

Síntese das discussões do fórum Livro-APF: Outubro/2010

Assunto: Manutenção em Requisitos Não Funcionais

Data: 06/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3090>

Dúvida: Há uma dúvida quanto à contagem de manutenções em requisitos não funcionais. Por exemplo:

- 1) Se numa mudança de plataforma (versão atualizada de algum browser) houver a necessidade de atualizar uma DLL global, que está ligada a toda aplicação. Considerando que essa será a única atualização a ser feita, o que considero na contagem dos Pontos de Função? Toda a aplicação?
- 2) Numa migração de Bancos de Dados de um ambiente para outro, algumas alterações foram efetuadas nos ".ini" dos sistemas para redirecioná-los para os novos endereços dos Bancos de Dados. O que se deve contar como PF, uma vez que essa mudança foi apenas na parte física da aplicação? Deve-se considerar que todo o sistema sofreu impacto com essa mudança e contar todos os PF's de todos os sistemas que acessam a esses BD que migraram?

Análise/Conclusão: Para a primeira questão, não se trata da contagem de um projeto de melhoria e não deve ser medido como tal nem tão pouco utilizar indicadores de produtividade (HH/PF ou R\$/PF) típicos dos projetos de melhoria. Meça a aplicação usando a contagem da aplicação e utilize indicadores de produtividade de teste (HH/PF ou R\$/PF) relativos a atividades dos tipos de testes envolvidos.

Para a segunda questão, deve-se seguir o mesmo raciocínio da opção anterior, se for o caso de testar a aplicação inteira.

Assunto: Produtividade (HH/PF)

Data: 07/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3092>

Dúvida: Quais as faixas de produtividade típicas de mercado (horas por ponto de função) para desenvolvimento ASP.Net / C# ou Java?

Análise/Conclusão:

1) Há uma equipe cuja meta para ASP .Net é 12HH/PF, mas há projetos cuja produtividade é baixa (18HH/PF) e outros cuja produtividade é mais alta (8HH/PF). O processo de software segue práticas do CMMi N3 e o volume de dados processados pela aplicação e a quantidade de transações simultâneas são grandes, o que exige um ambiente mais robusto para a fase de Testes. Para C#, temos 18, 15 e 10HH/PF para baixa, média e alta produtividade respectivamente. Para Java, o buraco é mais embaixo. Esperávamos projetos dentro da seguinte faixa: 22, 18 e 14, porém há projetos cuja produtividade é tão ruim que sequer consideramos na média para não desvirtuar a tabela. Um dos motivos para produtividade tão baixa é a constante alteração nos requisitos por parte do cliente depois de termos construído a funcionalidade. Não poderia contar a mesma função transacional 2 vezes no mesmo projeto, mesmo que ela tenha sido construída e testada e depois tenha sido completamente alterada.

Resumidamente:

Linguagem	Produtividade Baixa	Produtividade Média	Produtividade Alta
ASP .Net	18 (HH/PF)	12 (HH/PF)	8 (HH/PF)
C#	18 (HH/PF)	15 (HH/PF)	10 (HH/PF)

Java	22 (HH/PF)	18 (HH/PF)	14 (HH/PF)
------	------------	------------	------------

Complemento: Segue um procedimento para lidar com o do problema a respeito da "constante alteração nos requisitos por parte do cliente depois de termos construído a funcionalidade":

a) Quantificar e estabelecer em comum acordo com o cliente a fração (ou percentual) do custo total que corresponde a cada etapa do projeto (exemplo hipotético: requisitos = 10%, análise = 15%, projeto detalhado = 20%, construção = 30%, teste = 15%, homologação = 10%).

b) Fazer uma contagem de pontos de função estimativa logo no início do projeto, mapeando os requisitos nas funcionalidades (ALI/AIE/EE/CE/SE).

Obs: Como nessa etapa geralmente ainda não há informações detalhadas, pode-se usar o padrão de contagem estimativa da NESMA (complexidade baixa para ALI/AIE e média para EE/CE/SE). Assim, basta identificar as funções, sem se preocupar em contar RLR/DER/ALR.

c) A cada vez que se fizer uma entrega de qualquer artefato, indique explicitamente o conjunto de funcionalidades (ALI/AIE/EE/CE/SE) que estão sendo entregues, o valor total em PF dessas funcionalidades e a etapa que está sendo concluída para elas (análise, projeto detalhado, codificação, etc.).

Obs: nessa ocasião geralmente já dá pra contar os RLR/DER/ALR e indicar o valor real em PF de cada funcionalidade.

