

## Síntese das discussões do fórum Livro-APF: Abril/2011

Nessa síntese foram abordados, em 96 mensagens, os seguintes assuntos:

- Contagem em manutenção de Sistemas Legados
- Processos Elementares distintos de acordo com a Lógica de Processamento 5
- Quando podemos considerar um Arquivo, como um Arquivo Lógico Interno?
- Unicidade do Processo Elementar
- Como contar relatórios com *layouts* únicos, porém com validações distintas?
- Tratamento de Não Conformidade pela APF
- Manutenção em ALI: é uma alteração nos dados ou na estrutura do ALI?
- Mudança de cor de um campo e mensagens enviadas ao usuário: devem ser contados como Tipos de Dados ou são Requisitos de Qualidade?
- Contagem de cadastro dividido em várias telas
- Como contar o envio de mensagens automáticas e rotinas executadas automaticamente?
- Contagem de *Upload* de Arquivo
- Determinação da fronteira na contagem de Projetos SOA
- Contagem da manutenção de cadastros que só possuem Código e Descrição

**Assunto:** Contagem em manutenção de Sistemas Legados

**Data:** 01/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-afp/message/3594>

**Dúvida:** Os sistemas que serão contados já existem, nunca foram contados e normalmente sofrem apenas pequenas manutenções.

Na página 69 do livro tem um exemplo simples onde é contado um sistema de pontos de presença. Supondo que esse sistema já exista e esteja em funcionamento há muito tempo, nunca tenha sido contado, e chega o seguinte requisito: Permitir que o relatório de presença possa ser gerado em TXT. E suponha que no escopo esse requisito seja funcional.

Como o caso de uso “Emitir relatório de presença” (descrito na página 66) nunca foi contado, como exatamente devemos apresentar a contagem para o usuário/cliente?

**Análise/Conclusão:** Estamos falando de uma melhoria em um sistema existente. Neste caso não há a necessidade do sistema a ser mantido ter sido todo medido para que melhoria possa também ser medida. Ter esta medição do sistema é algo que facilita a medição da melhoria, porém não é essencial. Quando não há a medição do sistema que será melhorado, temos que medir toda a parte que será o alvo da manutenção. No exemplo citado, a contagem seria apresentada com a função "Emitir Relatório de Presença" medida do jeito que será entregue ao usuário após a alteração e com a informação de que a função foi alterada. Mas o exemplo de alteração apresentado (mudar o formato de saída do relatório para TXT) não é um bom exemplo de requisito funcional.

**Dúvida:** Supondo que no mesmo exemplo (pag. 69) o requisito seja: “Acrescentar no relatório de presença, o telefone da pessoa”. (suponha que na tabela Pessoa já exista o atributo Telefone).

Neste caso, qual a forma correta de apresentar a contagem da MANUTENÇÃO para o cliente:

**Forma 1:**

Descrição	Tipo	TD	AR/TR	Complexidade	Contribuição
Emitir Relatório de Presença	SE	10	3	Média	5
<b>Total</b>					5

Nesse exemplo, apenas foi aumentado a quantidade de TD de 9 para 10 (novo atributo apresentado no relatório) e apresentado na melhoria apenas a SE.

**Forma 2:**

Descrição	Tipo	TD	AR/TR	Complexidade	Contribuição
Pessoa	AIE	4	1	Baixa	5
Apontamento	ALI	4	1	Baixa	7
Justificativa	ALI	3	1	Baixa	7
Emitir Relatório de Presença	SE	10	3	Média	5
<b>Total</b>					24

Nesse exemplo, além de contar a SE, ainda são apresentados todos os ALIs e AIEs que o relatório utiliza.

A grande dúvida é quando contar ou não contar os AIEs e ALIs. Em todas as telas que acessamos, devemos considerar novamente a contagem deles? Podemos contar apenas uma vez no sistema inteiro? Ou devemos contar só quando alterar dados (*insert, update, delete* na tabela) nesses AIEs e ALIs?

**Análise/Conclusão:** Numa melhoria devemos medir apenas os requisitos que foram modificados, incluídos ou excluídos pelo projeto. No exemplo, apenas o relatório mudou; o arquivo não. Logo apenas ele deve ser medido.

**Assunto:** Processos Elementares distintos de acordo com a Lógica de Processamento 5

**Data:** 02/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3598>

**Dúvida:** Há uma dúvida quanto a Lógica de Processamento de número 5: Condições são analisadas para determinar as aplicáveis.

Segue um exemplo: Um sistema de processo seletivo está sendo desenvolvido. Um dos requisitos do usuário é "Gerenciar Processo Seletivo". Este requisito é basicamente um conjunto de informações de controle que dirá como o processo seletivo funcionará, desde a inscrição dos candidatos até a convocação dos aprovados.

Por questões de negócio, existem 4 possíveis regras de "Limitação da inscrição em vaga ofertada". Estas regras influenciarão na hora que o candidato estiver fazendo sua inscrição, onde ele poderá se inscrever em apenas uma vaga ofertada em todo o processo seletivo, ou poderá se inscrever em múltiplas vagas em destinos diversos, desde que as vagas sejam para uma mesma função, etc.

Dependendo da regra parametrizada para uso no processo seletivo, na tela de inscrição as lógicas utilizadas para disponibilizar vagas ofertadas ou não para o candidato vão ser totalmente diferentes. Podendo mudar não apenas o "filtro" das vagas, mas até mesmo uma separação em blocos distintos de vagas dependendo de algumas condições determinadas no cadastro das funções.

