

Roteiro de Métrica - Normas complementares ao CPM- IFPUG

Outubro de 2016
Versão 1.0 – 01/10/2016

SUMÁRIO

SUMÁRIO

1. OBJETIVO DO DOCUMENTO	4
2. DAS CONTAGENS ESTIMATIVAS	4
3. DAS CONTAGENS DETALHADAS	4
4. PROJETO DE DESENVOLVIMENTO	5
5. PROJETO DE MELHORIA	6
6. MANUTENÇÃO CORRETIVA	6
7. COMPONENTE INTERNO REUSÁVEL	7
8. ADAPTAÇÃO EM FUNCIONALIDADES SEM ALTERAÇÃO DE REQUISITOS FUNCIONAIS	7
9. MANUTENÇÃO DE DOCUMENTAÇÃO DE SISTEMAS LEGADOS	8
10. ATUALIZAÇÃO DE DADOS	9
11. MANUTENÇÃO EM INTERFACE	9
12. APURAÇÃO ESPECIAL	10
13. APURAÇÃO ESPECIAL - BASE DE DADOS	10
13.6. APURAÇÃO ESPECIAL – BASE DE DADOS – SEM CONSULTA PRÉVIA E COM ATUALIZAÇÃO DE DADOS	11
13.7. APURAÇÃO ESPECIAL – BASE DE DADOS – COM CONSULTA PRÉVIA SEM ATUALIZAÇÃO DE DADOS	11
13.8. APURAÇÃO ESPECIAL – BASE DE DADOS – COM CONSULTA PRÉVIA E COM ATUALIZAÇÃO DE DADOS	11
14. APURAÇÃO ESPECIAL – GERAÇÃO DE RELATÓRIOS	12
15. APURAÇÃO ESPECIAL – REEXECUÇÃO	12
16. DESENVOLVIMENTO, MANUTENÇÃO E PUBLICAÇÃO DE PÁGINAS ESTÁTICAS DE INTRANET, INTERNET ou PORTAL	12
17. VERIFICAÇÃO DE ERROS	13
18. PONTOS DE FUNÇÃO DE TESTE	13

19.	PROJETO DE MIGRAÇÃO DE DADOS.....	14
20.	ORIENTAÇÕES COMPLEMENTARES PARA CONTAGEM	14
I.	REQUISITO FUNCIONAL	14
II.	LOG, TRILHA DE AUDITORIA E HISTÓRICO	15
III.	IDENTIFICAÇÃO DE PROCESSO ELEMENTAR	16
IV.	“COMPLETO”	17
21.	CONTAGEM DE PONTOS DE FUNÇÃO DE DESENVOLVIMENTO DE SOFTWARE USANDO MÉTODOS ÁGEIS.....	19

1. OBJETIVO DO DOCUMENTO

- 1.1. O objetivo deste documento é estabelecer um Roteiro de Métricas sobre a aplicação da Análise de Pontos de Função nos projetos de desenvolvimento e manutenção de software por aquisição, com base nas regras de contagem de pontos de função do Manual de Práticas de contagem (CPM 4.3.1) e no Roteiro de Métricas do SISP versão 2.2.
- 1.2. Este roteiro deverá ser utilizado pela CONTRATADA de modo a nortear a aplicação dos métodos de contagens estimadas antecipadas (NESMA indicativa e estimativa) e posteriores contagens detalhadas e estando dessa forma alinhada com o praticado pela RESPONSÁVEL TÉCNICA.

2. DAS CONTAGENS ESTIMATIVAS

- 2.1. Os métodos de contagem dos Pontos de Função a serem utilizados nas contagens estimativas são:
 - a) Para a ESTIMATIVA INICIAL deverá ser utilizada o método de NESMA INDICATIVA ou NESMA ESTIMATIVA a ser realizado pela CONTRATADA ou RESPONSÁVEL TÉCNICA.
 - b) No método de NESMA INDICATIVA, cabe à RESPONSÁVEL TÉCNICA facultar também a sua utilização na íntegra, conforme fórmula pré-definida baseada somente na quantidade de arquivos lógicos existentes (ALIs e AIEs)
- 2.2. Determina-se a quantidade das funções do tipo dado (ALIs e AIEs)
- 2.3. Calcula-se o total de pontos de função não ajustados da aplicação da seguinte forma: tamanho indicativo (pf) = 35 x número de ALIs + 15 x número de AIEs.
- 2.4. Nas estimativas iniciais de tamanho de projetos de desenvolvimento, após o levantamento do escopo, considerando-se o documento de visão inicial do projeto, recomenda-se utilizar um percentual de 40% (quarenta) para evolução de requisitos.