Com isso, podemos comprovar para o cliente que a mudança de um determinado requisito depois que ele já está construído acarretará um esforço correspondente ao tamanho das funcionalidades a serem modificadas vezes o percentual das etapas que precisarão ser retrabalhadas (exemplo: 50% para projeto detalhado + construção).

Claro que algumas modificações podem ser pequenas e não ser justo cobrar o retrabalho integral. Uma sugestão seria utilizar o "fator de impacto" conforme o padrão NESMA de projetos de melhoria, visando quantificar com mais precisão o esforço de retrabalho devido à mudança de requisito.

Nesse caso ficaria "fator de impacto" x "percentual correspondente às etapas retrabalhadas" x "tamanho em PF das funcionalidades retrabalhadas".

Claro que qualquer cliente resistirá a um compromisso unilateral de ressarcir custos extras de retrabalho sem que haja alguma contrapartida que lhe seja vantajosa. A vantagem para o cliente é que, trabalhando assim de forma transparente, podemos cobrar um valor menor por PF, baseado no seu custo "enxuto" de HH/PF, calculado sem a habitual "gordura para absorver retrabalhos". Daí poderá ficar interessante também para o cliente, pois, se ele for mais cuidadoso e evitar mudanças frequentes de requisitos, poderá concluir o projeto com um custo menor.

2) Houve uma pesquisa com várias empresas (32 para ser mais preciso) a faixa é bem elástica, vai de 6 HH/PF até 14 HH/PF não havendo diferença significativa entre JEE e .Net. Estes valores valem para um determinado ambiente de desenvolvimento, que vai desde o ambiente físico, passando por hardware, frameworks disponíveis, uso de componentes, necessidade de interface com bases corporativas, até a senioridade da equipe, a absorção do processo usado, a complexidade dos sistemas construídos, etc.

3) Produtividade não é um fator simples de ser medido, depende do processo de construção e experiência da equipe. Por isso, não se pode planejar um projeto com equipes diferentes até na mesma empresa usando o mesmo fator de produtividade. Em todo caso, só para título de pesquisa, uma equipe trabalhou com uma média de 6H/PF para Java EE, sendo 48% análise e projeto e 52% desenvolvimento e teste.

Complemento: Com certeza depende de muitos fatores (equipe, ambiente, plataforma, arquitetura, ferramentas, metodologia, complexidade da documentação exigida, familiaridade com os processos do cliente, etc.).

A produtividade também é muito influenciada pelo tipo de sistema que está sendo construído.

Existe sistemas que consistem essencialmente de um enorme conjunto de cadastros do tipo "CRUD", quase todos muito parecidos entre si e acrescidos de muitos "botõesinhos" que fazem consultas de dados triviais. Nesse tipo de sistema ocorre muito reuso, bastando trocar as legendas da tela e os nomes das

tabelas e dos campos. Esse reuso vale não apenas para a construção do código, mas também para os artefatos de especificação e de testes. Isso sem falar que cada "consulta trivial" rende 3 PF com pouquíssimo esforço.

Já outros sistemas têm baixa pontuação em PF por terem poucas funcionalidades acionáveis pelo usuário, mas requerem grande esforço porque essas poucas funcionalidades são extremamente complexas, requerem algoritmos e cálculos complexos, requerem que os analistas e desenvolvedores estudem sistemas externos com os quais a aplicação tem que se comunicar, os testes são complicadíssimos e demorados porque cenários muito complexos têm que ser simulados, etc.

É imprescindível considerar a "profundidade do sistema" na hora de estimar a produtividade (se é um sistema "raso", como o primeiro ou se é "profundo", como o segundo), senão acaba se cobrando caro demais pelos sistemas "rasos" e tomando prejuízo nos sistemas "profundos".

Aquelas tabelas que determinam a complexidade baixa/média/alta deveriam ter mais níveis (por exemplo: 5 níveis em vez de 3) e deveriam também ter uma terceira dimensão que quantificasse a "complexidade conceitual" da aplicação, ou seja, o quanto precisamos "queimar as pestanas" pra entender o que tem que ser feito e encontrar uma solução adequada, em oposição a funcionalidades cuja especificação, implementação e teste são essencialmente "trabalho braçal". Note que esta é uma terceira dimensão de complexidade funcional, independente da quantidade de dados e da quantidade de arquivos referenciados, e que influencia muito no esforço.

Sabemos que o PF é o "metro quadrado" do software, mas o que me parece é que software é como uma piscina: os PF medem a área, que é um excelente parâmetro para expressar o tamanho da piscina; só que o esforço corresponde ao volume da água, que depende também de uma característica não está sendo medida, que seria como a "profundidade".

