

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

Nessa síntese foram abordados, em 65 mensagens, os seguintes assuntos:

- Contagem de botões de navegação
- Uma consulta que possui em contador incrementado deve ser classificada como uma Consulta Externa ou como uma Saída Externa?
- Alinhamento entre Processos Elementares e Casos de Uso
- Dúvida na contagem de *combobox*
- Mudança no comportamento do sistema
- Contagem de User Story

Assunto: Contagem de botões de navegação

Data: 01/10/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/4430>

Dúvida: Uma barra de botões de navegação em registros de bancos de dados com os botões primeiro, anterior, próximo, último pode ser contada como um TD? Ou não deve ser contada em uma função do tipo transação?

Análise/ Conclusão: Essa barra serve para acionar a consulta que exhibe os registros. Podemos considerar isto como um meio de acionamento desta consulta, e neste caso conta-se um único TD para todas as opções de navegação (pois todas acionam a mesma consulta).

Assunto: Uma consulta que possui em contador incrementado deve ser classificada como uma Consulta Externa ou como uma Saída Externa?

Data: 05/10/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/4439>

Dúvida: Um concurso considerou como verdadeira a seguinte afirmação: Uma consulta que possua contador incrementado é considerada uma Saída Externa.

Porém, o CPM, que define as regras de Pontos de Função, diz que paginação (incremento) não muda o tipo de transação (de CE para SE):

- ▲ Não conte variáveis de paginação (que é uma consulta incremental) ou “selos” gerados pelo sistema. Variáveis e “selos” gerados pelo sistema incluem:
 - Números de páginas;
 - Informações de posicionamento tais como “Coluna 37 a 54 da linha 214”;
 - Comandos de paginação tais como anterior, próximo, setas para cima e para baixo em uma aplicação GUI.

Sendo assim, essa consulta não deveria ser classificada como uma Consulta Externa?

Análise: Haver aquele paginador em uma tela de pesquisa realmente não faz com que o PE seja considerado como uma Saída Externa. Porém não está claro o que é um "Contador Incrementado". Será que realmente está falando do paginador? Mas se não há mais nenhuma informação complementar, então podemos considerar essa informação como sendo insuficiente para considerar o PE como uma Saída Externa.

Análise/ Complemento: Pode se tratar de um contador de consultas, ou seja, quantas vezes aquela consulta foi realizada, e não um contador para paginação. Se for esse o caso, aí sim o PE seria uma Saída Externa, pois mantém um ALI sempre que a consulta é realizada.

Assunto: Alinhamento entre Processos Elementares e Casos de Uso

Data: 06/10/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/4450>

Dúvida: Tendo em vista que a identificação do Processo Elementar é de suma importância para a APF, já foi estabelecida alguma relação para fins de alinhamento do conceito de Processo Elementar para a técnica de representação de requisitos por Casos de Uso?

Normalmente é difícil decidir se um determinado conjunto de interações constituem um caso de uso ou vários casos de uso.

Imagine o seguinte cenário: Sistema de Reciclagem de Material

Exemplo: - Inserir latas e garrafas

- Solicitar impressão do recibo

Quantos casos de uso podemos identificar neste exemplo?

Resposta: Apenas 1, pois são processos elementares dependentes.

Quantos processos elementares para a APF?

Resposta: 2, um componente de entrada e um de saída .

Existe alguma discussão/orientação/estudo quanto à melhor maneira de representar casos de uso de modo a facilitar a medição pela APF?

Análise: Casos de Uso e Processos Elementares não são a mesma coisa. Pode ser que um caso de uso represente um PE, mas não é regra. Observe que se um Caso de Uso utiliza {INCLUDE} ou {EXTEND} é uma indicação de que esses casos de uso formam um único processo elementar.

Dúvida: O ponto é que o cenário descrito é real e o consultor que está apoiando a empresa afirma que o mesmo deva ser representado em 1 Caso de Uso e jamais em 2, pois são processos dependentes e o segundo não existe sem o primeiro. Mas para este cenário, existem 2 processos elementares para a APF. Quanto à colocação acima, este processo elementar é a definição de processo elementar para a APF ou para a técnica de representação de requisitos por casos de uso? Poderia explicar melhor?

Análise: Não é possível apontar alguma literatura que proíba o analista de requisitos de separar em casos de uso distintos dois processos dependentes. A premissa de ser o mais simples, objetivo e significativo para o usuário é universal, mas nada impede de agrupar os casos de uso de formas distintas. Toda a literatura já lida da engenharia de requisitos sempre colocou que a quantidade de casos de uso utilizada em um projeto, independentemente do negócio, será sempre muito particular e dependerá fortemente da visão do analista de requisitos e seu alinhamento com o usuário. Não existe regra que proíba que esses dois casos de uso, por serem dependentes devam necessariamente ser representados em um único. Tal como foi colocado, na ocorrência de haver *includes* ou *extends*, pode sim, ter processos elementares para a APF mapeados para mais de um caso de uso.