3. DAS CONTAGENS DETALHADAS

- 3.1. O Manual de Práticas de Contagem do IFPUG define dois tipos de projetos de software, a saber:
 - a) **Projeto de Desenvolvimento:** projeto para desenvolver e entregar a primeira versão de uma aplicação de software. Seu tamanho funcional é a medida das funcionalidades entregues ao usuário no final do projeto. Também considera-se as funcionalidades de conversão de dados, caso seja requisitado no projeto a migração ou carga inicial de dados para a nova aplicação.

- b) **Projeto de Melhoria:** projeto de manutenção evolutiva ou melhoria funcional. Seu tamanho funcional é a medida das funcionalidades incluídas, alteradas e excluídas ao final do projeto. Também considera-se as funcionalidades de conversão de dados, caso seja requisitado a migração ou carga inicial de dados no projeto de melhoria.

3.2. Seguem abaixo as definições dos termos técnicos da Análise de Pontos de Função utilizados nas fórmulas de dimensionamento de projetos de software propostas neste roteiro:

PF_INCLUÍDO	pontos de função associados às novas funcionalidades que farão parte da aplicação após um projeto de desenvolvimento ou de manutenção.
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 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. O PF_CONVERSÃO deve ser suprimido das fórmulas de contagem de pontos de função de projetos de desenvolvimento quando houver um projeto específico de migração de dados. A contabilização dos pontos de função de conversão será realizada conforme as orientações do item 4.3.

4. PROJETO DE DESENVOLVIMENTO

- 4.1. É o projeto para desenvolver e entregar a primeira versão de uma aplicação de software. Seu tamanho funcional é a medida das funcionalidades entregues ao usuário no final do projeto. Também considera-se as funcionalidades de conversão de dados.
- 4.2. A fórmula de cálculo utilizada no dimensionamento de projetos de desenvolvimento de software:

$$PF_DESENVOLVIMENTO = PF_INCLUIDO + PF_CONVERSÃO$$

5. PROJETO DE MELHORIA

5.1. O Projeto de Melhoria (enhancement), também denominado de projeto 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.

5.2. A fórmula de cálculo utilizada no dimensionamento de projetos de melhoria de software:

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

Fator de Impacto (FI): pode variar de 50% a 75% conforme condições abaixo:

- **FI = 50%** para **funcionalidade de sistema desenvolvida ou mantida** por meio de um projeto de melhoria realizada pela CONTRATADA.
- **FI = 75%** para **funcionalidade de sistema não desenvolvida ou mantida** pela CONTRATADA.

6. MANUTENÇÃO CORRETIVA

6.1. 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. Quando o sistema estiver fora da garantia ou não tenha sido desenvolvido pela CONTRATADA, deverá ser estimado e calculado o tamanho do projeto 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.

$$PF_CORRETIVA = FI \times PF_ALTERADO$$

Fator de Impacto (FI):

- **FI = 50%** para **funcionalidade de sistema desenvolvida ou mantida** por meio de um projeto de melhoria realizada pela CONTRATADA.
- **FI = 75%** para **funcionalidade de sistema não desenvolvida ou mantida** pela CONTRATADA.

6.2. As demandas de manutenção corretiva contemplarão a geração da documentação da funcionalidade corrigida. Caso seja erro em requisitos, essa demanda deve ser tratada como projeto de melhoria (alteração de funcionalidade).

7. COMPONENTE INTERNO REUSÁVEL

7.1. 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.

7.2. 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 projetos de melhoria do CPM, seriam contadas todas as funcionalidades impactadas por essa mudança.

7.3. No entanto, este roteiro 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, apresentados na seção - Projeto 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.