Assunto: Métricas para de Estimativa de Prazo

Data: 08/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3094>

Dúvida: Baseado na contagem do Nesma, utilizamos a fórmula de Caper Jones para estimativa de entrega de um projeto, e encontramos um prazo longo demais. Existem outras fórmulas usadas para estimativa de prazo, em meses? Seria correto fazer esse cálculo pela contagem do Nesma, ou isso poderia gerar um erro muito grande, que aumentaria demais o tempo estimado de entrega do projeto?

Análise: O livro do Capers Jones trata desse assunto. Muitas pessoas pegam modelos em editais e utilizam os mesmos em seus próprios contextos sem realizar uma avaliação técnica e crítica dos mesmos. O próprio autor alerta que os números devem ser calibrados antes de sua utilização às condições locais. Como assim calibrados?

- 1) Selecione as demandas dos últimos 12 meses (ou menos se não houver se passado esse tempo desde o início do registro dos dados de prazo) a partir da base histórica.
- 2) Tabule em uma nova planilha as colunas:
 - a) Ponto de Função – Medição Construção
 - b) Prazo Negociado
 - c) Prazo Realizado
 - d) Prazo para Indicador de Nível de Serviço
- 3) As instruções a seguir são especificar para o MS Office Excel:
 - a) Selecionar as colunas referentes aos dados indicados em 3.a e 3.b
 - b) Inserir gráfico de dispersão
 - c) No gráfico, selecionar a série e inserir linha de tendência tipo potência, exibindo equação e exibindo o valor de R-quadrado no gráfico
 - d) O valor apresentado terá a forma $y=axb$, onde y representa o prazo e as constantes a e b o valor calibrado.
 - e) A partir de 10 demandas, é a quantidade considerada mínima para calibrar "a".
 - f) A partir de 5 demandas, é a quantidade considerada mínima para calibrar "b"

- 4) De maneira análoga ao passo .4, realize os mesmos procedimentos, com a ressalva:
 - a) Selecionar as colunas referentes aos dados indicados em 3.a e 3.c
- 5) De maneira análoga ao passo .4, realize os mesmos procedimentos, com a ressalva:
 - a) Selecionar as colunas referentes aos dados indicados em 3.a e 3.d
- 6) Compare o valor de R-quadrado (apresentado no gráfico como R2) obtido nos passos .4,.5 e .6 considerando que quanto mais próximo de 1,00 melhor a qualidade das constantes (a e b) geradas em cada passo.
- 7) Elabore um estudo comparativo com a avaliação do impacto nas demandas consideradas nos passos anteriores um INS de prazo calculado com as novas constantes a e b propostas ao invés do valor atual (originalmente a=0 e b=0,36).
- 8) Assim como qualquer outro modelo de estimativa paramétrico, esse é tão bom quanto os dados de calibração utilizados.
- 9) Existem vários outros modelos como o COCOMOII, a utilização de tabelas com patamares não lineares em função do tamanho, etc. Vários exemplos podem ser obtidos em editais, muitos disponíveis em <http://www.fattocs.com.br/editais.asp>.
- 10) Quanto a usar estimativas em momentos preliminares para esse fim, não há o menor problema desde que se lembre das incertezas inerentes a mesma. Para tal fim deve ser aplicada uma contingência técnica referente a isso e preferencialmente calculada a partir do MMER (Média da Magnitude do Erro Relativo, onde a Magnitude do Erro Relativo é calculada como $ABS(PF Realizado - PF Estimado) / PF Estimado$).

Assunto: Visões diferentes da mesma funcionalidade

Data: 08/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apy/message/3097>

Dúvida: Há uma aplicação grande que gera visões da mesma funcionalidade de consulta de acordo com o perfil do usuário. Um advogado e um procurador, ao executarem a mesma funcionalidade de consulta (CE), possuem visões distintas:

1. Perfil de advogado:

Arquivo referenciado (AR): ALI USUARIO (A entidade filho é o TR ALI_PERMISSAO_ADOVADO)

Consulta mostra 5 campos (DER)

2. Perfil de procurador:

Arquivo referenciado (AR): ALI USUARIO (A entidade filho é o TR ALI_PERMISSAO_PROCURADOR)

Consulta mostra 6 campos (DER)

Conta-se as duas consultas (CE) como PE (Processo elementar) idêntico? Ou a unicidade do PE foi quebrada pelo fato de se possuir menos campos na visão do advogado?

