas de intranet, internet ou portal.....	30
4.12.	Verificação de erros.....	31
4.13.	Pontos de função de teste	31
4.14.	Componente interno reusável	32
4.15.	Dados de Código (<i>code tables</i>)	33
5.	Orientações complementares para contagem.....	35
5.1.	Contagem de pontos de função com múltiplas mídias.....	35
5.1.1.	Cenário 1 - Mesmos dados apresentados em tela e impressos:.....	36
5.1.2.	Cenário 2 - Mesmos dados de saída como dados em arquivo e relatório impresso:37	
5.1.3.	Cenário 3 - Mesmos dados de entrada batch e on-line:	37
5.1.4.	Cenário 4 - Múltiplos canais de entrega da mesma funcionalidade:	37
5.1.5.	Cenário 5 - Relatório em múltiplos formatos:.....	38
5.2.	Log, trilha de auditoria e histórico	38
5.2.1.	Log:	38
5.2.2.	Trilha de auditoria:	39
5.2.3.	Histórico:	39
5.3.	Arquivos para processamento	40
5.4.	API/Webservices	41
5.5.	Principais falhas da contagem identificadas	44
6.	Regras auxiliares.....	45
6.1.	Contagem estimativa de pontos de função (CEPF)	45
6.2.	Distribuição de esforço por fase da iniciativa	50
6.2.1.	Evidências de entrega:	51
6.2.2.	Estratégias de desenvolvimento de <i>back</i> e <i>frontend</i> em separado:.....	51
6.3.	Fator de ajuste	51
7.	Referências bibliográficas	53
8.	Controle de versão	54

1. Introdução

Atualmente, a métrica de pontos de função é largamente utilizada por empresas de vários setores empresariais do Brasil, dos segmentos públicos e privados. O seu objetivo é estimar, dimensionar e definir o tamanho funcional de software. Sendo assim, trata-se de uma unidade de medida que é mensurada a partir do ponto de vista do usuário, onde deve ser observado o que o sistema faz, independentemente de como ele foi construído.

Por ser uma medida de tamanho, a métrica de pontos de função não mede o esforço total (utilização de recursos humanos, técnicos ou tecnológicos e financeiros) para a concepção de um sistema. Porém, é possível correlacionarmos a referida métrica a demais variáveis e identificar tanto o esforço total, como a produtividade de um time de desenvolvimento, auxiliando assim os envolvidos nos diversos campos de gestão das iniciativas de desenvolvimento de sistemas.

A utilização da métrica de pontos de função permite que as organizações percebam vários benefícios dentre os quais destacam-se:

- Aplicabilidade em vários momentos do ciclo de vida de um sistema.
- Aplicável para determinar o tamanho de um sistema e seus subconjuntos (módulo, funcionalidade, função, etc).
- Complementa o gerenciamento dos requisitos ao auxiliar na verificação da solidez e completeza dos requisitos especificados.
- Facilidade de estimativa nas fases iniciais do ciclo de vida de um sistema.
- Fator de normalização para comparação de software ou para a comparação da produtividade na utilização de diferentes técnicas.
- Independência da solução tecnológica utilizada.
- Regra de contagem objetiva que viabiliza também a mensuração objetiva do tamanho de um sistema ou funcionalidade.
- Suporta a análise de produtividade e qualidade, seja diretamente ou em conjunto com outras métricas como esforço, defeitos e custo.
- Viabiliza análise de fazer ou comprar um determinado pacote.

- Viabiliza meio de estimar custos e recursos para desenvolvimento ou manutenção de sistema.

2. Objetivo

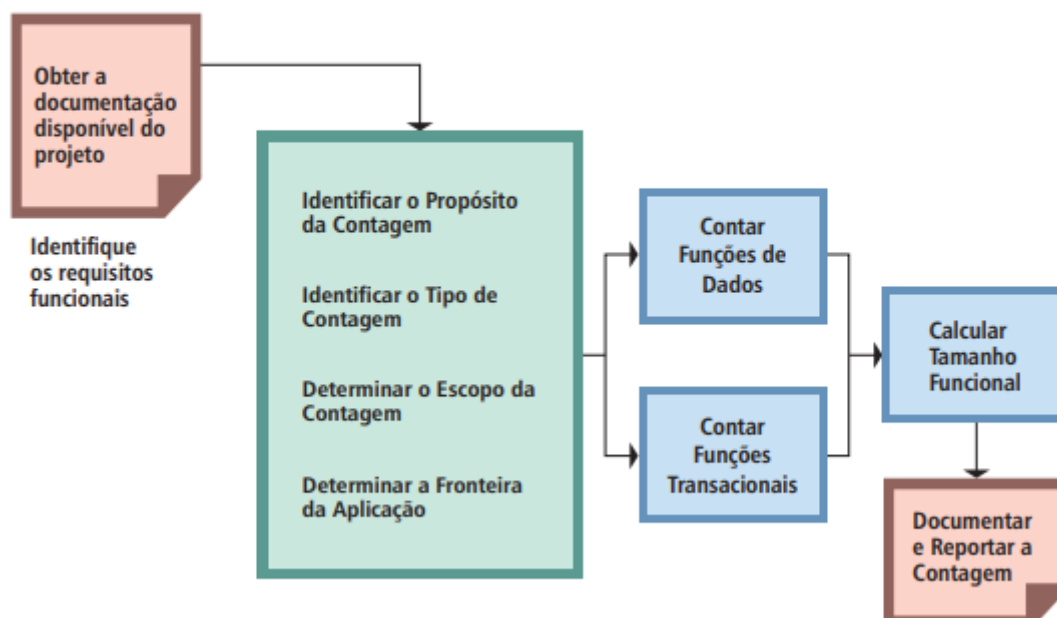
Este Guia de Métricas da Superintendência de Tecnologia da Informação – STI tem como objetivo principal apresentar um roteiro de métricas, com base e complementando as regras de contagem de pontos de função do Manual de Práticas de Contagem (*Counting Practices Manual* - CPM 4.3.1), para vários tipos de iniciativas de desenvolvimento e de manutenção de sistemas sob responsabilidade da STI, que estejam relacionados ou de posse das entidades atendidas por esta Superintendência: Confederação Nacional da Indústria – CNI, do Instituto Euvaldo Lodi Núcleo Central – IEL/NC, do Serviço Social da Indústria Departamento Nacional – SESI/DN e do Serviço Nacional de Aprendizagem Industrial Departamento Nacional – SENAI/DN.

Desta forma, espera-se promover o uso de métricas objetivas nos contratos de prestação de serviços dessas iniciativas de sistemas (no CPM o termo iniciativa é tratado como projeto).

3. Contagem de Pontos de Função:

A métrica PF mede o tamanho funcional de uma iniciativa de desenvolvimento de sistema, observando as funcionalidades implementadas, considerando a visão do usuário. O tamanho funcional é definido como tamanho do sistema a partir da contabilização dos requisitos funcionais do sistema que são observados pelo usuário. A métrica PF é independente da metodologia e tecnologia utilizadas. A Análise de Pontos de Função (APF) é um método padrão para a medição de iniciativas de desenvolvimento e de manutenção de sistemas, visando estabelecer uma medida de tamanho do software em pontos de função, com base na quantificação das funcionalidades solicitadas e entregues, sob o ponto de vista do usuário. Assim, a APF tem como objetivo medir o que o software faz, por meio de uma avaliação padronizada dos requisitos de negócio do sistema. O Manual de Práticas de Contagem (CPM 4.3.1.) [IFPUG, 2010b] apresenta as regras de contagem de pontos de função de iniciativas de desenvolvimento, iniciativas de melhoria e aplicações implantadas. A Figura 1 ilustra o procedimento de contagem de pontos de função, descrito nas seções seguintes.

Figura 1: Procedimento de Contagem de Pontos de Função



3.1. Determinar propósito, tipo e escopo da contagem e fronteira da aplicação

A contagem de pontos de função se inicia com a análise da documentação disponível da iniciativa em questão, visando a identificação dos requisitos funcionais. O próximo passo é o estabelecimento do propósito da contagem, o qual fornece uma resposta para uma questão de negócio a ser resolvida, por exemplo: necessidade de dimensionar uma iniciativa de um novo sistema para auxiliar o processo de contratação do mesmo. Com base no propósito da contagem são definidos o escopo da contagem e o tipo de contagem. O escopo da contagem identifica quais funcionalidades serão incluídas na contagem de pontos de função, e o tipo de contagem identifica se a iniciativa é de desenvolvimento, de melhoria ou aplicação instalada. A fronteira da aplicação, que é a interface conceitual que indica o limite lógico entre o sistema sendo medido e os usuários (pessoas, sistemas, dispositivos ou qualquer coisa que interaja com o sistema), deve ser definida com base na visão do usuário (do negócio), desconsiderando questões de implementação. Deve-se ressaltar que toda contagem de pontos de função é realizada dentro de uma fronteira estabelecida.

O estabelecimento da fronteira da aplicação pode ser subjetivo, por exemplo, em uma aplicação com vários módulos, a fronteira pode ser estabelecida para cada módulo ou subsistema ou, ainda, pode-se considerar toda a aplicação, dependendo da visão do usuário. De fato, a definição da fronteira depende de processos de negócios, além disso, o posicionamento da fronteira influencia fortemente a contagem de pontos de função. Desta forma, devido a essa subjetividade, em editais para contratação de iniciativas de manutenção é fortemente recomendado a definição das fronteiras de todas as aplicações a serem contratadas.