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

7.4. Exemplo de manutenção de componentes:

- a) 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”.
- b) Além disso, 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.

7.5. 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:

$$\text{PF_COMPONENTE_ARQUIVO} = 0,6 \text{ PF} \times \text{QTD_ARQUIVOS_ALTERADOS}$$

8. ADAPTAÇÃO EM FUNCIONALIDADES SEM ALTERAÇÃO DE REQUISITOS FUNCIONAIS

8.1. 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:

- 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;
- Permitir exclusões múltiplas em uma funcionalidade que antes só possibilitava a exclusão de um item;
- Adaptação de uma 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;
- 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;
- Modificar o servidor a ser acessado em uma funcionalidade de *download* de arquivo;
- 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 trata-se de mudança em validação com regra de negócio não funcional.

8.2. 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.

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

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

- **FI = 50%** para **funcionalidade de sistema desenvolvida ou mantida** por meio de um projeto de melhoria pela CONTRATADA.
- **FI = 75%** para **funcionalidade de sistema não desenvolvida ou mantida** por meio de um projeto de melhoria pela CONTRATADA.

8.3. Deve-se destacar que além da adequação das funcionalidades em questão, a documentação do projeto de manutenção adaptativa deve ser realizada, quando aplicável, mediante orientação da RESPONSÁVEL TÉCNICA.

9. MANUTENÇÃO DE DOCUMENTAÇÃO DE SISTEMAS LEGADOS

- 9.1. Nesta seção são tratadas demandas de documentação ou atualização de documentação de sistemas legados.
- 9.2. Para este tipo de projeto foi definido o fator de impacto (FI) dos pontos de função da aplicação em questão, considerando a fase do processo de desenvolvimento requisitos e a geração de artefatos associados a requisitos, conforme a fórmula abaixo.
- 9.3. As premissas utilizadas estão definidas no Termo de Referência e seus anexos e documentadas no documento de estimativas de serviço.

$$\text{PF_DOCUMENTAÇÃO} = \text{PF_INCLUÍDO} \times \text{FI}$$

10. ATUALIZAÇÃO DE DADOS

- 10.1. Em alguns casos, as demandas de correção de problemas em base de dados estão associadas a atualizações diretamente no banco de dados em um único registro.
- 10.2. 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.

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

11. MANUTENÇÃO EM INTERFACE

- 11.1. 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:
- 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;
 - Alteração de título de um relatório;
 - Alteração de labels de uma tela de consulta.
- 11.2. Neste tipo de manutenção não são contadas funções de dados.

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

- 11.3. Está contemplada a atualização da documentação das funcionalidades da aplicação impactadas pela manutenção nas demandas desta categoria. Assim, a documentação (documento de requisitos, documento de interface, protótipo, entre outros) das funcionalidades alteradas deve ser atualizada. Caso não exista documentação para as funcionalidades alteradas, não será contemplada a redocumentação das funcionalidades da aplicação impactadas pela manutenção nas demandas desta categoria.
- 11.4. Observação 1 – Help: As demandas de projetos de desenvolvimento de sistemas ou de manutenção de funcionalidades contemplam o desenvolvimento ou atualização do help da funcionalidade em questão, sendo tratada como uma atividade de documentação no processo de software. No caso de demandas específicas de desenvolvimento ou atualização de help estático de funcionalidades, estas podem ser enquadradas nesta seção e poderá ser usado um valor de multiplicação inferior a 0,6 PF conforme análise de impacto das mudanças propostas.

12. APURAÇÃO ESPECIAL

- 12.1. São funcionalidades executadas apenas uma vez para:
- a) Corrigir problemas de dados incorretos na base de dados das aplicações ou atualizar dados em bases de dados de aplicações, detalhados no **item 13 APURAÇÃO ESPECIAL – BASE DE DADOS**;
 - b) Gerar um relatório específico ou arquivo para o usuário por meio de recuperação de informações nas bases da aplicação, detalhados no **item 14 APURAÇÃO ESPECIAL – GERAÇÃO DE RELATÓRIOS**.
- 12.2. O **item 15 APURAÇÃO ESPECIAL – REEXECUÇÃO** considera os casos de reexecução de uma apuração especial.
- 12.3. A **CONTRATADA** deverá fornecer o script à **RESPONSÁVEL TÉCNICA** para permitir posterior reexecução.
- 12.4. Caso a apuração seja de correção de dados devido a erros de funcionalidades de aplicações desenvolvidas pela **CONTRATADA**, devem ser observadas as cláusulas contratuais com relação a garantias e prazos de correção.

