

Síntese das discussões do fórum Livro-APF: Janeiro/2011

Assunto: Contagem de Projetos de Melhoria

Data: 04/01/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3405>

Cenário: Como deve ser feita a contagem para a seguinte melhoria?

Antes da Alteração:

ALI Funcionário

1. Nome

2. Endereço

...

50. CPF

1 TR / 50 TD / Baixa

Depois da Alteração:

ALI Funcionário

1. Nome

2. Endereço

...

50. CPF

51. Telefone

1 TR / 51 TD / Média

Antes da Alteração:

Entrada Externa - Incluir Funcionário

1 AR / 50 TD / Média

Depois da Alteração:

Entrada Externa - Incluir Funcionário

1 AR / 51 TD / Média

A tela de inclusão recebeu mais um campo que deverá ser informado pelo usuário.

"Se a mudança envolve apenas a inclusão de novos registros no arquivo lógico ou novos valores em um campo existente dentro do arquivo lógico, não existe justificativa para contar a função de dados como sendo alterada."

"Se novos campos são incluídos em um ALI, procure por novas modificações nas funções de transação que mantém o campo neste ALI para confirmar que a mudança ocorreu"
 PF de projeto de melhoria (EFP) = [(ADD + CHGA + CFP) x VAFA] + (DEL x VAFB)
 Fonte: CPM 4.2

Não ficou claro o que é alteração para a contagem de melhoria. Foi identificado que uma Entrada Externa foi alterada. Devemos contar a Entrada Externa novamente e utilizá-la na fórmula $EFP = [(ADD + CHGA + CFP) \times VAFA] + (DEL \times VAFB)$? Mesmo que a complexidade continue a mesma? Se o usuário quer saber quantos PFs a alteração gerará, devemos contar a Entrada Externa inteira novamente?

Análise/Conclusão: Segundo esse exemplo, temos o tamanho da melhoria que compreende as funcionalidades que foram Incluídas, Alteradas e Excluídas (tamanho da melhoria), conforme manual CPM 4.3.1 Parte 3 - Práticas de Contagem Projetos de Melhoria e Atividades de Manutenção:

1 ALI médio = 10 PF

1 EE médio - Incluir Funcionário= 4PF

Aplica-se então a fórmula

$EFP = ADD + CHGA + CFP + DEL$

Onde:

- EFP é a contagem de pontos de função do projeto de melhoria
- ADD é o tamanho das funções que estão sendo adicionadas pelo projeto de melhoria
- CHGA é o tamanho das funções sendo alteradas pelo projeto de melhoria - como elas são / serão após a

www.fattocs.com.br

implementação

- CFP é o tamanho das funcionalidades de conversão
- DEL é o tamanho das funções sendo excluídas pelo projeto de melhoria

$$EFP = 0 + (10+4)+0+0 = 14$$

Assim, temos 14 PFs resultantes do tamanho da melhoria solicitada.

Segundo o manual da NESMA deve-se aplicar redutores para cada tipo de melhoria solicitada, a saber:

Funcionalidades incluídas = 100% (ou seja, sem redução)

Funcionalidades excluídas = 40%

Para as funcionalidades alteradas, o cálculo é bem mais complexo e para isso seria necessário acompanhar o seu desenvolvimento do manual da NESMA.

Contratualmente este redutores podem também ser definidos. Algumas empresas os utilizam da seguinte maneira:

Funcionalidades incluídas = 100% (ou seja, sem redução)

Funcionalidades excluídas = 40%

Funcionalidades alteradas = 70 % (independente do tipo da função ou do tipo de modificação feita).

A fórmula indicada acima representa o tamanho total aplicação após a melhoria e não o tamanho da melhoria em si, que não foi muito afetada em seu tamanho a não ser pela alteração proposta no ALI (de baixo = 7PF passou a médio = 10PF). Resultando em uma diferença de 3 PF.

Pode ser necessário somente o tamanho funcional que representa o pedaço onde está sendo aplicada a melhoria, para então aplicarmos métodos de conversão e estimarmos o tempo e o custo, por exemplo, que incidam sobre a realização da mesma. Tais informações são de grande valia quando se precisa negociar com os clientes.