3.2. Identificação de processo elementar

Antes de identificarmos as funções, vale ressaltar o conceito básico acerca do processo elementar. Um Processo Elementar é a menor unidade de

atividade que é significativa para o(s) usuário(s). O Processo Elementar deve ser autocontido e deixar o negócio da aplicação que está sendo contada em um estado consistente.

Um processo elementar com múltiplas formas de processamento lógico não deve ser dividido em múltiplos processos elementares. Se um processo elementar é subdividido inapropriadamente, o mesmo não reúne os critérios de um processo elementar.

Como exemplo pode-se destacar o recurso de abas para a construção de interfaces em sistemas computacionais. Em geral, cada aba de uma tela não representa uma transação completa e desta forma não deve ser considerada um processo elementar isolado. O adequado para este tipo de situação é identificar um processo elementar para um conjunto de abas que se relacionam em vez de considerar um processo elementar para cada aba.

Ressalta-se a importância do atendimento a todos os critérios listados no Manual de Práticas de Contagem do IFPUG e da observação dos seus exemplos para a correta identificação de um processo elementar evitando, por exemplo, relacionar processos elementares com telas ou abas de uma transação, considerar fluxos alternativos como processos elementares distintos ou considerar transações que atualizem dados de código.

3.3. Identificar Funções de Dados e Funções Transacionais

Uma vez estabelecida a fronteira da aplicação e o escopo, o próximo passo é o mapeamento dos requisitos de dados e de funções transacionais para os tipos funcionais da APF.

3.3.1. Funções do tipo dado:

- a) Arquivo Lógico Interno (ALI): é um grupo de dados, logicamente relacionados, reconhecido pelo usuário, mantido por meio de um ou mais processos elementares da aplicação que está sendo contada.

- b) Arquivo de Interface Externa (AIE): é um grupo de dados, logicamente relacionados, reconhecido pelo usuário, mantido em outra aplicação e referenciado pela aplicação que está sendo contada. O AIE é obrigatoriamente um ALI de outra aplicação.

3.3.2. Funções do tipo transação:

- a) Entrada Externa (EE): é um processo elementar que processa dados ou informação de controle que entram pela fronteira da aplicação. Seu objetivo principal é manter um ou mais ALI ou alterar o comportamento do sistema.
- b) Consulta Externa (CE): é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. Seu objetivo principal é apresentar informação para o usuário através da recuperação de dados ou informação de controle de ALI ou AIE.
- c) Saída Externa (SE): é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. Seu objetivo principal é apresentar informação para um usuário ou outra aplicação através de um processamento lógico adicional à recuperação de dados ou informação de controle. O processamento lógico deve conter cálculo, ou criar dados derivados, ou manter ALI ou alterar o comportamento do sistema.

3.3.3. Contribuição dos tipos funcionais:

Após a identificação dos tipos funcionais para cada requisito funcional definido no documento de requisitos do sistema, deve-se avaliar a complexidade (Baixa, Média, Alta) e a contribuição funcional do mesmo para a contagem de pontos de função, observando as regras de contagem de pontos de função descritas no *CPM*. A identificação e a avaliação das complexidades dos tipos funcionais é feita pela avaliação objetiva de dois parâmetros, com suas regras de contagem descritas no *CPM*. A contagem de pontos de função deve seguir rigorosamente as regras de contagem do *CPM* e as definições complementares

por ventura presentes neste guia. A Tabela 1 apresenta a contribuição dos tipos funcionais na contagem de pontos de função.

Tabela 1: Contribuição Funcional dos Tipos Funcionais (Fonte: CPM 4.3)

Tipo Funcional	Complexidade		
	Baixa	Média	Alta
Arquivo Lógico Interno (ALI)	7 PF	10 PF	15 PF
Arquivo de Interface Externa (AIE)	5 PF	7 PF	10 PF
Entrada Externa (EE)	3 PF	4 PF	6 PF
Saída Externa (SE)	4 PF	5 PF	7 PF
Consulta Externa (CE)	3 PF	4 PF	6 PF

3.4. Calcular tamanho funcional

Seguem as definições dos termos técnicos da Análise de Pontos de Função utilizados nas fórmulas de dimensionamento de iniciativas de sistemas propostas neste guia:

- PF_INCLUÍDO:** pontos de função associados às novas funcionalidades que farão parte da aplicação após uma iniciativa de desenvolvimento ou de manutenção.
- PF_ALTERADO:** pontos de função associados às funcionalidades existentes na aplicação que serão alteradas na iniciativa de manutenção.
- PF_EXCLUÍDO:** pontos de função associados às funcionalidades existentes na aplicação que serão excluídas na iniciativa de manutenção.
- PF_CONVERSÃO:** pontos de função associados às funcionalidades de conversão de dados das iniciativas de desenvolvimento ou de manutenção. Exemplos de funções de conversão incluem: migração ou carga inicial de dados para popular as novas tabelas criadas (Entradas Externas) e relatórios

associados à migração de dados, caso requisitado pelo usuário (Saídas Externas ou Consultas Externas). Observe que os dados carregados em um processo de migração não devem ser contados como Arquivos de Interface Externa.

3.5. Requisitos não funcionais

A métrica Ponto de Função é uma métrica de tamanho funcional, ou seja, dimensiona iniciativa de sistemas com base nos requisitos funcionais da aplicação, não contemplando diretamente os requisitos não funcionais da iniciativa. Nesse sentido, em contratos de sistemas baseados na métrica Ponto de Função é fundamental definir claramente no edital os requisitos não funcionais da iniciativa a serem atendidos pela empresa contratada. Os requisitos não funcionais impactam no esforço e, conseqüentemente, no custo da iniciativa, porém, não apresentam contagem direta na métrica de pontos de função. Os requisitos não funcionais estão associados aos aspectos qualitativos de um sistema, considerando aspectos relacionados ao uso do mesmo. Seguem abaixo alguns tipos de requisitos não funcionais, com exemplos, que podem ser mencionados nos editais:

- a) Usabilidade: a solução deve atender aos requisitos dos Padrões Web em Governo Eletrônico (e-PWG) – Cartilha de Usabilidade; a aplicação deve ter help on-line de sistema, tela e campo (sensível a contexto); a aplicação deve ser disponibilizada nos idiomas Português, Espanhol e Inglês.
- b) Técnicos: a aplicação deve funcionar adequadamente nos navegadores: Internet Explorer 7.0 ou superior e Mozilla Firefox 3.0 ou superior; a solução deve ser desenvolvida em linguagem Java com banco de dados PostgreSQL; para o desenvolvimento da solução, deve ser utilizado preferencialmente um dos seguintes frameworks Java: Demoiselle, Jaguar e MDArt; a solução deve atender aos requisitos do e-PWG; deve utilizar as

ferramentas AWSTATS e Google Analytics para gerar estatísticas de acesso.

- c) Segurança: a aplicação deve realizar controle de segurança dos dados de acordo com política de backup definida em conformidade com a norma ISO/IEC 27002.
- d) Acessibilidade: a solução deve ser aderente ao Modelo de Acessibilidade de Governo Eletrônico (e-MAG).
- e) Performance: o tempo de resposta da aplicação não deve exceder 10 segundos; a solução deve suportar até 1.000 acessos simultâneos.
- f) Interoperabilidade: a solução deve ser aderente aos Padrões de Interoperabilidade de Governo Eletrônico (e-PING).

4. Cálculo de pontos de função

Este capítulo tem como propósito descrever os diversos tipos de iniciativas de sistemas e definir métricas para seu dimensionamento baseadas nas regras de contagem de pontos de função do *CPM*.

Existe a previsão para tratamento dos casos onde não há a necessidade de contratar todas as fases do ciclo de vida do sistema. Dessa forma, a contratada será remunerada pela contagem de pontos de função considerando apenas os percentuais das fases contratadas, conforme os níveis percentuais sugeridos na Tabela 7, seção 6.2. Exemplo: para uma nova iniciativa de desenvolvimento de um sistema de treinamentos, que não exista a intenção de contratar as fases de refinamento e de testes, a contratada será remunerada pela contagem de pontos de função desconsiderando os percentuais dessas fases.

4.1. Iniciativa de desenvolvimento

É a iniciativa para desenvolver e entregar a primeira versão de uma aplicação de sistema. Seu tamanho funcional é a medida das funcionalidades entregues ao usuário no final da iniciativa. Também são consideradas as funcionalidades de conversão de dados. Segue a fórmula de cálculo utilizada no dimensionamento de iniciativas de desenvolvimento de sistemas:

$$PF_DESENVOLVIMENTO = PF_INCLUIDO + PF_CONVERSÃO$$

4.2. Iniciativa de melhoria

A iniciativa de melhoria (iniciativa de melhoria funcional ou manutenção evolutiva), está associado às mudanças em requisitos funcionais da aplicação, ou seja, à inclusão de novas funcionalidades, alteração ou exclusão de funcionalidades em aplicações implantadas. Trata-se de modificação de um produto de software existente para mantê-lo funcionando adequadamente em um ambiente que sofre mudanças.