13. APURAÇÃO ESPECIAL - BASE DE DADOS

- 13.1. Este tipo de apuração especial é um projeto que inclui a geração de procedimentos para atualização da base de dados.
- 13.2. 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. 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.

- 13.3. 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.
- 13.4. Como produto da APURAÇÃO ESPECIAL deve ser gerado um relatório para homologação pela CONTRATANTE.
- 13.5. Foram identificados três tipos de Apuração Especial - Base de Dados, cujas fórmulas de cálculo são apresentadas a seguir:
- 13.6. APURAÇÃO ESPECIAL – BASE DE DADOS – SEM CONSULTA PRÉVIA E COM ATUALIZAÇÃO DE DADOS**

$$PF_APURAÇÃO_BD = PF_INCLUÍDO$$

13.7. APURAÇÃO ESPECIAL – BASE DE DADOS – COM CONSULTA PRÉVIA SEM ATUALIZAÇÃO DE DADOS

- 13.7.1. 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 CONTRATADA, obrigatoriamente essa deve ser solicitada pelo CONTRATANTE para a avaliação da viabilidade de implementar a Apuração Especial -Base de Dados. 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:

$$PF_CONSULTA_PRÉVIA = PF_INCLUÍDO$$

13.8. APURAÇÃO ESPECIAL – BASE DE DADOS – COM CONSULTA PRÉVIA E COM ATUALIZAÇÃO DE DADOS

- 13.8.1. 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, seguindo a fórmula abaixo.

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

14. APURAÇÃO ESPECIAL – GERAÇÃO DE RELATÓRIOS

- 14.1. 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 essas funções transacionais são Saídas Externas, devido à atualização do Arquivo Lógico Interno.
- 14.2. 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.
- 14.3. 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.

$$PF_APURAÇÃO_RELATÓRIOS = PF_INCLUÍDO$$

15. APURAÇÃO ESPECIAL –REEXECUÇÃO

- 15.1. Em alguns casos, a CONTRATANTE pode ter interesse em executar uma apuração especial mais de uma vez. 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:

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

16. DESENVOLVIMENTO, MANUTENÇÃO E PUBLICAÇÃO DE PÁGINAS ESTÁTICAS DE INTRANET, INTERNET OU PORTAL.

- 16.1. 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.
- 16.2. Caso o desenvolvimento de páginas estáticas esteja contido em um projeto de desenvolvimento, então elas serão contabilizadas no projeto de desenvolvimento e não devem ser mensuradas em separado. Ou seja, esta seção se aplica quando ocorrer a demanda exclusivamente para o desenvolvimento ou manutenção de páginas estáticas.
- 16.3. Cada página desenvolvida ou mantida, de acordo coma fórmula abaixo:

$$\text{PF_PUBLICAÇÃO} = 0,2 \text{ PF} \times \text{QUANTIDADE DE PÁGINAS ALTERADAS OU INCLUÍDAS}$$

17. VERIFICAÇÃO DE ERROS

- 17.1. As verificações de erro ou análise e solução de problemas são as demandas referentes a todo comportamento anormal ou indevido apontado pelo CONTRATANTE nos sistemas aplicativos.
- 17.2. Se for constatado algum erro de sistema, a demanda será atendida como manutenção corretiva. Caso contrário, será realizada a aferição do tamanho em pontos de função das funcionalidades verificadas que o CONTRATANTE reportou erro.

$$\text{PF_VERIFICAÇÃO} = \text{PF_FUNCIONALIDADE_REPORTADA_COM_ERRO} \times 0,15$$

- 17.3. É importante ressaltar que a demanda de verificação de erros deve ser associada a uma funcionalidade específica.
- 17.4. Os casos de sistema fora do ar por conta de problemas de infraestrutura (ex: rede ou banco de dados, servidor de arquivo, servidor de aplicação, barramento etc) 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 roteiro de métricas, não se aplicando verificação de erros nestes casos.