Assunto: Contagem de Pontos de Função

Data: 05/01/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3409>

Cenário: No desenvolvimento de um sistema, foi definido pelo arquiteto que todas as regras de negócio terão funcionalidades armazenadas no banco de dados.

Por exemplo, a validação do CPF será um registro em banco de dados que será chamado quando necessário.

E ele está armazenado na tabela TB_Regra_Negocio

```
id_regra_negocio = 1234
```

```
no_regra_negocio = 'Validação CPF'
```

```
txt_regra = '//VERIFICA SE O QUE FOI INFORMADO É NÚMERO
```

```
if(!is_numeric($cpf)) {
```

```
  $status = false;
```

```
}
```

```
else {
```

```
  //VERIFICA
```

```
  if( ($cpf == '1111111111') || ($cpf == '2222222222') ||
```

```
  ($cpf == '3333333333') || ($cpf == '4444444444') ||
```

```
  ($cpf == '5555555555') || ($cpf == '6666666666') ||
```

```
  ($cpf == '7777777777') || ($cpf == '8888888888') ||
```

```
  ($cpf == '9999999999') || ($cpf == '0000000000') ) {
```

```
  $status = false;
```

```
}
```

```
else {
```

```
  //PEGA O DIGITO VERIFICADOR
```

```
$dv_informado = substr($cpf, 9,2);
```

```
for($i=0; $i<=8; $i++) {
$digito[$i] = substr($cpf, $i,1);
}
```

```
//CALCULA O VALOR DO 10º DIGITO DE VERIFICAÇÃO
```

```
$posicao = 10;
```

```
$soma = 0;
```

```
for($i=0; $i<=8; $i++) {
$soma = $soma + $digito[$i] * $posicao;
$posicao = $posicao - 1;
}
```

Agora estamos em fase de manutenção. Para melhoria do *software* é necessário a criação de novas regras de negócio, ou seja, desenvolvimento de uma regra negocial. Como podemos fazer a contagem de Pontos de Função?

Análise/Conclusão: O artifício do arquiteto favorece à manutenibilidade do *software* no sentido que facilitará mudanças na aplicação, entre outras.

Entretanto, o tamanho funcional da aplicação não está relacionado a artifícios de programação ou similares, como neste caso, mas relaciona-se à percepção do usuário.

O que devemos fazer é rastrear as funcionalidades impactadas pela criação ou alteração destas regras, pois mudanças em processos de negócio muito provavelmente implicarão em mudanças na lógica de processamento de uma ou mais transações.

Decorrida esta análise, a Análise de Pontos de Função pode ser aplicada normalmente.

Assunto: Contagem de Pontos de Função

Data: 06/01/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3411>

Dúvida: Cadastro de perguntas: Incluir, Alterar, Excluir e Consultar.

Na alteração e na exclusão existe uma consulta implícita para mostrar os dados cadastrados. A funcionalidade de consulta, também mostra parte destes dados, porém no formato de lista. Devemos contar duas CEs distintas?

Análise/Conclusão: Serão duas CE distintas se (e somente se):

- O conjunto dos itens de dados for diferente; OU
- O conjunto dos arquivos referenciados for diferente; OU
- A lógica de processamento do processo elementar for diferente.

Dúvida: Mostrar os mesmos campos dos mesmos arquivos referenciados em formato de lista ou em outro formato caracteriza mudança na lógica de processamento?

Análise/Conclusão: Haver a tela com todos os detalhes e outra tela com informações resumidas deve ser contado como dois processos elementares quando ambas isoladamente atenderem uma necessidade específica do usuário independentemente uma da outra. Haverá ocasiões onde apenas uma lista com informações resumidas atende a uma necessidade específica e outras ocasiões onde todos os detalhes são necessários para atender outra necessidade específica.

Devemos ter cuidado nessa avaliação para não nos depararmos com dependência introduzida pela interface gráfica que obriga ao usuário antes de consultar os detalhes passar por uma lista ao invés de informar diretamente qual registro deseja consultar.