A iniciativa de melhoria é considerada um tipo de iniciativa de manutenção adaptativa com mudanças em requisitos funcionais da aplicação, ou seja, com funcionalidades incluídas, alteradas ou excluídas na aplicação, segundo o CPM 4.3.1.

Este guia separa a iniciativa de melhoria (quando as mudanças são associadas aos requisitos funcionais) da iniciativa de manutenção adaptativa (quando as mudanças estão associadas aos requisitos não funcionais da aplicação). Uma iniciativa 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. Segue a fórmula de cálculo utilizada no dimensionamento de iniciativas de melhoria de sistemas:

$$PF_MELHORIA = PF_INCLUIDO + (FI \times PF_ALTERADO) + (0,30 \times PF_EXCLUIDO) + PF_CONVERSÃO$$

a) FI (Fator de Impacto) pode variar de 50% a 90% conforme condições abaixo:

- FI = 50% para funcionalidade de sistema desenvolvida ou mantida por meio de uma iniciativa de melhoria pela empresa contratada.
 - FI = 75% para funcionalidade de sistema não desenvolvida ou mantida por meio de uma iniciativa de melhoria pela empresa contratada e sem necessidade de redocumentação da funcionalidade.
 - FI = 90% para funcionalidade de sistema não desenvolvida ou mantida por meio de uma iniciativa de melhoria pela empresa contratada e com necessidade de redocumentação da funcionalidade.
- FI = 90% representa a adição de 15% como fator de redocumentação ao Fator de Impacto anterior (75%). Nesse caso, a contratada deve redocumentar a funcionalidade mantida, gerando a documentação completa da mesma, aderente ao processo de software da contratante. Se houver uma nova demanda de iniciativa de melhoria na funcionalidade em questão, será considerado que a contratada

desenvolveu a funcionalidade. Observe que o percentual de 90% apenas será considerado na primeira demanda de iniciativa de melhoria em cada funcionalidade

Este guia propõe um fator de redocumentação menor para iniciativas de manutenção (melhoria, corretiva e adaptativa) do que o fator proposto em iniciativas específicas de redocumentação seção 4.12 deste guia. Isso porque, em iniciativas de manutenção de uma funcionalidade sem documentação, é necessário realizar o entendimento da funcionalidade para poder modificá-la e testá-la, ou seja, é necessário realizar a engenharia reversa da funcionalidade para executar os testes corretamente. Assim sendo, a redocumentação requisitada em iniciativas de melhoria requer um esforço menor do que em iniciativas de redocumentação, descritos na seção 4.12, onde é necessário remunerar todo o esforço de engenharia reversa e a atividade de documentação. Em iniciativas de manutenção, o fator de 15% está remunerando apenas a atividade de documentação.

O PF_INCLUIDO e PF_ALTERADO preveem a contratação e realização de todas as fases do processo de desenvolvimento de software. Caso alguma fase não seja contratada ou realizada, deve-se aplicar, em cada um deles, um redutor que corresponde ao percentual da fase não contratada ou não realizada, conforme percentuais inseridos na Tabela 7, seção 6.2.

4.2.1. Observações sobre alterações:

- a) Função de dados: Uma função de dados (Arquivo Lógico Interno ou Arquivo de Interface Externa) é considerada alterada quando houver inclusão ou exclusão de Tipos de Dados (TD). De acordo com o glossário do CPM 4.3.1, um Tipo de Dados (*DET – Data Element Type*) é um atributo único, reconhecido pelo usuário e não repetido. Também é considerada alterada se algum tipo de dado sofrer mudança de tamanho (número de posições) ou tipo de campo (por exemplo: mudança de numérico ou alfanumérico), caso a mudança decorra de alteração de regra de negócio.

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

- Mudança de tipos de dados;
- Mudança de arquivos referenciados;
- Mudança de lógica de processamento.

c) Sobre a lógica de processamento: O CPM 4.3.1 define lógica de processamento como requisitos especificamente solicitados pelo usuário para completar um processo elementar. Esses requisitos devem incluir uma ou mais das seguintes ações:

- Condições são analisadas para verificar quais são aplicáveis;
- Dados derivados são criados através da transformação de dados existentes, para criar dados adicionais;
- Dados ou informações de controle são recuperados;
- Dados são filtrados e selecionados através da utilização de critérios;
- Dados são reordenados;
- Fórmulas matemáticas e cálculos são executados;
- 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;
- Um ou mais ALIs ou AIEs são referenciados;
- Um ou mais ALIs são atualizados;
- Validações são executadas;
- Valores equivalentes são convertidos.

4.2.2. Outros tipos de funções alteradas:

Este guia considera como função alterada qualquer mudança em funcionalidades da aplicação devido às mudanças de regras de negócio. Por exemplo, uma funcionalidade de cadastro envolvia a inclusão de um telefone do gerente. Devido a mudanças no processo de negócio, a funcionalidade deve

sofrer uma manutenção para cadastrar dois telefones do gerente. Desta forma, o guia considera esta função como uma Entrada Externa alterada, PF_ALTERADO em uma iniciativa de melhoria, mesmo que não existam mudanças de lógica de processamento, de tipos de dados ou de arquivos referenciados. Serão tratadas como manutenções adaptativas apenas as manutenções que implicarem exclusivamente em mudanças em requisitos não funcionais. Se uma mesma funcionalidade tiver mudanças em requisitos funcionais e não funcionais, esta deve ser contada apenas uma vez, como função alterada em uma iniciativa de melhoria.

4.3. Iniciativa de migração de dados – PF_CONVERSÃO

As iniciativas de migração de dados devem ser contadas como nova iniciativa de desenvolvimento de um sistema, seguindo a fórmula abaixo:

$$PF_CONVERSÃO = PF_INCLUIDO$$

Uma iniciativa de migração deve contemplar minimamente: as Entradas Externas – considerando as cargas de dados nos ALI – e, caso seja solicitado pelo usuário, os relatórios gerenciais das cargas, que serão contados como Saídas Externas. Todas as contagens de PF devem ser realizadas com base nas funcionalidades requisitadas e recebidas pelo 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 Internos.

4.4. Manutenção corretiva

Mesmo com a execução de atividades de garantia da qualidade, pode-se 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 de sistemas em produção.

É importante destacar que as demandas de manutenção corretiva frequentemente precisam ser atendidas com urgência. Assim, o grau de criticidade da iniciativa poderá trazer impacto nas estimativas de custo e esforço.

Quando o sistema em produção tiver sido desenvolvido pela contratada, a manutenção corretiva será do tipo Garantia se estiver no período de cobertura e em conformidade com as demais condições de garantia previstas em contrato. Caso não exista cláusula contratual de garantia, deve ser considerada a garantia preconizada por lei (Código do Consumidor). Desta forma, não é aplicável a mensuração de pontos de função, e, tampouco, o pagamento destas manutenções.

Quando o sistema estiver fora da garantia ou não tenha sido desenvolvido pela empresa contratada, deverá ser estimado e calculado o tamanho da iniciativa de manutenção corretiva. Nestes casos, a aferição do tamanho em pontos de função da funcionalidade ou das funcionalidades corrigidas deve considerar um fator de impacto (FI) sobre o PF_ALTERADO.

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

a) Fator de Impacto (FI):

- 50% quando estiver fora da garantia e a correção for feita pela mesma empresa que desenvolveu a funcionalidade.
- 75% quando estiver fora da garantia e a correção for feita por empresa diferente daquela que desenvolveu a funcionalidade.

As demandas de manutenção corretiva não contemplam atualização de documentação da funcionalidade corrigida, pois este guia considera que, normalmente, manutenção corretiva não se refere a erros de requisitos. Caso seja erro em requisitos, essa demanda deve ser tratada como iniciativa de melhoria (alteração de funcionalidade), descrito na seção 4.2. Porém, quando o erro for causado por documentação dúbia ou imprecisa (elaborada pela contratada) da funcionalidade corrigida, a manutenção corretiva poderá contemplar os ajustes na documentação, mesmo fora da garantia, mediante negociação entre as partes.

Caso seja demandada a redocumentação da funcionalidade corrigida, porque a documentação não existe ou está desatualizada, deve-se adicionar ao FI um fator de redocumentação de 15%, conforme descrito na seção 4.2.

4.5. Mudança de plataforma

São considerados nesta categoria, iniciativas que tratam a migração de software para outra plataforma. Por exemplo, um sistema legado em COBOL que necessita ser redesenvolvido em JAVA; o banco de dados de um sistema legado que precisa ser migrado para o DB2.

Recomenda-se enfaticamente a realização da análise de impacto das mudanças propostas, para efeito de determinação do percentual adequado para aplicação sobre o total de pontos de função das funcionalidades impactadas. Por exemplo, em uma análise de impacto pode ser identificado que não haverá mudanças no código-fonte ou em função transacional, sendo necessário apenas testar o sistema, então deve-se utilizar um percentual contemplando apenas a fase de testes. No caso de o teste apontar a necessidade de atualizar alguma função transacional, não deve ser contado o esforço do teste, mas sim o esforço abordado nesta seção, conforme as fórmulas apresentadas nos tópicos seguintes.

As próximas subseções apresentam os tipos de iniciativas de mudança de plataforma. As iniciativas de mudança de plataforma que se enquadram em mais de uma subseção, devem ser contados apenas uma vez, considerando o tipo de iniciativa com maior contagem de pontos de função.