18. PONTOS DE FUNÇÃO DE TESTE

- 18.1. O escopo de uma medição de ponto de função abrange funcionalidades incluídas, alteradas e excluídas e conversão de dados, conforme as regras do IFPUG. A CONTRATANTE pode optar pela contratação especificamente de testes em funcionalidades além deste escopo. Essa opção não se refere a eventuais testes que o desenvolvedor tenha que fazer como parte de uma melhoria.
- 18.2. Vale ressaltar que a CONTRATADA só poderá executar estes testes tendo sido solicitados ou previamente aprovados pela CONTRATANTE ou RESPONSÁVEL TÉCNICA.
- 18.3. A contagem de PFT será o somatório dos tamanhos em pontos de função das funções transacionais envolvidas no teste:

$$\text{PFT} = \text{SOMATÓRIO DOS TAMANHOS DAS FUNÇÕES TRANSACIONAIS TESTADAS}$$

- 18.4. A conversão do PFT em ponto de função deve ser feita de acordo com a fórmula abaixo:

$$PF_TESTES = PFT \times 0,15$$

- 18.5. 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, consideradas no PFT, devem ser documentadas pela CONTRATADA considerando-se a documentação de testes definida no processo de desenvolvimento da RESPONSÁVEL TÉCNICA. Observe que estas funções farão parte do escopo do projeto de manutenção.

19. PROJETO DE MIGRAÇÃO DE DADOS

- 19.1. O projeto de migração de dados deve ser contado como um novo projeto de desenvolvimento de um sistema, seguindo a fórmula abaixo:

$$PF_CONVERSÃO = PF_INCLUIDO$$

- 19.2. Não deverão integrar o escopo da medição os componentes de dados (ALIs e/ou AIEs) que já tiverem sido contados no projeto de desenvolvimento ou melhoria, evitando, desta maneira uma recontagem indevida e um super dimensionamento deste serviço.

20. ORIENTAÇÕES COMPLEMENTARES PARA CONTAGEM

- 20.1. Este tópico 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 aquisições que podem impactar significativamente o resultado de uma contagem em caso de falhas.
- 20.2. As diretrizes para contagens de pontos de função envolvendo Múltiplas Mídias têm como base o artigo “Considerations for Counting with Multiple Media” Release 1.1 publicado pelo IFPUG [IFPUG, 2010a].

I. REQUISITO FUNCIONAL

- 20.3. **Definição:** Os requisitos funcionais do usuário são um subconjunto dos requisitos do usuário; descrevem o que o software deve fazer, em termos de tarefas e serviços.
- 20.4. **Diretriz:** São aqueles requisitos particulares e específicos de uma tarefa ou serviço; contrasta com aqueles requisitos já atendidos de maneira geral por infraestrutura que fornece serviços genéricos, independentes de uma tarefa em particular.
- 20.5. **Cenário A – Requisito não Funcional:** Todos os relatórios do sistema devem por força de uma política da RESPONSÁVEL TÉCNICA ser passíveis de geração nos formatos: portable document format (pdf), Microsoft Word (doc), Microsoft Excel (xls) e texto (.txt).

20.5.1. A CONTRATANTE está orientada pela RESPONSÁVEL TÉCNICA a não solicitar essa característica como um requisito ao expor às suas necessidades em função dela já ser uma limitação do projeto. Trata-se de um requisito não funcional .

20.5.2. Exceções a essa regra devem ser tecnicamente discutidas e aprovadas pela RESPONSÁVEL TÉCNICA.

20.6. **Cenário B – Requisito Funcional:** Um determinado relatório em particular deve poder ser gerado em papel e em um leiaute específico para envio para um órgão regulador. Ambos os produtos são idênticos em termos de conteúdo e boa parte de sua lógica de processamento. Diferem na apresentação dos dados que é feita de uma forma específica em cada caso. Trata-se de um requisito funcional.