Análise/Conclusão: Quando a pessoa que abre uma demanda solicitou as consultas, é bem provável que ela não tenha dito: eu quero uma consulta para procuradores e outra para advogados. Porém, quando o desenvolvedor foi implementar, ele viu que havia uma consulta para advogado e outra para procurador. Imagine que decidíssemos mudar totalmente a consulta feita por um procurador, isso produziria alguma alteração em uma consulta feita por um advogado? Provavelmente não, e vice-versa. Alterando a consulta feita por um advogado, não altera em nada a consulta feita por um procurador. Dessa forma, é fácil verificar que esses processos são independentes.

Dúvida:

- 1) Então mesmo que os campos da tela de advogado e procurador sejam os mesmos, e os ALIs os mesmos, considerando o que diz o CPM, devo considerar lógicas de processamento diferentes de acordo com as permissões dos usuários (advogado e procurador)?

Mesmo porque podemos imaginar que os campos da tela de Consulta dos advogados pode mudar completamente sem afetar a tela de Consulta dos procuradores - são lógicas mesmo diferentes.

Análise/Conclusão:

- 1) Só e somente só lógicas de validação (1) e análise de condições para ver se as aplicações (2) sejam particulares e específicas da consulta em análise. Sejam elas parte de uma infraestrutura comum que proveja essa funcionalidade temos um requisito não funcional que é desconsiderado na análise. Por exemplo:

- 1) Existe uma tela que apresenta 10 campos dependendo da configuração dos direitos do usuários alguns campos lhes são exibidos nessa tela outros não. Nesse caso temos uma infraestrutura geral que fornece em qualquer tela do sistema esse recurso.

- 2) Existe uma tela que apresenta 10 campos. Especificamente numa tela de consulta é programado que quando o usuário tiver determinado perfil alguns dados não lhe serão apresentados. Nesse caso temos uma funcionalidade específica do usuário à essa tela.

O analista de pontos de função deve basear a sua análise em evidências materiais ou entrevistas com os donos dos requisitos. Hoje se há os mesmos campos na tela, os mesmos arquivos referenciados, todos os registros são igualmente passíveis de consulta; não há como justificar haver mais que uma consulta. Agora se a tela tem como campos a serem apresentados um determinado universo e, conforme o perfil, um ou outro subconjunto desse universo é apresentado, não pode se considerar que sejam os mesmos campos! O projeto aproveita um único item de configuração para atender dois diferentes requisitos e daí o comentário de haver duas funções.

Dúvida: Esse mesmo conceito serve para EE? Considere que há um formulário (Entrada Externa) que a visão é diferente para cada perfil. Sendo assim, podemos contar uma EE para o Perfil A e outra para o Perfil B,?

Análise/Conclusão: Sim, o conceito de permissão serve para qualquer PE (Processo Elementar - EE, CE ou SE).

Dúvida: Quando se afirma que se trata de uma CE, seria com a finalidade de montar um menu dinâmico? Pelos ALI acessados, há a impressão que Advogado tem um conjunto de permissões e Procurador outro conjunto.

Caso esteja montando um menu dinâmico em função dos níveis de acesso (por ex: Advogado, Procurador, Comum, etc) ainda que se tenha as permissões distribuídas em vários repositórios/campos, a CE (se é que vai ser pontuada assim) seria uma só, e o processo elementar seria, por exemplo, LOGIN.

Análise/Conclusão: Interessante a colocação sobre a funcionalidade de LOGIN já construir o menu dinâmico de acordo com o perfil do usuário. Pode ser.

Porém, o foco do problema levantado é sobre número de campos (TD) e repositórios (AR) diferentes em consultas de diferentes usuários (consultas chamadas por estes menus).

No caso do Sistema em análise, podemos entender cada LOGIN como tendo acesso a uma diferente lógica de processamento. Cada consulta exhibe os campos de acordo com a programação daquele perfil específico. Inclusive os ALI são diferentes: ALI PERMISSAO_ADVOGADO, ALI PERMISSAO_PROCURADOR.

O exemplo citado anteriormente foi o de um único ALI PERFIL - requisito geral do Sistema, que já define os acesso dos usuários. No caso exposto na primeira dúvida da discussão, existem funcionalidades específicas de acordo com o perfil do usuário. É como se cada LOGIN com perfil diferente iniciasse uma nova aplicação, ou melhor, uma nova Visão do Negócio da Aplicação para o usuário que entrou.

Dúvida: Há uma dúvida a respeito da definição e de como calcular o Tipo de Registro das Funções de Dados. Alguém teria um caso prático para exemplificar?