A lógica citada acima, hoje na versão 4.3.1, não necessariamente se aplica. Isso na medida em que o padrão de medição admite pequenas variações em qualquer dos três itens citados. Essa regra foi alterada para evitar a prática de alguns profissionais que contavam um processo relativo a mostrar uma lista sem

parâmetros e um outro processo para contar a apresentação da mesma lista com parâmetros.

Dúvida: Ainda há uma dúvida. Mais especificamente no seguinte trecho: "(...) A lógica citada, hoje na versão 4.3.1, não necessariamente se aplica. Isso na medida em que o padrão de medição admite pequenas variações em qualquer dos três itens citados..."

Todas as informações obtidas até este momento eram de que o CPM 4.3.1 apenas explicava melhor pontos que na versão anterior dava margem a dúvidas.

Não encontrei no 4.3.1 nenhum trecho que aludisse a alteração na regra de avaliação de PE, citada acima:

- O conjunto dos itens de dados for diferente; OU
- O conjunto dos arquivos referenciados for diferente; OU
- A lógica de processamento do processo elementar for diferente."

Seria possível citar a parte do CPM que dispõe sobre qualquer divergência ou variância da regra aludida?

Análise/Conclusão: Vamos às evidências: CPM 4.3.1, Parte 2, Página 7-11.

Assunto: ALI

Data: 10/01/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3421>

Cenário: Considere um cenário em que há 2 ALIs:

ALI 1 – Veículo

ALI 2 - Combustível

Premissa 1 - Combustível não é um *code data*.

Para todo veículo temos que informar seu combustível, então temos na tabela física de Veículo uma chave estrangeira.

Já que o código do combustível está sendo contabilizado no ALI - Combustível, não contabilizamos ele como um DET no ALI - Veículo.

Partindo dessa premissa, temos a seguinte situação: Quando fazemos uma transação para incluir um registro no ALI veículo, o código do combustível atravessa a fronteira da aplicação e também deveria ser contabilizado. Mas se ele não consta no ALI de Veículo, e a operação de inclusão é apenas do ALI de Veículo, como que contabilizamos isso?

Análise/Conclusão: Ao que parece, Combustível é um grupo de dados lógico, mantido por alguma aplicação. Neste caso, Veículo e Combustível parecem entidades independentes, pois Veículo é mantido sem Combustível, e Combustível é mantido sem Veículo.

Sendo entidades independentes, Veículo conta 1 ALI e 1 RLR, e Combustível conta 1 ALI (ou AIE) e 1RLR.

A transação que acessa Veículo + Combustível conta 2 Arquivos Referenciados. Para tipos de dados dos ALIs, Veículo conta 1 TD para Combustível, e Combustível conta 1 TD para Veículo. Se as funções de transação utilizarem o mesmo tipo de dado, exemplo, *Codigo_Combustível*, não precisa contar duas vezes, só uma vez para a função de transação.

Dúvida: Há uma coisa que não concordo: *Para tipos de dados dos ALIs, Veículo conta 1 tipo de dado para Combustível, e Combustível conta 1 tipo de dado para Veículo.*

Combustível realmente é chave estrangeira em Veículo (e conta 1 TD - Código do Combustível). Mas o inverso não é verdade, ou seja, Veículo não é chave estrangeira em Combustível. Não há qualquer menção a essa relação.

Análise/Conclusão: Considerando um relacionamento (n):(m).

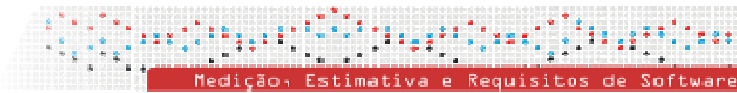
Deve ter uma tabela associativa de Combustível x Veículo, e nesse caso, as chaves estrangeiras da associativa vão para as entidades, e são contadas como tipos de dados de cada entidade.

Complemento: Não se pode deduzir esse relacionamento (n) : (m) a partir da questão. A única menção na questão sobre relacionamentos entre os dois arquivos lógicos é: *Para todo veículo temos que informar seu combustível, então temos na tabela física de Veículo uma chave estrangeira.*

Esse relacionamento é do tipo 1 : (N)

Comb. Veíc.

E as entidades são independentes entre si.