4.5.1. Mudança de plataforma – Linguagem de programação:

Nesta categoria encontram-se as demandas de redesenvolvimento de sistemas em outra linguagem de programação. Como os sistemas legados, frequentemente, não possuem documentação, devem ser considerados como novas iniciativas de desenvolvimento. Assim, será utilizada a fórmula de iniciativas de desenvolvimento do *CPM* 4.3.1.

Observe que caso não exista mudança nas funções de dados, ou seja, o banco de dados da aplicação seja mantido, as funções de dados não devem ser contadas. No entanto, nesse caso, deve ser realizada a contagem das funções de dados a fim de compor a documentação da contagem final da iniciativa.

Outro ponto a ser observado são as fases contratadas. Caso a iniciativa já possua documentação de requisitos, a fase de refino não será contratada. Deve-se considerar apenas os percentuais das fases contratadas, aplicando-se sobre o PF_INCLUIDO o fator de ponderação previsto na Tabela 7, seção 6.2.

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

Este guia recomenda a supressão do PF_CONVERSÃO da fórmula de contagem de pontos de função de iniciativas de redesenvolvimento quando for caracterizado um esforço relativamente maior dessa atividade, conforme descrito na seção 3.4.

4.5.2. Mudança de plataforma – Banco de dados:

Nesta categoria encontram-se as demandas de redesenvolvimento de sistemas para utilizar um outro sistema gerenciador de banco de dados.

Observe que caso não exista mudança nas funções de dados, ou seja, o banco de dados da aplicação seja mantido, então as funções de dados não devem ser contadas. No entanto, nesse caso, deve ser realizada a contagem das funções de dados a fim de compor a documentação da contagem final da iniciativa.

Caso a iniciativa já possua documentação de requisitos, então a fase de refino não deve ser contratada. É importante destacar que isso se aplica a qualquer fase que não se deseja contratar. Deve-se considerar apenas os percentuais das fases contratadas, aplicando-se sobre o PF_INCLUIDO o fator de ponderação previsto na Tabela 7, seção 6.2. Desta forma, a seguinte forma deve ser utilizada:

$$PF_REDESENVOLVIMENTO_BD_RELACIONAL = (PF_ALTERADO \times 0,30) + PF_CONVERSÃO$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7 – seção 6.2).

Indica-se tratar o PF_CONVERSÃO dentro da mesma iniciativa. Geralmente a estrutura de dados não é alterada, desta forma não contamos as funções de dados.

4.6. Atualização de versão

São consideradas nesta categoria, demandas para uma aplicação existente - ou parte de uma aplicação existente - executar em versões diferentes de browsers (ex: Internet Explorer, Firefox, Chrome, etc) ou de linguagens de programação (ex: versão mais atual do JAVA). Também são consideradas nesta categoria atualização de versão de banco de dados.

Outro ponto a ser observado é a classificação, em alguns casos, dessas demandas como componente interno reusável (seção 4.15).

Recomenda-se enfaticamente a realização da análise de impacto das mudanças propostas para efeito de determinação do percentual adequado para aplicação sobre o total de pontos de função das funcionalidades impactadas. Por exemplo, em uma análise de impacto, pode ser identificado que não haverá mudanças no código-fonte ou em função transacional, sendo necessário somente testar o sistema, então deve-se utilizar um percentual contemplando apenas a fase de testes. No caso de o teste apontar a necessidade de atualizar alguma função transacional, não deve ser contado o esforço do teste, mas sim o esforço abordado nesta seção, conforme as fórmulas apresentadas nas subseções seguintes.

4.6.1. Atualização de versão – Linguagem de programação:

Nesta categoria encontram-se as demandas de atualização de versão de linguagem de programação de sistemas. As funções de dados não devem ser contadas. Estas demandas devem ser dimensionadas de acordo com a fórmula a seguir.

$$PF_ATUALIZAÇÃO_VERSÃO_LINGUAGEM = PF_ALTERADO \times 0,30$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7 – seção 6.2).

4.6.2. Atualização de versão – Browser:

Nesta categoria encontram-se as demandas de atualização de aplicações Web para executar em novas versões de um mesmo browser e para suportar a execução em mais de um browser. É importante destacar que este tipo de procedimento usualmente é realizado quando é necessário resolver algum problema de incompatibilidade. As funções de dados não devem ser contadas. Estas demandas devem ser dimensionadas de acordo com a fórmula abaixo.

$$PF_ATUALIZAÇÃO_VERSÃO_BROWSER = PF_ALTERADO \times 0,30$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7 – seção 6.2).

Essas atualizações podem implicar em manutenções em componentes específicos da plataforma utilizada. Nesse caso, a demanda deve ser contada como componente interno reusável, descrita na seção 4.15 deste guia.

4.6.3. Atualização de versão – Banco de dados:

Nesta categoria encontram-se as demandas de atualização de versão do sistema gerenciador de banco de dados. As funções de dados não devem ser

contadas. Estas demandas devem ser dimensionadas de acordo com a fórmula a seguir.

$$PF_ATUALIZAÇÃO_VERSÃO_BD = PF_ALTERADO \times 0,30$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7 – seção 6.2).

4.7. Manutenção em interface

A manutenção em interface, denominada na literatura de manutenção cosmética, é associada às demandas de alterações de interface, por exemplo: fonte de letra, cores de telas, logotipos, mudança de botões na tela, mudança de posição de campos ou texto na tela. Também se enquadram nessa categoria as seguintes manutenções:

- Alteração de *labels* de uma tela de consulta;
- Alteração de título de um relatório;
- Desenvolvimento ou atualização de help estático de funcionalidades;
- Mudanças de texto em mensagens de 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.

Nestes casos, a aferição do tamanho em pontos de função das funções transacionais impactadas será realizada com a aplicação de um fator de redução de modo a considerar 20% da contagem de uma função transacional de mais baixa complexidade (3 PF), ou seja 0,6 PF, independentemente da complexidade da funcionalidade alterada. Neste tipo de manutenção não são contadas funções de dados.

$$PF_INTERFACE = 0,6 PF \times QUANTIDADE \text{ DE FUNÇÕES} \\ \text{TRANSACIONAIS IMPACTADAS}$$

4.8. Adaptação em funcionalidades sem alteração de requisitos funcionais

São consideradas nesta categoria as demandas de manutenção adaptativa associadas a solicitações que envolvem aspectos não funcionais, sem alteração em requisitos funcionais. Seguem alguns exemplos:

- Adaptação de uma funcionalidade para possibilitar a chamada por um webservice ou para outro tipo de integração com outros sistemas;
- Adequar mensagem do sistema que em algumas telas apresenta “Usuário Não está Habilitado a ver esta Página”, para que passe a enviar uma mensagem mais adequada ao fato do usuário não possuir mais uma sessão ativa e ainda estar navegando no sistema. A demanda deve ser contada como manutenção adaptativa considerando as funcionalidades impactadas. Observe que se trata de mudança em validação com regra de negócio não funcional;
- Alteração na aplicação para adaptação às alterações realizadas na interface com rotinas de integração com outros softwares, por exemplo, alteração em subrotinas chamadas por este software;
- Aumentar a quantidade de linhas por página em um relatório;
- Colocar paginação em um relatório;
- Limitar a quantidade de linhas por página em uma consulta existente;
- Modificar o servidor a ser acessado em uma funcionalidade de download de arquivo;
- Permitir exclusões múltiplas em uma funcionalidade que antes só possibilitava a exclusão de um item;
- Replicação de funcionalidade: chamar uma consulta existente em outra tela da aplicação.

Nestes casos, a aferição do tamanho em pontos de função da funcionalidade ou das funcionalidades que sofreram impacto deve considerar um fator de impacto (FI) sobre o PF_ALTERADO, seguindo os conceitos do CPM 4.3.1, apresentados na seção 4.2.

$$PF_ADAPTATIVA = FI \times PF_ALTERADO$$

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

- FI = 50% para funcionalidade de sistema desenvolvida ou mantida por meio de uma iniciativa de melhoria pela empresa contratada.
- FI = 75% para funcionalidade de sistema não desenvolvida ou mantida por meio de uma iniciativa de melhoria pela empresa contratada.

Deve-se destacar que além da adequação das funcionalidades em questão, a documentação da iniciativa de manutenção adaptativa deve ser realizada. Além disso, caso exista a documentação das funcionalidades impactadas, estas deverão ser atualizadas, caso contrário, se for demandada a redocumentação dessas funcionalidades, deve-se adicionar ao FI um fator de redocumentação de 15%, conforme descrito na seção 4.2.

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7 – seção 6.2).

4.9. Apuração especial

São funcionalidades executadas apenas uma vez para:

- Corrigir problemas de dados incorretos na base de dados das aplicações ou atualizar dados em bases de dados de aplicações, detalhados na subseção 4.9.1;
- Gerar um relatório específico ou arquivo para o usuário por meio de recuperação de informações nas bases da aplicação, detalhados na subseção 4.9.2;

- A subseção 4.9.3 considera os casos de reexecução de uma apuração especial.

Caso a apuração seja de correção de dados devido a erros de funcionalidades de aplicações desenvolvidas pela contratada, observar as cláusulas contratuais com relação a garantias e prazos de correção.

Recomenda-se fortemente ao contratante sempre solicitar formalmente para a empresa contratada o armazenamento do script para permitir posterior reexecução.