20.7. **Cenário C – Requisito não Funcional:** A regra geral é que um relatório ou uma consulta tenham o total de registros nos dados apresentados. A lógica de processamento de cálculo pertence a uma infraestrutura comum de suporte e não a descrição do requisito funcional. Dessa forma não se deve considerar, por este motivo, estas consultas e relatórios como Saídas Externas.

II. LOG, TRILHA DE AUDITORIA E HISTÓRICO

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

20.9. LOG

20.9.1. 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.

20.9.2. 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 abaixo performance no sistema, travamentos e outros comportamentos inesperados.

20.9.3. Diante desse conceito, o Log não deve ser mensurado com Pontos de Função, já que ele não armazena informações negociais reconhecidas pelo usuário da aplicação.

20.10. TRILHA DE AUDITORIA

20.10.1. 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.

- 20.10.2. Trilha de Auditoria: apoia a auditoria para os dados de negócio, armazenando informações das ações realizadas pelo usuário na aplicação.
- 20.10.3. 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.
- 20.10.4. A trilha de auditoria deve ser solicitada pelo usuário da aplicação e, para a contagem, deve existir funcionalidade de consulta a tais dados.
- 20.10.5. Caso a trilha de auditoria faça parte do framework adotado pelo CONTRATANTE/REPOSÁVEL TÉCNICA, ela deve ser considerada como um requisito não funcional e, portanto, não será mensurável em ponto de função.

20.11. HISTÓRICO

- 20.11.1. 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.
- 20.11.2. A função de consulta aos dados de um histórico deverá ser contada de acordo com as regras de contagem das funções transacionais do CPM.
- 20.11.3. 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.

III. IDENTIFICAÇÃO DE PROCESSO ELEMENTAR

- 20.12. Um Processo Elementar é a menor unidade de atividade que é significativa para o(s) usuário(s). O Processo Elementar deve ser auto contido e deixar o negócio da aplicação que está sendo contada em um estado consistente.
- 20.13. 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.
- 20.14. 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.

20.15. PEQUENAS DIFERENÇAS EM PROCESSOS

20.15.1. **Definição:** Um processo elementar pode incluir pequenas variações nos TD ou AR assim como múltiplas alternativas, variações ou ocorrências de lógicas de processamento.

20.15.2. **Diretriz:** Esta diretriz é feita por meio de exemplos que ilustram o que é considerada uma pequena diferença.

20.15.3. Exemplos:

- Uma interface com o usuário apresenta uma lista de elementos, ao escolher nessa lista um elemento, são apresentados os mesmos campos de um único elemento em outra interface.
- Uma interface com usuário permite consultar dados a partir de um painel onde o usuário especifica os mais diversos critérios de seleção. Uma vez definidos esses critérios, uma lista com os registros que atendem aos mesmos é apresentada com os mesmos dados.
- A exclusão de um registro pode ser realizada a partir de: uma lista onde o usuário escolhe uma série de itens a excluir; uma interface com o usuário em que os dados de um item são apresentados em um painel.
- Haver uma interface com o usuário para incluir uma localidade e haver a possibilidade de incluir uma localidade no contexto de outra função: Existe uma interface com o usuário especificamente para incluir uma localidade; em outra interface que tem por objetivo incluir uma ocorrência, também é possível incluir uma localidade visando tornar mais ágil a utilização do sistema por seus usuários.
- Ao cadastrar uma pessoa ela pode ser classificada como pessoa física ou pessoa jurídica. Ambos compartilham a maioria dos atributos e regras. A pessoa física tem alguns outros atributos específicos de seu tipo, enquanto a pessoa jurídica, outros atributos específicos. Há um mesmo ator que em um mesmo passo do fluxo de trabalho realiza essa função; um único caso de uso descreve adequadamente o processo em questão independentemente de se tratar de uma pessoa física ou jurídica.

IV. “COMPLETO”

20.16. **Definição:**

- a) Estado consistente: Ponto no qual o processamento foi totalmente executado; o Requisito Funcional do Usuário foi satisfeito e não há nada mais a fazer.
- b) Auto-contido: Não há passos de processamento anteriores ou posteriores necessários para iniciar ou completar o requisito funcional do usuário.

20.17. **Diretriz:** considerar a ótica do negócio e não a ótica do sistema para julgar um estado como consistente e um processamento como auto-contido.