Assunto: Dúvida na contagem de *combobox*

Data: 07/10/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/4451>

Dúvida: Considere um *combobox* que lista Tipos de Usinas (não são dados de código). Este combo é exibido em três telas do sistema:

- ⤴ Na tela 1 são listadas somente as Usinas com situação igual a ativa
- ⤴ Na tela 2 são listadas as Usinas ativas e inativas
- ⤴ Na tela 3 são listadas as Usinas do tipo Hidrelétricas, por exemplo

Devemos contar apenas uma consulta ou três?

Análise: Pelo o que foi dito, devemos contar 3 CE's pelo motivo de a lógica de processamento de seleção ser diferente. Se elas são diferentes, provavelmente existe uma razão negocial para tal.

Dúvida: Não poderíamos contar apenas uma consulta, considerando que o que difere ativo de inativo é apenas uma questão de valores, pois a lógica de processamento "4. Dados são filtrados e selecionados segundo critérios especificados para comparar vários conjuntos de dados" continua a mesma, já que o campo utilizado para a seleção dos registros (filtro) continua o mesmo, só tendo seu valor mudado? Quanto a selecionar as usinas do tipo hidrelétricas, essa lógica pode estar mudando, já que o campo pesquisado é outro.

Análise: Não há motivo para considerar a seleção de ativos ou todos (ativos e inativos) como sendo uma única consulta externa. Nestas duas consultas, os TD's são os mesmos, os ALR's são os mesmos, porém a lógica é diferente.

Transformando a cláusula SQL das 3 consultas em um texto comum:

- 1º Selecionar todas as usinas ativas.
- 2º Selecionar todas as usinas ativas E inativas.
- 3º Selecionar todas as usinas que sejam Hidrelétricas.

A lógica de processamento é diferente. Se existe uma combo de usinas ativas e outra que traz ativas e inativas, isso significa que o atributo "ativo" não é algo técnico, mas sim uma informação reconhecida pelo usuário, que inclusive pediu uma lista com e outra sem o filtro. O atributo "ativo" ou "inativo" somente não seria motivo para diferenciação de PE's se o atributo fosse algo técnico.

Complemento: Se o usuário pediu 3 consultas distintas, são 3 PE's. Até poderia ser o outro caso, se estivéssemos falando de uma consulta onde o usuário pudesse filtrar quais informações ele quer de retorno. Nesse caso, ativo e inativo e até o tipo de usina seriam parâmetros de filtro de uma única consulta. Mas já que estamos falando de 3 combos com retornos específicos, não é o caso. Tecnicamente, as 3 podem ser só uma consulta onde no componente você define o parâmetro de filtro para a *combobox*. Só que aí estamos saindo da "fronteira" de APF. Algo que pode confundir um pouco é ter se referido ao componente (*este combo* aparece em 3 telas). Se ignorássemos a parte do combo e considerássemos uma lista para usinas ativas, uma lista para usinas ativas e inativas e uma lista para hidrelétricas, estaríamos falando de 3 listas diferentes.

Assunto: Mudança no comportamento do sistema

Data: 26/10/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/4485>

Dúvida: Aquela função de "*congelar painéis*" das planilhas eletrônicas poderia ser considerada como um exemplo de processo elementar que muda o comportamento do sistema?

Análise: No CPM, há um exemplo de contar como um processo elementar salvar os parâmetros de uma consulta. Ele conta o respectivo arquivo como um ALI. O CPM também estabelece que um ALI pode

conter não só dados mas também informações de controle. Na parte 3, exemplifica como dados de referência passíveis de contagem como ALI ou AIE, parâmetros operacionais do negócio.

Independentemente dessas considerações e ao avaliar uma aplicação em particular, deve-se verificar se congelar painéis não é uma funcionalidade geral provida por uma infraestrutura anterior (provavelmente será isso). É uma característica que tipicamente se pede em diversas consultas apresentadas em quadros e não apenas em uma em particular. Se esse não for o caso, há uma série de informações de contexto que ficam guardadas em algum lugar e são modificadas por esse “congelar painéis”. Há um ALI sendo mantido com a intimação de quais painéis estão congelados e quais estão livres. Mas em termos de requisitos, não parece um requisito ligado às tarefas e serviços do usuário, estando na esfera de uma restrição de usabilidade.

Assunto: Contagem de *User Story*

Data: 27/10/2011

Link: <http://br.groups.yahoo.com/group/livro-apf/message/4490>

Dúvida: Considere uma empresa que utiliza a metodologia *Scrum* para gerenciar o desenvolvimento. Sendo assim, os requisitos são escritos em forma de *User Stories*. No entanto, para facilitar a conversa com os clientes no momento da priorização do *backlog*, principalmente, é por bem realizar a estimativa dos requisitos utilizando Pontos de Função, uma vez que não precisaria reunir o time todo (esse serviço ficaria a cargo do PO) e seria possível “padronizar” as estimativas, com uma técnica documentada e aceita. Como é a melhor maneira de fazer estimativas de APF, a partir somente das *User Stories*?

Análise: Uma pergunta a ser feita é: Quem deve estar no controle da medição dos resultados? Os usuários (clientes) ou os desenvolvedores (fornecedores)?