Cabe ressaltar que é necessário avaliar a complexidade das demandas típicas de apuração especial, podendo utilizar um percentual redutor nas fórmulas descritas nas subseções seguintes. Por exemplo, o redutor percentual pode ser aplicado em função da complexidade das demandas, documentação demandada e/ou do processo de desenvolvimento utilizado.

4.9.1. Apuração especial – Base de dados:

Este tipo de apuração especial é uma iniciativa 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 de valor “sim” ou “não” no campo “indicador de matriz” referente ao CNPJ. Normalmente, nesse tipo de atualização são afetados múltiplos registros. Nestes casos, considera-se 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 Internos.

Foram identificados três tipos de Apuração Especial - Base de Dados, cujas fórmulas de cálculo são apresentadas a seguir:

a) Atualização de dados sem consulta prévia:

$$PF_APURAÇÃO_BD = PF_INCLUÍDO$$

b) Consulta prévia sem atualização:

Em alguns casos de Apuração Especial – Base de Dados, o usuário solicita uma consulta prévia das informações. Deve-se ressaltar que essa consulta deve ser realizada antes da construção da funcionalidade, não se trata de homologação. A consulta prévia não é definida pela empresa contratada, obrigatoriamente essa deve ser solicitada pelo contratante 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 a seguir:

$$PF_CONSULTA_PRÉVIA = PF_INCLUÍDO$$

c) Atualização de dados com consulta prévia:

Caso a Apuração Especial - Base de Dados seja solicitada após uma demanda de consulta prévia, deve-se aplicar um fator de 60% na fórmula de contagem da Apuração Especial - Base de Dados, conforme fórmula a seguir:

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

4.9.2. Apuração especial – Geração de relatórios:

Este tipo de apuração especial é uma iniciativa 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 essas funções transacionais são Saídas Externas, devido à atualização do Arquivo Lógico Interno.

Deve-se destacar que essas 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.

$$PF_APURAÇÃO_RELATÓRIOS = PF_INCLUÍDO$$

4.9.3. Apuração especial – Reexecução:

Em alguns casos, a empresa contratante pode ter interesse em executar uma apuração especial mais de uma vez. Nestes casos, ela 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 redutor de 10% na contagem de pontos de função da apuração especial em questão, da seguinte maneira:

$$PF_REEXECUÇÃO_APURAÇÃO = PF_NÃO_AJUSTADO \times 0,10$$

4.10. Atualização de dados

Em alguns casos, as demandas de correção de problemas em base de dados estão associadas a atualizações manuais (de forma interativa), diretamente no banco de dados em um único registro, e que não envolvem cálculos ou procedimentos complexos. São exemplos desse tipo de demanda, a atualização do valor de um campo de uma tabela cadastrado erroneamente ou a exclusão de um registro de uma tabela.

Nestes casos, a aferição do tamanho em Pontos de Função deve considerar 10% do PF de uma Entrada Externa e os Tipos de Dados da Entrada

Externa são todos os TD considerados na funcionalidade – campos atualizados e campos utilizados para a seleção do registro.

$$PF_ATUALIZAÇÃO_BD = PF_INCLUÍDO \times 0,10$$

Deve-se ressaltar que neste tipo de demanda não há gestão de configuração (armazenamento de script, versionamento, etc) das atualizações. Caso a contratante identifique a necessidade de realização de gestão de configuração das atualizações no banco de dados, então a demanda será classificada como Apuração Especial - Base de Dados (subseção 4.9.1).

4.11. Desenvolvimento, manutenção e publicação de páginas estáticas de intranet, internet ou portal

Nesta seção são tratados desenvolvimentos e manutenções específicas em páginas estáticas de portais, intranets ou websites. As demandas desta seção abrangem a publicação de páginas Web com conteúdo estático. Por exemplo: criação de página *HTML*, atualização de menu estático, atualização de texto ou banner estáticos em páginas *HTML* existentes.

Caso o desenvolvimento de páginas estáticas esteja contido em uma iniciativa de desenvolvimento, então elas serão contabilizadas na iniciativa de desenvolvimento e não devem ser mensuradas em separado. Ou seja, esta seção 4.11 se aplica quando ocorrer a demanda exclusivamente para o desenvolvimento ou manutenção de páginas estáticas.

Estas demandas são consideradas como desenvolvimento de consultas. Nestes casos, considera-se 20% dos pontos de função das consultas desenvolvidas. Cada página é contada como uma consulta. As consultas são consideradas consultas externas simples (3 PF). Ou seja, 0,6 PF por cada página desenvolvida ou mantida, de acordo com a fórmula a seguir:

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

É recomendada a construção de portais com ferramentas que apoiem a construção de conteúdo pelo usuário, os chamados Gerenciadores de Conteúdo, de modo a minimizar as demandas de criação de páginas estáticas.

4.12. Verificação de erros

As verificações de erro ou análise e solução de problemas são as demandas referentes a todo comportamento anormal ou indevido apontado pelo cliente nos sistemas aplicativos. Neste caso, a equipe de desenvolvimento da contratada se mobilizará para encontrar as causas do problema ocorrido. Se for constatado algum erro de sistema, a demanda será atendida como manutenção corretiva (seção 4.4).

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 que o cliente reportou erro. Será considerado 20% do tamanho funcional dessas funcionalidades com solicitação de análise pelo contratante, segundo a fórmula a seguir:

$$PF_VERIFICAÇÃO = PF_Funcionalidade_Reportada_Com_Erro \times 0,20$$

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 de rede ou banco de dados devem ser tratados como serviços de suporte e não serviços de desenvolvimento e manutenção de sistemas. Esses serviços de suporte não fazem parte do escopo desse guia de métricas, não se aplicando verificação de erros nestes casos.

4.13. Pontos de função de teste

Muitas vezes, em iniciativas de manutenção, o conjunto de funções transacionais a serem testadas é maior do que a quantidade de funções a serem implementadas, isto é, além das funcionalidades que são afetadas diretamente

pela iniciativa de manutenção, outras precisam ser testadas. O tamanho das funções a serem apenas testadas deve ser aferido em Pontos de Função de Teste (PFT). Não considerar as funcionalidades incluídas, alteradas ou excluídas da iniciativa de manutenção na contagem de Pontos de Função de Teste.

A contagem de PFT será o somatório dos tamanhos em pontos de função das funções transacionais envolvidas no teste:

PFT = Somatório dos Tamanhos das Funções Transacionais Testadas

A conversão do PFT em ponto de função deve ser feita de acordo com a fórmula a seguir:

PF_TESTES = PFT X 0,20

É importante ressaltar que no caso de uma função ser testada várias vezes, com cenários diferentes, a função só pode ser contada uma vez. Outra observação é que as funções testadas, previstas no PFT, devem ser registradas pela contratada considerando-se a documentação de testes definida no processo de desenvolvimento da contratante.

4.14. Componente interno reusável

Em alguns casos são demandadas manutenções em componentes, que implementam regras de negócio, específicos de uma aplicação e estes são reusados por várias funcionalidades da aplicação. Por exemplo, uma mudança em uma rotina de validação de um CPF usada em várias funcionalidades de cadastro. Se considerarmos o método de contagem de iniciativas de melhoria do *CPM*, seriam contadas todas as funcionalidades impactadas por essa mudança.

No entanto, este guia propõe que o componente, o qual deverá ser testado, seja considerado como um processo elementar independente e sua alteração seja contada aplicando-se um fator de impacto (FI) sobre o PF_ALTERADO, seguindo os conceitos do *CPM* 4.3.1, apresentados na seção 4.2 – Iniciativa de Melhoria. 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 Teste proposta na seção 4.14.

$$PF_COMPONENTE = FI \times PF_ALTERADO$$

Exemplo de manutenção de componentes:

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

Além do item descrito anteriormente, existem casos onde são realizadas manutenções de valores de elementos internos de configuração que afetam o comportamento ou a apresentação do sistema de forma geral, tais como páginas de estilos (arquivos CSS de sistemas Web), arquivos com mensagens de erro, arquivos de configuração de sistema e arquivos de internacionalização. Nestes casos, a aferição do tamanho em pontos de função será realizada com a aplicação de um fator de redução de modo a considerar 20% da contagem de uma função transacional de mais baixa complexidade (3 PF), ou seja 0,6 PF. Assim sendo, deve ser utilizada a seguinte fórmula de cálculo:

$$PF_COMPONENTE_ARQUIVO = 0,6 PF \times QTD_ARQUIVOS_ALTERADOS$$

4.15. Dados de Código (*code tables*)

Os dados de código em geral estão relacionados a tabelas de domínio que são utilizadas pela aplicação para decodificar valores, metadados, dados e tabelas estáticas. Não raramente dados de código também estão associados a utilização de estruturas adicionais de banco tanto para oferecer uma maior performance a aplicação, na emissão de relatórios (exemplo: tabela de vendas diárias derivada de tabela de vendas onde a emissão de relatórios enxerga a tabela de vendas diária), como para guardar dados de configuração técnica (dados de IP, porta de conexão, string de conexão de banco de dados, etc).

O manual do IFPUG aborda em um capítulo inteiro o assunto Dados de Código, e é taxativo em afirmar que Dados de Código são relativos a requisitos

não funcionais e que, portanto, não devem ser medidos como parte do tamanho funcional.