20.18. Cenário A

20.18.1. É muito comum em ambiente Web haver gravações de quadros em um formulário que o compreende e representando um único documento alvo como uma transação completa em nível de sistema.

20.18.2. Trata-se do mesmo ator realizando uma mesma função em um único passo do fluxo de trabalho. Por motivações não funcionais (instabilidade do ambiente) o trabalho realizado no quadro é gravado antes de terminar todo esse passo para atender limitações de ordem geral quanto à qualidade e à organização.

20.18.3. Os conceitos de completo e autocontido aplicam-se ao preenchimento do documento alvo em termos dos dados definidos naquele passo do fluxo de trabalho por um mesmo ator, incluindo eventuais conjunto de dados que tenham várias possíveis ocorrências informadas e apresentadas em quadros.

20.19. Cenário B

20.19.1. Antes de prosseguir para um próximo passo em um fluxo de trabalho automatizado, o documento pode ser salvo como um rascunho ou então como uma versão final. Haverá uma série de consequências em termos dos próximos passos: Se salvo como um rascunho, o usuário pode realizar alterações ou mesmo excluir o documento informado no sistema; Se salvo como uma versão final, o documento não pode ser mais alterado por esse usuário e se torna disponível para processamento em outros passos do fluxo de trabalho. Mesmo que salvar como um rascunho ou como uma versão final sejam duas ações em um formulário, trata-se de um campo sendo informado pelo usuário durante um processo de inclusão e alteração. O que é completo é informar os dados do formulário, informar a sua condição (rascunho ou versão final) é salvar os dados.

20.19.2. Se atores diferentes ou eventualmente o mesmo ator, desde que em diferentes passos do fluxo de trabalho, realizam atualizações em um mesmo documento referentes às informações que são criadas e devem ser registradas nos diferentes passos, vários processos diferentes são considerados completos e não simplesmente as funções de inclusão e alteração do documento são contadas.

20.20. Cenário C

20.20.1. Antes prosseguir para um próximo passo em um fluxo de trabalho automatizado, o documento deve ser caracterizado como uma versão final. Sempre ao salvar o documento nas funções de inclusão ou alteração, ele é guardado na condição de rascunho. Existe um passo no fluxo de trabalho em que nenhum dado é alterado e o usuário indica, após a revisão do mesmo, que ele agora está na condição de versão final. Neste cenário, promover documento à condição de versão

final é completo e independente da alteração ou inclusão. Observe o contraste deste exemplo com o exemplo do **Cenário B** anteriormente apresentado.

- 20.20.2. O critério para identificar um processo completo é de que o requisito do usuário possa ser atendido com apenas uma interface com o usuário ao invés de utilizar diversas, ainda que dessa forma funcione mal (pouca usabilidade e desempenho sofrível, por exemplo) e ofereça maior complexidade técnica.

20.21. Cenário D

- 20.21.1. Todos os atores habilitados a incluir ou alterar uma solicitação de serviço, podem também registrar um complemento à mesma ou incluir um novo participante. Existem três interfaces com o usuário para esse fim, contudo, não há nada em termos das práticas e procedimentos do usuário que impeça disso ser entregue como uma única interface com o usuário.

21. CONTAGEM DE PONTOS DE FUNÇÃO DE DESENVOLVIMENTO DE SOFTWARE USANDO MÉTODOS ÁGEIS

- 21.1. O ciclo de pagamento será a “Release”.
- 21.2. As mudanças em funcionalidades podem ser decorrentes de mudanças no domínio do negócio - como alteração de escopo, de regras de negócio - ou mudanças legais/regulamentares ou, ainda, refinamentos de requisitos.
- 21.3. Dentro da Release não haverá remuneração extra pelo retrabalho em nenhum destes casos.
- 21.4. Fora da release, onde a mesma já foi homologada, caso haja mudança de requisitos motivada por fatores externos ao processo de desenvolvimento (mudanças em legislação, diretrizes de negócio, etc.), será aberto um projeto de melhoria e remunerado como tal. Não haverá remuneração nos casos onde a mudança no software desenvolvido foi motivada pelo levantamento incompleto de requisitos.