Dúvida/Complemento: A ideia do uso da APF é ter uma forma padronizada de dimensionar o tamanho das solicitações dos usuários, uma vez que nas reuniões de planejamento de *release* isso pode ser um fator determinante sobre o que entra ou não em uma determinada versão. O cenário passado é para desenvolvimento/manutenção de softwares “de prateleira”. Nesses casos, temos um “Comitê do Produto” com representantes da área comercial e do suporte técnico, basicamente. É com esse “comitê” que o PO negocia a prioridade dos itens do *backlog*. Apesar da importância do ROI, muitas vezes a primeira pergunta que é feita no comitê é: “Quanto tempo leva?”. A ideia do uso de APF é poder responder essa pergunta, mesmo que de maneira estimada, mas tendo uma forma padrão de estimativa entre os diversos PO's que podemos ter, visto que tem mais do que um produto nessa mesma situação.

Análise: A APF não é a melhor maneira de estimar o tamanho das histórias entre *sprints* pois existem alguns problemas de contagem principalmente relacionados a escopo, ou funções que já foram contadas e são novamente manipuladas, também questões relacionadas a *refactoring* que tem que ser estimadas e mesmo correções que podem entrar em um *sprint*. Os pontos de história são uma medida da complexidade, do esforço necessário para se desenvolver algo até o estado de pronto, assim parece mais adequado para estimativas entre *sprints*. Além disso, existe o fator psicológico. Quando a equipe estima o esforço ela passa a estar comprometida com aquela previsão.

Os PF são uma medida funcional que não leva em conta por exemplo: se usamos um *framework* padrão ou um desconhecido da equipe, se teremos que realizar testes para um ou dez perfis de usuário diferentes, ou se haverá necessidade de integração com um ou dez bases diferentes usando protocolos diferentes. As iterações são curtas e pequenos desvios podem significar grandes problemas.

Dúvida/ Complemento: Realmente, não estamos pensando no uso de APF para estimar as histórias para desenvolvimento. A questão maior é antes disso, ou seja, criar uma maneira padrão dos *Product Owners* estimarem o tamanho de cada solicitação do “Comitê do Produto”, sendo que muitas vezes a ordem de execução acaba sendo determinada pelo tempo. Quando um recurso for entrar em um *sprint*, será discutido pelo Time Técnico e terá sua estimativa feita em pontos de história.

O que acontece é que como trabalhamos com softwares de prateleira, esses comitês são formados pelas diferentes áreas (comercial, suporte, etc). Cada área traz suas sugestões, e cabe ao PO manter o *backlog* do produto.

O que acontece hoje é o seguinte: Como o comitê tem representantes de áreas diversas, muito do trabalho do PO consiste em "negociar" com as diversas áreas que trazem solicitações ao comitê, sobre o que vai entrar ou não em cada *release*. O que acontece muitas vezes é que nessa negociação, um dos fatores de barganha acaba sendo o tempo, visto que diversas solicitações tem o mesmo grau de importância para o produto, por exemplo: O comercial pediu o recurso X, e o suporte necessita do recurso Y. Nenhum dos dois é prioritário, mas o PO sabe que no próximo *release* somente um deles entrará em execução (por uma restrição qualquer). Dessa maneira, a negociação acaba caindo para o prazo : "Caso façamos apenas o recurso Y, o novo *release* ficará pronto em tal data ... Enquanto se fizermos o recurso X essa data será um pouco mais adiante." Muitas vezes essa informação é crucial, principalmente pela janela de oportunidade (tanto do suporte como do comercial). A dificuldade tem sido encontrar uma maneira com que o próprio PO consiga estimar (de maneira não empírica) o tamanho de cada solicitação. Com base nessa estimativa, e na velocidade do time nos últimos *sprints* desse produto, a ideia é que possamos passar um prazo aproximado para o comitê, de maneira que "todos saiam felizes" e o próximo *release* não demore muito a sair.

Análise/Conclusão: Com alguns sistemas, apesar de não serem de prateleira, há problema semelhante. Temos equipes dedicadas, vários clientes de órgãos diferentes para serem atendidos. Neste caso o projeto não tem fim, as demandas chegam continuamente, podendo ser corretivas, adaptativas ou novos desenvolvimentos. Estamos usando o que o mercado vem chamando de SCRUMBAN, pois é, as demandas chegam como no kaban, desestruturadamente, vai se formando um *sprint backlog*. É claro, corretivas urgentes e demandas legais passam a frente de tudo. Formamos filas priorizadas para cada tipo de cliente, ex.: financeiro, comercial, etc.

Na reunião de início de *sprint* as demandas mais prioritárias são estimadas pela equipe e aí o PO cria o *backlog* do *sprint*, juntando as demandas priorizadas que fazem sentido estarem juntas e que cabem no *time box* do *sprint*.

Mas no caso citado, a APF não vai ajudar. O problema apresentado não se aplica nem tão diretamente em relação à APF, mas sim à cultura da Organização, que terá que ser um pouco alterada por conta da nova metodologia.