Para o contexto de um contrato de desenvolvimento/manutenção de sistemas baseado em PF, é comum que o escopo do trabalho do fornecedor em uma determinada demanda, abranja Dados de Código. Em sendo o escopo desta demanda a criação de novas funcionalidades, e derivadas delas a criação de dados de código, então nada deve ser medido para a parte de dados de código. A remuneração do fornecedor é feita pela medição exclusiva dos PF e o trabalho de dados de código é remunerado indiretamente pelo preço do PF do contrato.

Em sendo o escopo de trabalho da demanda a criação ou manutenção de dados de código que não estejam associados a requisitos funcionais novos na mesma demanda, então haverá uma medição a ser aplicada à parte de dados de código, conforme definido a seguir. A eventual alteração de conteúdo ou criação de listas estáticas também deverá ser medida da mesma forma.

PF_CODETABLE = 0,3 PF x QUANTIDADE DE TABELAS E TRANSAÇÕES DE MANUTENÇÃO/CONSULTA AOS DADOS INCLUÍDAS, ALTERADAS E EXCLUÍDAS

5. Orientações complementares para contagem

Este capítulo tem o propósito de apresentar diretrizes complementares ao Manual de Práticas de Contagem do IFPUG para contagens de pontos de função e reforçar pontos sensíveis nas contratações atuais que podem impactar significativamente o resultado de uma contagem em caso de falhas.

5.1. Contagem de pontos de função com múltiplas mídias

A contagem de PF de funcionalidades entregues em mais de uma mídia, na 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*.

É importante enfatizar que o IFPUG reconhece ambas abordagens, *single instance* e *multiple instance*, para a aplicação das regras definidas no CPM. A determinação da contagem de PF seguindo a abordagem *multiple instance* ou *single instance* depende da avaliação dos especialistas em contagem da contratante em acordo com os especialistas da contratada.

As estimativas e contagens de PF abordadas neste documento são baseadas em *multiple instance*, com exceção dos casos de consultas em .pdf, .doc, .xls e consultas idênticas em tela e papel, que serão consideradas uma única funcionalidade.

A seguir são descritos os termos comuns definidos pelo:

- Canal: também se refere a mídia. Múltiplos canais é sinônimo de múltiplas mídias.
- Mídia: descreve a maneira como 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, apenas 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 funcionalidade, 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 do objetivo da contagem, permitindo uma função de negócio ser reconhecida no contexto das mídias que são requisitadas para que a funcionalidade seja entregue. A abordagem *multiple instance* reconhece que a mídia para entrega constitui uma característica de diferenciação na identificação da unicidade da função transacional.

Os cenários descritos nas seções seguintes não representam uma lista completa de situações de múltiplas mídias. O entendimento dos exemplos a seguir facilitará o entendimento de outros cenários envolvendo múltiplas mídias. Deve-se atentar que o cenário de múltiplas mídias existe somente no contexto de um único sistema (fronteira). Quando há mais de uma fronteira envolvida, conta-se a função em cada um dos sistemas.

5.1.1. 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, sugere-se a abordagem *single instance*, considerando que dados idênticos sendo apresentados em tela e em 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 (*multiple instance*). Neste caso, duas funções são contadas: apresentação de dados em tela e apresentação de dados impressos.

5.1.2. 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 às gravadas no arquivo.

Nesses casos, sugere-se a utilização da 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. Além disso, 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*.

5.1.3. 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, da mesma forma que o processamento da entrada *on-line* também executa validações das informações. Neste caso, sugere-se a utilização da 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*.

5.1.4. Cenário 4 - Múltiplos canais de entrega da mesma funcionalidade:

Numa estratégia de desenvolvimento de software orientada a serviços, é comum que a camada de backend seja toda exposta para consumo por sistemas externos, além de ser também consumida pela sua camada de front-end. Em não existindo diferença de lógica ou campos quando a funcionalidade é acionada via tela por um usuário humano ou via sistema pelo webservice, deve-se contar sempre uma única função.

5.1.5. Cenário 5 - Relatório em múltiplos formatos:

Um relatório deve ser entregue em diferentes formatos, por exemplo: um arquivo *html* e um arquivo com valores separados por vírgula (.csv).

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. 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 se contará apenas uma vez (*single instance*), observando que a funcionalidade será da ferramenta e não da aplicação.

5.2. Log, trilha de auditoria e histórico

O objetivo dessa sessão é descrever orientações sucintas a respeito de contagem de log, trilha de auditoria e histórico.

5.2.1. Log:

Conceituamos o termo “Log” como o registro de procedimentos ou ações realizadas pela aplicação, em determinado período de tempo, com o objetivo de apoiar a auditoria do ambiente tecnológico e a identificação das causas raízes de falhas em sistemas. Diante desse conceito, definimos que o Log não deve ser mensurado com Pontos de Função, já que ele não armazena informações negociais reconhecidas pelo usuário da aplicação.

5.2.2. Trilha de auditoria:

Conceituamos “Trilha de Auditoria” como a funcionalidade que tem o objetivo de armazenar informações referentes às ações realizadas pelos usuários da aplicação no passado, de modo que seja possível apurar quais foram as ações executadas quando da utilização do sistema.

Para isso, devem existir no mínimo as informações para identificar quem realizou a ação, quando e o que foi realizado, além de outras informações que o usuário da aplicação defina como necessárias.

A trilha de auditoria deve ser solicitada pelo usuário da aplicação e, para a contagem, deve existir funcionalidade de consulta a tais dados.

Caso a trilha de auditoria faça parte da política corporativa de segurança da informação do contratante, ela deve ser considerada como um requisito não funcional e, portanto, não será mensurável em ponto de função.

Diante do exposto, a principal diferença entre o Log e a Trilha de Auditoria é:

- Log: apoia a coleta de informações no âmbito tecnológico, ou seja, em problemas decorrentes da arquitetura tecnológica que precisam ser investigados, por meio da análise do conjunto de procedimentos executadas pela aplicação, como exemplo a baixa performance no sistema, travamentos e outros comportamentos inesperados.
- Trilha de Auditoria: apoia a auditoria para os dados de negócio, armazenando informações das ações realizadas pelo usuário na aplicação.

5.2.3. Histórico:

Conceituamos “Histórico” como um registro de estados com informações anteriores de um registro em determinado momento. O usuário poderá consultar a evolução dessas informações em uma linha do tempo e sua existência é justificada pelo negócio. Assim, para fazer parte do tamanho funcional, deve ser solicitado pelo gestor e deverá existir funcionalidade de consulta a tais dados.

A função de consulta aos dados de um histórico deverá ser contada de acordo com as regras de contagem das funções transacionais do CPM.

Não devem ser consideradas na contagem funções de transação separadas para incluir, alterar e excluir as informações históricas, pois o armazenamento dessas informações é parte integrante das mesmas funcionalidades que processam os dados de negócio. Apenas quando o histórico for mantido de forma independente do registro principal, por exemplo no caso do ALI principal ter sido excluído, o histórico se torna um ALI independente e não um registro lógico do ALI relacionado.

5.3. Arquivos para processamento

É um requisito bastante comum que uma aplicação se comunique com outras trocando dados através de arquivos (neste caso o termo arquivo é o de um arquivo para o sistema operacional, não de arquivo lógico para a APF). Estes arquivos possuem leiaute pré-definido e acordado entre as aplicações e podem ser arquivos texto com campos de tamanho fixo, arquivos texto com campos delimitados por caracteres especiais, arquivos XML, etc. Um exemplo é a Solução Integradora que recebe mensalmente dados de produção de atendimento dos vários Departamentos Regionais através do envio de arquivos CSV por cada DR. Provavelmente, cada DR gera o arquivo a ser enviado à Solução Integradora através de um sistema distinto.

O primeiro passo na análise deste cenário é delimitar as fronteiras dos sistemas envolvidos: a Solução Integradora e o sistema do DR que produz o arquivo a ser carregado.

O arquivo gerado para processamento não deve ser contado como arquivo lógico (nem ALI, nem AIE) em nenhum dos sistemas, pois não representa requisitos de armazenamento de nenhum deles, mas sim requisitos de processamento que serão medidos como transações.

Do ponto de vista do sistema do DR (origem) que produz o arquivo para envio à Solução Integradora existe meramente um relatório, que em vez de impresso, está em formato CSV. Mede-se a transação que é responsável por gerar o arquivo como uma CE ou SE.

Do ponto de vista da Solução Integradora (destino), o processamento do arquivo carregado é classificado como EE, que cumpre o papel de uma tela em que o usuário poderia digitar os dados presentes no arquivo carregado.

Atenção especial quando o processamento do arquivo recebido contiver várias lógicas de processamento distintas para cada tipo de informação a ser processada. Nesse caso, podem existir várias EE, em vez de uma única. Exemplo: o arquivo de produção de serviços de Atendimentos de Tecnologia e Inovação processado pela Solução Integradora possui cinco tipos de registro: atendimento, produção, cliente PF, cliente PJ e parceiros. Cada tipo de registro possui um conjunto distinto de atributos e validações específicas. Portanto o processamento do arquivo abrangeria cinco EEs. No entanto, três destes tipos de registro permitem dois tipos de operação distintas no processamento: inclusão de registro e exclusão de registro. Então, para cada um dos três tipos de registro com duas operações possíveis se contaria uma EE, somando com a EE de processamento dos outros dois tipos de registro que possuem um único tipo de operação, chega-se a um total de 8 EEs.