Análise/Conclusão: Imagine uma entidade em um modelo conceitual que após normalização em terceira forma normal fica com três entidades. Por exemplo, compromisso (entidade no modelo conceitual) vira Cabeçalho do Compromisso; Itens de Planilha de Pagamento; e Rateio de Despesas. Haverá um AL (ALI ou AIE) e três TR.

Assunto: APF e Gerenciadores de Conteúdo

Data: 11/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3103>

Dúvida: Existe uma dúvida a respeito do uso da APF para mensurar Gerenciadores de Conteúdo. Mais especificamente, como mensurar manutenções em um CMS (Joomla)?

Análise/Conclusão:

1) Conte o sistema (Joomla) em questão antes de dar manutenção. Assim teremos o tamanho do sistema em produção ou antes de mexermos.

2) Sobre o escopo/requisito da manutenção/alteração, escolha um método (IFPUG ou NESMA) e conte as alterações usando como linha de base a contagem do sistema que se fez no passo 1. Esta contagem será o tamanho do projeto/iteração.

Assunto: Processo Elementar

Data: 18/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3134>

Cenário: Para uma aplicação de controle de solicitações, o dono do negócio solicitou que o sistema informe ao usuário solicitante através do envio de e-mail, a situação atual da solicitação toda vez que a mesma mudar de status (aberto, pendente, em solução, concluído, cancelado). Nesse caso, devemos contar a funcionalidade "Notificar usuário solicitante sobre situação da solicitação", como está no documento de requisitos e contar uma CE como único processo elementar ou devemos contar apenas os grupos lógicos envolvidos em cada funcionalidade que envolver a mudança de status da solicitação?

Análise/Conclusão: Para definir o tipo de função de transação é necessário saber qual é a intenção primária do Processo Elementar, ou seja, o que realmente o usuário deseja que a funcionalidade execute. Como citado, o título da funcionalidade é "Notificar usuário solicitante sobre situação da solicitação", e o mais importante é a consulta, e não a atualização do *status* da solicitação. Interpretando desta forma, o Processo Elementar é uma **SE** (pois se envia um e-mail - apresentação de dados e atualiza-se um arquivo - o *status* da solicitação é atualizado em algum ALI da aplicação). "Atualizar status" é apenas um passo do processo elementar.

Porém, se existem várias funcionalidades no Sistema que alterem o *status* da solicitação e notifiquem o usuário de forma secundária, estes dois procedimentos podem ser apenas passos dessas várias funcionalidades que são os Processos Elementares descritos como "Funcionalidades de cada um dos grupos lógicos". Então, seriam várias funcionalidades, ou seja, vários PE, de acordo com o grupo lógico de dados envolvido (AR). "Notificar usuário solicitante sobre situação da solicitação" seria apenas um passo desses Processos Elementares. Não seria um PE.

Assunto: Processo Elementar

Data: 19/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3138>

Cenário: Em um sistema existe uma tela de pesquisa que após a busca dos dados, o usuário tem a possibilidade de ordenar o resultado utilizando alguns critérios. Podemos considerar essa ordenação como processo elementar? Considere que uma consulta externa referente a busca já foi contada, e a ordenação é feita na camada de interface.

Análise/Conclusão: Pela descrição do cenário, a ordenação do resultado faz parte da lógica de processamento do processo elementar "Pesquisar" do Sistema.

A Ordenação é uma das lógicas de processamento citadas no **CPM 4.3.1, PARTE 2, CAP. 7-6 e 7-7**. Na página 7-7, lê-se que *"Este forma de processamento lógico (dados reclassificados ou rearrumados) não implica na identificação de tipo ou contribuição na unicidade de um processo elementar; ou seja, a orientação dos dados não constitui unicidade."* Portanto, a ordenação não deve ser considerada como um processo elementar.

Assunto: Contagem de Campos de Somatório

Data: 21/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3149>

Dúvida: Como devemos contar os campos de somatório de um relatório que tem vários totais (para cada coluna)? Devemos contar somente um TOTAL ou todos os TOTAIS ?

Análise/Conclusão: Deve-se contar todos os totais. Cada total tem o seu próprio significado e possui informações diferentes. Qualquer campo significativo para o negócio do usuário em tela deve ser contado.

Assunto: Fator de Ajuste

Data: 21/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3150>

Dúvida: Onde está a coerência em utilizar o valor mínimo do fator de ajuste, que é o valor 0,65 sobre os pontos de função não-ajustados, se teremos uma redução de 35% na pontuação final? Qual é a utilidade disso? Não seria então melhor se não aplicarmos esse fator e permanecer com os pontos de função não-ajustados?