Assunto: Dúvida de Contagem – Entidade de Dados

Data: 13/01/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3425>

Dúvida: No livro Análise de Pontos de Função – Medição, Estimativas e Gerenciamento de Projetos de Software, edição 9, página 88, o autor mostra um exemplo em que uma entidade Funcionário tem subtipos: "Horista" e "Mensalista", e que Funcionário é obrigatório, mas Horista e Mensalista são subgrupos opcionais. Então, esta contagem fica 1 arquivo lógico e 3 RLR: Funcionário, Funcionário Horista e Funcionário Mensalista.

No exemplo do Manual 4.2 do IFPUG, Parte 2 - página 2-32, tem um exemplo parecido. Uma entidade funcionário tem dois subtipos: "Permanente" ou "Contratado". Um funcionário tem que ser um permanente ou contratado, mas não pode ser os dois. Os dados comuns do funcionário são pertinentes a todos os funcionários e são obrigatórios. Além disso, os dados comuns são herdados pelas entidades subtipo obrigatórias, permanente e contratado. Neste caso, foi contado 1 arquivo lógico e apenas 2RLR.

Por que no primeiro caso foi contado 3RLR e no segundo caso apenas 2RLR?

Análise/Conclusão: No caso do exemplo do livro temos três grupos de dados: Dados de Funcionário (Dados pertinentes ao funcionário independentemente dele ser mensalista ou horista e que pode ser criado nessa condição - ao contrário do exemplo apresentado no CPM e daí a diferença); Dados opcionais específicos do funcionário mensalista; e Dados opcionais específicos de funcionário horista.

Assunto: Contagem Final do Projeto – Aplicação x Detalhada

Data: 24/01/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3451>

Dúvidas:

- A contagem de Aplicação pode ser usada para fins de ajuste de remuneração com a contratada? Ela é importante para sabermos o tamanho do projeto final, mas não representa de fato o esforço que se gastou para ser desenvolvido (devido às mudanças de escopo). Portanto, não pode ser usada para final do projeto para fins de ajustes financeiros com a contratada.
- Neste caso é indicado que tipo de contagem? A Detalhada? Em termos de quantidade de PF é a mesma coisa usar Detalhada ou Aplicação, não?
- Então, como “acertar as contas”? Pois não podemos contar o tamanho final e simplesmente subtrair o que já se pagou, pois se a mudança de escopo foi grande, corre o risco da fábrica ficar devendo dinheiro. O que os contratantes ou contratadas de fábrica fazem?

Análise/Conclusão: A contagem da Aplicação reflete as funcionalidades atualmente disponibilizadas aos seus usuários. A contagem de Desenvolvimento é o espelho do seu escopo inicial.

As diversas contagens de Melhoria são justamente esses "Ajustes Financeiros" com a contratada. Não se pode dizer que a empresa contratada ganhou mais do que devia pelo simples fato de após um projeto de melhoria, a contagem da Aplicação ter diminuído.

As melhorias são mudanças de escopo e devem estar documentadas em Relatórios de Controle de Mudanças (RCM). Cada alteração feita no projeto é contada separadamente, e conseqüentemente é paga separadamente.

Um Exemplo em números:

- 1- Projeto de Desenvolvimento: 250 Pontos de Função
- 2- Aplicação: 250 Pontos de Função (consideremos que não houve conversões de dados)
- 3- Projeto de Melhoria para a mesma aplicação, após implantada: 40 Pontos de Função.
- 4- Aplicação após a melhoria: 240 Pontos de Função (as melhorias incluíram exclusões de funcionalidades)

Neste exemplo em números, não poderíamos "punir" a empresa contratada pelo fato de ter diminuído 10 Pontos de Função na Aplicação. O que deve ser feito, é pagar os 40 Pontos de Função que foram necessários para o Projeto de Melhoria. Com isso, ao todo foram pagos 290 Pontos de Função, mesmo a contagem de Aplicação Final do projeto apontar apenas 250.

Dúvida: No início do projeto fazemos uma contagem estimada. Em algum momento deveremos fazer a detalhada para fins de ajustes em tamanho, custo e prazo. Devemos fazer isso ao final do Projeto com toda documentação e código entregue? Lembre-se que até agora estávamos trabalhando com a estimada porque não era suficiente contar a detalhada.