Neste exemplo do tipo de operação da Solução Integradora, o tipo de operação com valor branco indica que o registro será incluído (caso não exista) ou atualizado (caso já exista), porém na ótica da APF conta-se uma única transação, pois a decisão de que ação tomar é interna à transação, não do ponto de vista externo (do usuário). Caso houvesse a distinção explícita dos tipos de operação inclusão e alteração na definição de leiaute do arquivo, aí sim se contaria duas transações.

Um requisito comum no processamento de arquivo é o de gerar um relatório de erros ou de críticas com as situações de erro ou exceção ocorridos durante o processamento do arquivo recebido. Nesse caso, o relatório é parte do requisito de processamento do arquivo e deve ser contado como parte da EE original. Os erros listados neste relatório são como mensagens de erro em uma tela de entrada de dados.

5.4. API/Webservices

Uma API (ou Interface de Programação de Aplicativos) é um conjunto de rotinas e padrões estabelecidos por um software para uso de suas funções por programas que não querem se envolver nos detalhes de implementação, apenas usar seus serviços. A API é composta por funções acessíveis somente via programação e que permitem usar características do software às vezes não disponível ao usuário final. Um webservice é um exemplo de API que permite a interação entre sistemas na plataforma web. Uma DLL no sistema operacional Windows é um exemplo de API que possibilita a interação de um aplicativo com outro.

Delimitando o contexto, há um sistema que provê serviços via API e outro que consome estes serviços. Do ponto de vista da APF, o consumidor da API é um usuário do primeiro sistema. Porém há casos em que a API é criada exclusivamente para consumo do próprio sistema. Por exemplo, a lógica de cálculo do dígito verificador do CPF foi implementada numa API do sistema e é consumida somente por ele. Neste caso não há funções a serem medidas relativas à API, pois um sistema não pode ser usuário de si mesmo.

Do ponto de vista do sistema usuário da API, não há medição de funções de transação para a API. Há casos em que o sistema usa a API com a finalidade de referenciar dados que estão armazenados no sistema origem. Se o uso da API foi por uma decisão de implementação, e um acesso direto à tabela origem dos dados também funcionasse, os dados referenciados via API são contados como AIE no sistema usuário. Se os dados consumidos não puderem ser acessados diretamente porque é necessário conhecer regras do negócio da origem para derivar os dados, então não se mede estes dados como AIE. Neste caso e para todas as outras situações de uso da API pelo sistema consumidor, a influência na medição funcional do consumidor se dará apenas na complexidade da sua transação que dispara o uso da API, considerando os campos passados para e recebidos da API como tipos de dados desta transação.

Do ponto de vista do sistema origem que fornece a API, as funções expostas ao mundo externo serão medidas como transações, conforme sua intenção principal seja atualizar dados internos (EE) ou apresentar dados ao

exterior (CE ou SE). Na avaliação da complexidade da transação, considera-se os parâmetros de entrada e saída como seus tipos de dados (sem repetição).

Exemplo 1: Os Correios possuem uma função implementada através de um webservice que fornece o logradouro, bairro, cidade, estado de um determinado CEP. Esta função será medida como uma CE para o sistema dos Correios, pois é uma simples recuperação de dados e os parâmetros do webservice contam como seus tipos de dados. Na loja virtual Americanas.com, durante o registro de um novo cliente, ao usuário informar o CEP do cliente, o sistema de comércio eletrônico invoca o serviço dos Correios para recuperar o logradouro, bairro, cidade e estado. Para o sistema de comércio eletrônico da Americanas.com, a função Registrar Cliente será contada como uma EE, e na classificação de sua complexidade, o AIE CEP será contado como mais um arquivo referenciado. A passagem e recebimento de parâmetros para o webservice não computam como seus tipos de dados.

Exemplo 2: Os Correios possuem uma função implementada através de um webservice que fornece o valor do Sedex, dados o CEP origem e destino, peso e dimensões da encomenda. Assumindo que o valor do serviço é fruto de um cálculo, esta função será contada como uma SE para o sistema dos Correios, e os parâmetros do webservice contam como seus tipos de dados. Na loja virtual Americanas.com, durante o processo de compra, o cliente é informado do valor da remessa da sua encomenda via Sedex dos Correios. Para o sistema de comércio eletrônico da Americanas.com, a função Compra será contada como uma EE. Durante a compra há o acionamento do webservice dos Correios que fornece o valor do Sedex. A transação continua a ser a mesma (Compra), e na classificação de sua complexidade, os campos informados e recebidos dos Correios serão contados como seus tipos de dados. Não se mede os dados derivados do Sedex como AIE.

Exemplo 3: O sistema de Internet Banking necessita atualizar o saldo da conta corrente sempre que o cliente efetuar um pagamento de boleto. Porém o sistema responsável pelo saldo da conta corrente é o Conta Corrente. Atualizar o saldo de uma conta corrente compreende diversas regras de negócio, por exemplo: saber o saldo disponível para a operação envolve considerar o saldo

das aplicações com resgate automático e o limite do cheque especial; toda atualização de saldo deve também gerar um lançamento de movimentação da conta corrente. Enfim, todo sistema que pretendesse atualizar o saldo da conta diretamente deveria ter completo conhecimento das regras de negócio envolvidas com a sensibilização do saldo, o que na prática é inviável. O Conta Corrente fornece então uma função para atualização de saldo da conta corrente que pode ser usada por quaisquer outros sistemas sem que estes precisem conhecer as regras de negócio envolvidas. Para a transação Pagar Boleto (EE) do Internet Banking o uso da função de atualização de saldo implica na contagem de tipos de dados adicionais relativos aos parâmetros do webservice.

5.5.Principais falhas da contagem identificadas

A listagem abaixo apresenta situações que constituem falhas de contagem:

- Contar DER (Dado Elementar Referenciado) para variáveis de paginação, número de páginas e informação de posicionamento;
- Contar DER para ajudas de navegação como habilidade de navegar com uma lista utilizando “anterior”, “próximo”, “primeiro”, “último” e seus equivalentes gráficos;
- Contar mais de um DER por função de transação para a habilidade de iniciar ações quando há múltiplos meios para isso;
- Contar mais de um DER quando a função de transação tem capacidade de enviar várias mensagens;
- Considerar fluxos alternativos como processos elementares distintos não levando em consideração as regras de identificação e unidade do processo elementar;
- Considerar transações que atualizam dados de código;
- Identificação de um ALI para armazenamento de dados de código;
- Relacionar processos elementares com telas ou abas de uma transação não levando em consideração as regras de identificação e unicidade do processo elementar.

6. Regras auxiliares

Este capítulo visa compilar as regras auxiliares deste guia para sua perfeita aplicação, trazendo assim indicações complementares para a realização das contagens sob responsabilidade da STI.

6.1. Contagem estimativa de pontos de função (CEPF)

Esta seção tem o objetivo de suportar e normatizar ações da STI que visam apoiar processo decisório que, para sua realização, necessita de estimativa do tamanho do desenvolvimento ou manutenção que se pretende realizar. A partir das regras descritas nesta seção será permitida a definição de uma estimativa, em pontos de função, da forma mais eficiente e coerente possível.

A Contagem Estimativa de Pontos de Função – CEPF, significa fornecer uma avaliação aproximada do tamanho de um software utilizando métodos diferentes da contagem de pontos de função do IFPUG.

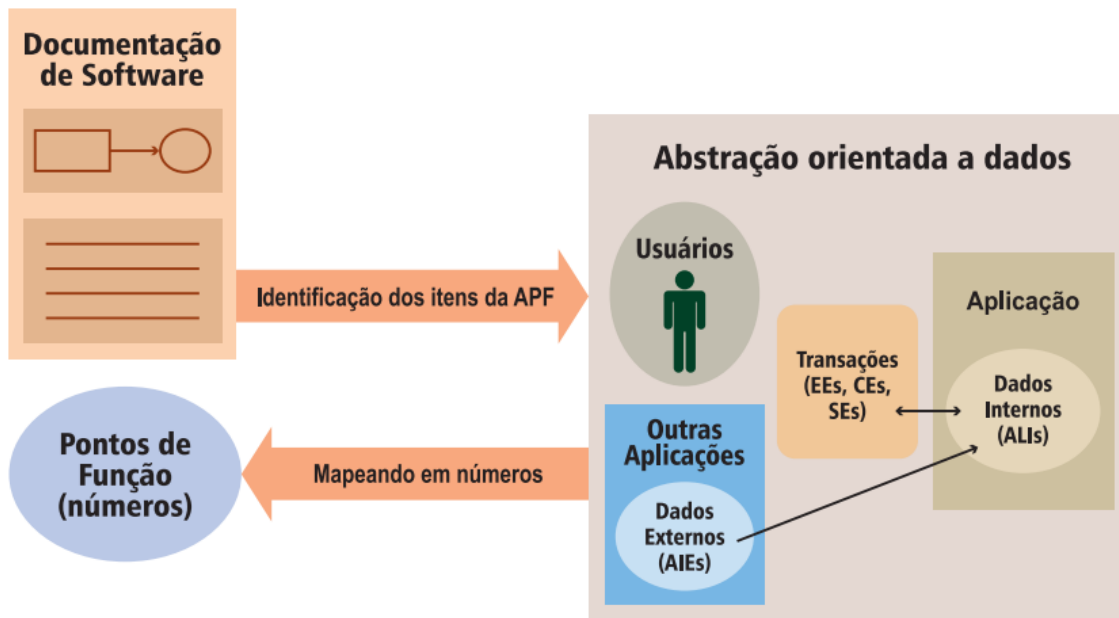
A CEPF foi definida com base nas diretrizes adotadas no método Contagem Estimada de Pontos de Função da NESMA [NESMA, 2005]. A diferença é que o método da NESMA não recomenda a análise da complexidade das funções identificadas, considerando todas as funções de dados identificadas com complexidade Baixa e as funções transacionais com complexidade Média.