Análise/Conclusão: Deixando um pouco mais claro, o fator de ajuste tem a responsabilidade de medir e indicar as funcionalidades gerais fornecidas ao usuário da aplicação. Conforme o CPM 4.2.1, deve ser aplicado para obtermos os Pontos Ajustados de uma contagem, seja ela do tipo desenvolvimento, melhoria e ou aplicação. A sua aplicabilidade deve ser feita não em teorias e sim com responsabilidade em uma contagem, pois a sua aplicação ajusta +/- 35% o tamanho de sua aplicação. Portanto, quando tivermos que utilizar do VAF, utilizemos com objetivo de medir as funcionalidades gerais e não na tentativa de aplicar um valor mínimo ou máximo para ajustar a contagem.

Dúvida: Gostaria de saber uma situação real sobre o impacto dos -35%.

Análise/Conclusão: Os seguintes passos descrevem como obter o VAF:

1 - Avalie cada uma das 14 características gerais do sistema na escala de 0 a 5 para determinar o nível de influência (NI).

2 - Some os níveis de influência das 14 características gerais do sistema para produzir o total do nível de influência (TDI).

3 - Substitua o valor do TDI na seguinte fórmula para produzir o fator de ajuste.

Fórmula VAF = (TDI * 0,01) + 0,65

Por exemplo, o seguinte fator de ajuste é calculado se houver o nível de influência 0 para cada uma das 14 GSC (0 * 14)

VAF = (0 * 0,01) + 0,65

VAF = 0,65

Podemos derivar da fórmula que existe um intervalo mínimo e máximo para o VAF de 0,65-->1,35. Para se obter o valor máximo, todos os níveis de influência de uma aplicação deverá receber valores igual a 5. Pegando o exemplo acima, imaginamos que foi obtido os pontos para um projeto XZW e o valor de pontos não-ajustados é igual a 100 PF.

Calculamos: $DFP = (UFP + CFP) * VAF$; $DFP = 100 * 0,65$, temos 65 PF.

Podemos observar que nesse exemplo tivemos uma redução de (-)35% no tamanho do projeto de desenvolvimento, após ser ajustados os pontos de função.

Assunto: Funcionalidade de Conversão

Data: 22/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3161>

Cenário: Em um sistema de RH antigo, os dados de funcionários eram armazenados em um Banco de Dados e as suas fotos e certificados de cursos eram armazenados em um servidor de arquivos. Não havia funcionalidade de *upload* destes arquivos no sistema de RH: os documentos eram escaneados e armazenados manualmente por fora do sistema, sendo nomeados com o número de matrícula do funcionário. No novo sistema (não é uma evolução do antigo) há funcionalidade de *upload* e os dados e arquivos são armazenados no Banco. Na época da implantação do sistema novo, foi desenvolvida uma rotina para mover os dados e arquivos dos funcionários do Banco e do sistema de arquivos para o novo Banco.

Identificamos dois processos elementares de conversão para cada tipo de documento. Por exemplo, para as fotos: uma CE para buscar o registro de funcionário no banco antigo que se refere uma foto no servidor de arquivos e uma EE para carregar estes dados no novo banco. Essa análise está correta?

Análise/Conclusão:

Funcionalidades de Conversão - As funcionalidades de conversão consistem em pontos de função fornecidos por qualquer funcionalidade de conversão solicitada pelo usuário, fonte CPM 4.2.1.

É muito comum surgir em projeto de desenvolvimento onde o novo projeto irá substituir uma aplicação legada, e nessas situações os usuário necessitam de garantir que o novo projeto tenha os dados na sua parcialidade ou totalidade.

Podemos classificar esse tipo de funcionalidade como transações descartáveis, onde será utilizado apenas uma vez para dar carga nos dados ao novo sistema e posteriormente não tem mais utilidade para a nova aplicação.

Dentro do cenário descrito acima, podemos contemplar apenas uma EE- Funcionário Conversão - Importação (Apenas).

Segue um exemplo do próprio CPM 4.2.1 :

Requisitos do Usuário: O usuário havia comprado um novo pacote do Sistema de RH. O usuário solicita a habilidade de migrar informações existentes do funcionário (Nome, CPF, Número de dependentes, Código do Tipo, Nível de Supervisão, Taxa Padrão da Hora, Número da Unidade coletiva de Negociação e Nome do Local) para a nova aplicação.