Neste caso deveremos contar a detalhada (pois pode ter conversão de dados) para ter um novo valor financeiro e ajustar as contas com o fornecedor?

Análise/Conclusão: Nesse caso, realmente existe ajuste de contas a se fazer com a contratada. A contagem estimativa que é realizada no início do projeto, deve ser comparada com a contagem detalhada ao final da codificação.

Muito provavelmente existirão diferenças (para mais ou para menos) e com certeza elas devem ser ajustadas junto à empresa contratada.

Assunto: Mensuração de Projeto

Data: 25/01/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3454>

Dúvida: Sabemos que a mensuração de um projeto é bem útil se tratando do desenvolvimento. Entretanto, para o detalhamento do cronograma, como devemos mensurar os tempos de especificação (análise), gerência de projetos, infra (configuração ambiente), homologação, entre outros?

Análise/Conclusão: Deve-se converter esse PF em horas de esforço.

Por exemplo: se o projeto tem 100PF e utilizamos a produtividade da linguagem de 10h/PF, o projeto terá 1000 horas.

Dessas mil horas, devemos distribuir entre % 's estimados em fases/disciplinas dependendo da empresa ou metodologia utilizada. Depois, devemos alocar os profissionais para cada fase e disciplina.

Dúvida: Mas neste caso, não estaríamos suprimindo o tempo de desenvolvimento de fato do projeto? No exemplo, se temos 100 PF de projeto, isso seria a pontuação do desenvolvimento da aplicação. Não há como incluir aí, pela conversão de PF por horas, os tempos das demais fases do ciclo de vida do sistema – análise, especificação, testes, etc...

É bem verdade que pode ser uma metodologia particular de empresa, mas a impressão é que está suprimindo as demais atividades do projeto como todo. O tempo encontrado no exemplo – mil horas – seria apenas de desenvolvimento do projeto – fase de implementação/codificação do sistema.

Análise/Conclusão: Isto dependerá de quais esforços estão sendo considerados na produtividade. Se a produtividade de 10h por ponto de função contemplou (nessas 10h) todas as disciplinas de desenvolvimento então estamos estimando tudo.

Na prática, esta produtividade usada pelas empresas já contempla todas as fases do ciclo de vida de desenvolvimento de software (seja esta cascata, processo unificado ou ágil).

É que se ao iniciarmos uma demanda de 1PF, cronometrássemos o tempo para todas as atividades executadas (gestão, requisitos, análise, especificação, construção, teste implantação, etc) iremos constatar que ao final de tudo, se gastou 10 horas para fazer este ponto de função.

Assunto: Contagem para Interface de Telefonia

Dúvida: Há um Sistema que está integrado a uma plataforma de telefonia através de um objeto OCX. Este objeto OCX provê métodos/eventos que permitem a codificação da lógica de negócio em 2 telas do Sistema.

Contudo, esta plataforma de telefonia será substituída e, conseqüentemente, um novo objeto OCX precisará ser utilizado, sendo necessário um desenvolvimento das lógicas de negócio em função da adequação à nova plataforma. É necessário dimensionar o desenvolvimento para esta nova plataforma. Como devemos utilizar a contagem de PF para estimar o esforço necessário das adequações a esta nova plataforma?

Análise/Conclusão: É necessário conhecer o escopo para estimar esforço. Neste caso, já há um ponto de partida, que é a plataforma atual. Imaginando que as regras de negócio serão mantidas, com a medição da plataforma atual, podemos chegar a um número real (ou, ao menos, bem próximo dele) do tamanho funcional da nova plataforma.

Quanto ao escopo da contagem, temos dois pontos específicos: Desenvolvimento, no que diz respeito ao desenvolvimento de um novo objeto; e a manutenção do sistema legado, caso novas funcionalidades sejam criadas ou precisem ser mantidas. Essa manutenção parece ser necessária, uma vez que o legado precisará usar o novo objeto.

Por fim, para estimar o esforço, é necessário também conhecer a produtividade da equipe na(s) nova(s) linguagem(ens) que serão utilizadas.