Primeiramente, os requisitos funcionais iniciais do sistema 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 (Tabela 1).

O estimador deve realizar uma leitura do 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. 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.

FIGURA 2
Modelo Lógico da Análise de Pontos de Função



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 a iniciativa, contidas no documento inicial de requisitos, devem ser enquadradas em uma das seguintes tabelas:

- a) Tabela 2 – Contagem dos Arquivos Lógicos Internos (ALI): banco de dados lógico da aplicação (tabelas e arquivos mantidos pela aplicação).
- Considerações: É necessário identificar 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. Arquivos físicos, arquivos de índices, arquivos de trabalho e tabelas de relacionamento sem atributos próprios (tabelas que existem para quebrar o relacionamento $m \times n$ e apenas transportam as chaves estrangeiras) devem ser desconsiderados. As entidades fracas também não são consideradas um ALI. É opcional a identificação dos 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 ALI dos sistemas são de complexidade Baixa.

Tabela 2: Identificação dos Arquivos Lógicos Internos da Aplicação

Nº ALI Baixa:	X 7 PF
Nº ALI Média:	X 10 PF
Nº ALI Alta:	X 15 PF
Total PF:	

- b) Tabela 3 – Contagem de Arquivos de Interface Externa (AIE): banco de dados de outras aplicações, apenas referenciados pela aplicação que está sendo estimada (tabelas e arquivos mantidos por outra aplicação).
- Considerações: É necessário identificar 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 AIE. Não são considerados AIE arquivos físicos, arquivos de índice, arquivos de trabalho, tabelas de relacionamento sem atributos próprios e entidades fracas. Geralmente, os AIE 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.

Tabela 3: Identificação dos Arquivos de Interface Externa da Aplicação

Nº AIE Baixa:	X 5 PF
Nº AIE Média:	X 7 PF
Nº AIE Alta:	X 10 PF
Total PF:	

c) Tabela 4 - Contagem de Entradas Externas (EE): funcionalidades que mantêm os Arquivos Lógicos Internos (ALI) ou alteram o comportamento da aplicação.

- Considerações: É necessário identificar as funcionalidades de manutenção de dados. Inclusão, alteração e exclusão de dados, isto é, cada função independente de inclusão, alteração ou exclusão devem ser contadas separadamente. Caso a aplicação possua funções de entrada de dados que alteram o comportamento dela, por exemplo: processamentos batch ou processamento de informações de controle, estas funções também devem ser identificadas como Entradas Externas. Não sendo possível definir a complexidade, considere as Entradas Externas identificadas com complexidade Média.

Tabela 4: Identificação das Entradas Externas da Aplicação

Nº EE Baixa:	X 3 PF
Nº EE Média:	X 4 PF
Nº EE Alta:	X 6 PF
Total PF:	

d) Tabela 5 – Contagem de Consultas Externas (CE): 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: Caso a função vise apresentar informações para o usuário (uma consulta, relatório, listbox, download, geração de um arquivo, geração de arquivo pdf, xls), caso não possua cálculos ou algoritmos para derivação dos dados referenciados, caso não altere um Arquivo Lógico Interno e nem mude o comportamento do sistema, esta função deve ser identificada como Consultas Externa. Não sendo possível definir a complexidade, considere as Consultas Externas identificadas com complexidade Média.

Tabela 5: Identificação das Consultas Externas da Aplicação

Nº CE Baixa:	X 3 PF
Nº CE Média:	X 4 PF
Nº CE Alta:	X 6 PF
Total PF:	

e) Tabela 6 - Contagem de Saídas Externas (SE): 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 com

atualização log, downloads com cálculo de percentual, entre outros.

- Considerações: Caso a função vise apresentar informações para o usuário que necessitarão ser calculadas, de alguma forma (uma consulta ou relatório com totalização de dados, etiquetas de código de barras, gráficos, relatórios estatísticos, download com percentual calculado, geração de arquivo com atualização de log) esta função deve ser identificada como Saída Externa. 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. Não sendo possível definir a complexidade, considere as Saídas Externas identificadas com complexidade Média.

Tabela 6: Identificação das Saídas Externas da Aplicação

Nº SE Baixa:	X 4 PF
Nº SE Média:	X 5 PF
Nº SE Alta:	X 7 PF
Total PF:	

A estimativa de tamanho da iniciativa em PF deve ser gerada com a totalização dos PF obtidos nas Tabelas 2, 3, 4, 5 e 6.

6.2. Distribuição de esforço por fase da iniciativa

Na Tabela 7 temos a distribuição de esforços pelas macroatividades (fases) relacionadas ao ciclo de vida de desenvolvimento de um software. Os percentuais indicados nesta tabela são um fator de ponderação a ser aplicado no cálculo dos pontos de função das iniciativas com o objetivo de racionalizar a sua aferição, prevendo assim somente a remuneração das ações que foram efetivamente empregadas na iniciativa.

Tabela 7: Distribuição de Esforço por Macroatividades da Iniciativa

Macroatividades do Processo de Desenvolvimento de Software	Percentual de Esforço (%)
Refino	20%
Implementação	60%
Teste	20%

6.2.1. Evidências de entrega:

Para caracterizar a efetiva entrega de cada uma das 3 fases descritas na tabela 7 é necessário convencionar o que será aceito como entrega.

Salvo especificidades que podem ser inseridas no contexto de gestão e contratação de determinados sistemas, considera-se os seguintes cenários como padrões de evidências de entrega e, por consequência, como critérios válidos para aplicação do fator de ponderação em torno dos percentuais de esforço para remuneração:

- Refino: Demanda refinada e com história de usuário e seus critérios de aceitação.
- Implementação: Código fonte versionado e no padrão de codificação entregue no repositório indicado.
- Teste: Evidência de teste para cada um dos critérios de aceitação da história.

6.2.2. Estratégias de desenvolvimento de *back* e *frontend* em separado:

Nos casos em que o cliente aprovar estratégia de desenvolvimento que divida em Sprints distintas as camadas de uma funcionalidade (por exemplo, primeira realiza-se o desenvolvimento e entrega do *backend*, e, depois realiza-se o desenvolvimento do *frontend*), para efeito de remuneração parcial deve-se adotar uma relação de 60% para *backend* e 40% para *frontend* considerando-se somente o percentual da macroatividade de Implementação.

6.3. Fator de ajuste

O fator de ajuste não faz mais parte do processo de medição funcional aderente à ISO/IEC 14133, contudo, ainda é parte do *CPM 4.3.1* como um apêndice, para manter a compatibilidade com aqueles usuários que usam pontos de função com a sua aplicação.

Complementando este caráter opcional de sua aplicação, o que acusa algum nível de obsolescência da mesma, também podemos considerar que esta mesma técnica apresenta alto grau de subjetividade, inter-relacionamento entre variáveis, incompletude de questões técnicas, conceituais e metodológicas envolvidas na sua aplicação e não traz benefícios para otimizar questões relacionadas a estimativas de esforço. Por isso, este guia segue recomendação da comunidade de contagem de pontos de função e assim determina o não uso de fator de ajuste nas contagens sob responsabilidade da STI.

7. Referências bibliográficas

[SISP,2018] Brasil. Ministério do Planejamento, Desenvolvimento e Gestão. Secretaria de Tecnologia da Informação e Comunicação. Roteiro de Métricas de Software do SISP: versão 2.3 / Ministério do Planejamento, Desenvolvimento e Gestão. Secretaria de Tecnologia da Informação e Comunicação - Setic. – Brasília: MP, 2018. 96 p.: il.

[IFPUG,2010b] IFPUG. Counting Practices Manual. Version 4.3, January, 2010

[Vazquez, 2013] VAZQUEZ, C. et al. Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software. 13ª Edição, Editora Érica Ltda, São Paulo, 2012.

8. Controle de versão

Revisão	Data	Descrição	Elaboração
Versão 1.0	28/10/2019	Elaboração do documento	<p>Gilcimar Francisco Dias – Gerência de Soluções do Negócio STI</p> <p>Giovanni Giazzon dos Santos – Gerência de Soluções do Negócio STI</p> <p>Guilherme Siqueira Simões – Consultoria FATTO</p> <p>Fabio Leandro Bernardes Duarte – Gerência de Soluções do Negócio STI</p> <p>Fernando Cesar Kurimori – Gerência de Governança STI</p> <p>Jemima da Costa Mendes - Gerência de Governança STI</p> <p>José Francisco Gonçalves Júnior – Gerência de Soluções do Negócio STI</p>