Questionamos:

1. A Função de Transação satisfaz os requisitos de um Processo Elementar? Sim
2. A Função de Transação satisfaz a Intenção Primária de uma EE? Sim. A Intenção Primária é manter um ALI.
3. Os dados ou informações de controle são recebidos de fora da fronteira da aplicação ? Sim. Os dados do arquivo funcionário da aplicação de RH antiga atravessam a fronteira.
4. Pelo menos um ALI é mantido se os dados que entram pela fronteira não são informações de controle que alteram o comportamento do sistema? Sim. O ALI Funcionário é mantido.

Conclusão: Existe 1 EE.

Dúvida: Considerando que parte dos dados vem de um sistema legado, e outra parte (fotos e certificados) vem de um servidor de arquivos (sem sistema relacionado a eles), a atividade de relacionar os dois conjuntos (pelo número de matrícula constante no nome do arquivo) não chega a constituir um processo elementar?

Análise/Conclusão: Existe apenas um processo elementar, que seria uma EE para conversão dos dados. A

consulta seria parte deste processo. Ela não existe por si só, ou seja, para o usuário consultar os dados existentes não significa nada. O processo só fica consistente e auto contido se após a consulta os dados forem gravados na nova base. Também não devemos considerar relevante o fato de as informações estarem separadas (parte em BD, parte em uma estrutura de diretórios), porque consideramos isto como um requisito técnico utilizado no armazenamento de dados, que deve ser desconsiderado para contagem PF.

Assunto: Contagem para Relatórios que atualizam dados

Data: 25/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3168>

Dúvida: Num sistema, temos um *job* (relatório) que atualiza os dados em uma tabela, recuperando-os de outra. Porém, esse relatório não apresenta nenhum dado para o usuário. Podemos considerar essa contagem como uma Entrada Externa?

Análise/Conclusão:

- 1) Outro sistema que interage com o sistema em análise é um usuário.
- 2) Cuidado para não contar uma réplica (não funcional) em termos de PF. Se tratar-se de uma réplica o que se conta é um AIE.
- 3) Em se tratando de uma passagem de movimento, tipicamente se conta uma SE para o sistema que envia os dados e uma ou várias EE para o sistema que a recebe.

Visite o link para mais detalhes: <http://www.fattocs.com.br/freemind/APF.html>

(Escolha a opção Práticas de Contagem e depois Dados Compartilhados.)

Dúvida: No caso do item 3, como não há apresentação de dados para o usuário, mas somente a atualização de tabelas (porém a atualização é realizada através de um processamento *batch*) podemos contar uma Entrada Externa?

Análise/Conclusão: Muitas vezes acessamos diretamente uma tabela de sistemas externos ao sistema em análise não com o propósito de obter dados para validação e referência, mas sim com o propósito de receber dados externos e internalizá-los pelo processamento de uma entrada externa do sistema em análise.

Vamos analisar: Por exemplo, existe uma tabela com a variação cambial de várias moedas e cestas de moedas mantidas externamente à aplicação em análise. O sistema em análise requer receber do mundo do usuário externo a ele um movimento de transações com a variação cambial, para a partir dela atualizar a base de lançamentos na contabilidade. Esse é o requisito funcional, isso é o que deve ser feito.

Podemos ver 5 cenários de implementação desse requisito (de como ele pode ser feito):

- 1) Enviar um arquivo contendo em seus registros as informações necessárias para processamento pela aplicação a ser atualizada;
- 2) Enviar mensagens XML para processamento pelo sistema a ser atualizado por meio de um *Web Service*;
- 3) Abrir um *socket* TCP/IP para um *daemon* que estabelecerá a conexão entre as aplicações e a partir daí enviar mensagens com os dados;
- 4) Gravar um registro numa tabela e a aplicação que atualizará a base de lançamentos ler dessa tabela e a partir dos dados lá presentes processar a atualização (*rug*);
- 5) O processamento de atualização lê os dados diretamente da base de cotações ;

A modelagem da solução não deve ser relevante na sua análise. Se ao modelar a solução, ao invés de receber uma mensagem com essa cotação, ou receber um arquivo com essa cotação, utiliza-se da opção (4), o sistema que mantém esses dados de cotação é o usuário da aplicação que atualiza os lançamentos, obviamente um AIE não é contado referente aos dados recebidos.

CONTRASTE: Existe uma entrada de dados on-line. Por motivos de desempenho, parte do processamento é feita a posteriori. Esse "a posteriori" é o seu *job batch*..Conte como se a lógica de processamento fosse parte da entrada de dados on-line.

Assunto: Como mensurar um jar através de PF

Data: 25/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3175>

Dúvida: Foi solicitada a construção de um jar (função desenvolvida em java, que será chamada por uma função Oracle). Esta função que será desenvolvida terá como atributos de entrada uma lista de arquivos (independente do formato, pode ser .pdf ou .rtf ou .doc ... etc). A função deverá transformar todos os arquivos em PDF e concatenar todos os arquivos em um único arquivo (ou seja, um único PDF contendo todos os documentos passados como atributo). Este arquivo deverá ser gravado em uma área dentro do *file system*. Ou seja, a função simplesmente retornaria uma mensagem de sucesso após a criação deste arquivo. Como podemos mensurar isto em pontos de função?

Análise/Conclusão: Existe algumas situações que encontramos no nosso dia a dia que a APF na sua totalidade não consegue resolver todos os nossos problemas. Devemos recordar que a APF mede requisitos funcionais dentro da perspectiva e/ou visão dos usuários. Nesse exemplo, se medirmos essa funcionalidade teremos uma funcionalidade (SE) de baixa complexidade e lógica de processamento muito complexa que poderíamos tentar ajustar nos fatores de ajuste. Mas, nessa solução temos mais características técnicas que não é atendido na sua totalidade pela APF. Nesse tipo de situação é bastante comum termos calibragem nessas funcionalidades que possibilite a medição, pois o nosso objetivo não é penalizar quem vai desenvolver e muito menos o cliente que vai receber essa implementação. Negocie essa calibragem com o cliente, derivando da APF para um número de horas que serão gastas na implementação dessa funcionalidade.

Imaginando que baseado na sua expertise teremos:

Produtividade Java = 12 H/PF

Esforço = 180 H (para realizar o desenvolvimento)

Aplique na fórmula
$$\text{Esforço (Hh)} = \text{Tamanho (PF)} \times \text{Taxa de Entrega (Hh/PF)}$$
 e encontre o tamanho da sua

funcionalidade.

Assunto: Função de Dados

Data: 26/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3180>

Dúvida: Considere a seguinte regra: “Dados Elementares repetidos contam apenas uma vez”.

Essa regra não referencia apenas o nome do campo (atributo) da tabela, mas sim o significado do mesmo, mesmo que tenha outro nome? Ou seja, se eu tiver uma tabela que possui telefone de contato 1, telefone de contato 2, para o mesmo cliente, eu contaria apenas 1 Dado Elementar para o atributo telefone? ou 2 Dados Elementares?

Análise/Conclusão: A forma de implementação não importa neste caso. Sendo assim, de acordo com o exemplo citado, seria apenas um TD contado. O mesmo vale para campo de entrada de dados na tela (ex: e-mail/confirmação de e-mail - conta 1 DER apenas).

Assunto: Contagem de ALI

Data: 28/10/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/3185>

Dúvidas:

1) Para realizar um determinado processo o usuário deve utilizar um certificado digital (token). A aplicação captará desse dispositivo o número do CPF do proprietário do token e o utilizará para preencher um determinado campo. Esse procedimento faz do token um AIE na aplicação que está sendo contada? Em caso positivo, essa consulta deve ser contada como uma CE?

2) A aplicação deve guardar um log de todas as operações tipo CRUD do usuário. Devemos contar esse repositório como um ALI ou podemos considerar como um requisito não funcional, não contando nada?

3) Validações do tipo: “ver na base se já existe o registro com mesmas características”, “verificar CPF no SERASA” também não contam PF?

4) Considere um campo CEP, que ao ser informado consulta numa base externa o valor para preencher automaticamente os dados do endereço. Devemos considerar apenas essa base como AIE e não contar essa consulta?

Análise/Conclusão:

1 – O token não é um AIE. A informação do token é de domínio do usuário. Em vez de trazer informações no token, poderia ser o caso de ele preencher um formulário com as informações do token. O que muda é a forma como o usuário entra com essa informação no sistema. Imagine que em vez de buscar o CPF no token o sistema buscasse as digitais do usuário. Via de regra, o arquivo associado a esses processos de validação de usuário por certificado ou por biometria é o próprio arquivo de usuários da aplicação. O processo a ser contado é o "determinado processo" citado acima.

2 - As informações de auditoria via de regra servem para satisfazer algum requisito de qualidade ou técnico do sistema. Caso haja processo de negócio que referencie essas informações, a situação pode mudar. O caso deve ser analisado nessa perspectiva.

3 - Ver na base ou verificar CPF no SERASA significa fazer uma consulta em arquivo, logo os arquivos consultados devem contribuir para a complexidade da funcionalidade em análise.

4 - Se buscar o CEP faz parte de um outro processo realmente só se deve contar o AIE dos correios.