A dúvida é a seguinte:

- Devemos contar 4 transações para "Incluir Inscrição" e "Editar Inscrição", já que são estas duas transações que serão alteradas pela lógica de processamento? Atualmente estão sendo contadas apenas 2 transações, e não 8.

Ainda nesta linha de pensamento e usando o exemplo de sistema dado, a criação / parametrização de um processo seletivo, pode variar / influenciar várias transações.

**Análise/Conclusão:** Veja se a dúvida é essa mesmo:

1. Existem duas funcionalidades: Cadastro e Atualização de parâmetros de sistema. Neste caso, devemos contar dois processos elementares. Esta funcionalidade apenas mantém um ALI que será consultado nas funcionalidades que utilizam esta informação.
2. Inscrição em vagas. Ao que parece, trata-se de apenas uma funcionalidade com regras de negócios mais complexas. O usuário deve ter requerido assim: "Criar uma tela de inscrição de candidatos". Se for este o caso, deve-se contar um processo elementar e incluir o ALI dos parâmetros de sistemas como participante desta funcionalidade.

**Dúvida:** Quanto ao cadastro do parâmetro, não há dúvidas. Quanto ao item 2 citado acima, é mais ou menos isso. O que acontece é que o requisito de Cadastrar/Editar inscrição de candidato tem um funcionamento em uma parte importante e totalmente diferente dependendo dos parâmetros cadastrados em processo seletivo. Neste mesmo sistema temos alguns cálculos que são feitos de uma forma ou de outra dependendo dos parâmetros, porém a ação com o usuário não muda, sendo necessário apenas apertar um botão para o cálculo acontecer.

No caso da inscrição, há uma dúvida, pois a tela funciona de forma diferente. Esse processo elementar em particular foi dividido em 4 abas e em uma delas (Vagas Ofertadas) seu funcionamento pode mudar muito. Quando pensamos na alteração na tela, somos tendenciosos a pensar em mais de um PE. Porém quando pensamos que o cliente pediu o "Manter Inscrição", porém com um funcionamento diferente dependendo de algumas condições, somos tendenciosos a pensar em apenas um PE.

**Assunto:** Quando podemos considerar um Arquivo, como um Arquivo Lógico Interno?

**Data:** 08/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3610>

**Dúvidas:**

1. Um arquivo só é considerado como um ALI quando há Inclusão, Alteração, Exclusão e Consulta?
2. Esse arquivo poderá ser considerado como ALI se não tiver Alteração?

**Análise/Conclusão:**

1. Não.
2. Sim.

**Complemento/Dúvida:** Considere a Aplicação Apl01. Essa aplicação vai receber um arquivo contendo dados expurgados de outra Aplicação, Apl02. Após o processamento deste arquivo os dados serão excluídos definitivamente da Apl02. Dessa forma não será mais possível para Apl02 consultar estes dados.

Considere que estes dados só podem ser excluídos após um tempo (pelo menos 5 anos). E o expurgo foi necessário por que a plataforma em que está Apl02 tem problemas de espaço.

Apl01 manterá esses dados para o usuário consultar e excluir. Sendo que a exclusão deverá seguir regras, ou seja, os dados só poderão ser apagados se atenderem as regras parametrizadas. Considere também que Apl01 está em uma plataforma diferente da Apl02.

Como em Apl01 tem inclusão, consulta e exclusão, um ALI é necessário. Porém, há um entendimento de que por ser tratar de expurgo, a aplicação Apl01 não teria um ALI e sim um AIE. Eis o motivo da dúvida.

**Análise/Conclusão:** Para APL01, qual a relevância dos dados estarem sendo expurgados em outro sistema? APL01 funciona se receber esse mesmo movimento de dados a partir de uma terceira aplicação que nada tem a ver com APL02? Podemos jogar fora APL02 que APL01 funciona a partir de dados digitados em arquivo texto pelo BROffice? Isso é reconhecido pelo usuário e pode ser evidenciado em seus documentos de requisitos? Ou APL01 é na verdade parte integrante de APL02, como se fosse uma nova camada de apresentação e trata-se de uma melhoria em uma aplicação existente?

As premissas quanto aos requisitos devem ser esclarecidas antes de qualquer manifestação objetiva.

**Dúvida:** Há mais dúvidas relacionadas ao assunto. Quando devemos contar os PFs do ALI e quando não podemos contar?

- 1) Em relatórios que não tem alteração de dados (há apenas consulta), devemos contar os ALIs ou apenas SE?
- 2) Supondo que em uma aplicação temos apenas uma tela de manutenção da tabela Cliente. Nas demais telas do sistema, apenas acessamos as informações de Cliente para mostrar nome e demais dados do Cliente. Nesse caso, devemos contar apenas o ALI na tela de manutenção de Cliente, ou em todas as telas que são exibidas as informações de Cliente?
- 3) Supondo que temos que fazer uma manutenção na tela de Cliente (validação de CPF). Porém nessa validação não vamos incluir, excluir ou alterar "atributos" da tabela de Cliente. Na contagem dessa manutenção, devemos contar o ALI de Cliente?

**Análise/Conclusão:**

- 1) A que se refere por "contar"? Devemos contar como um AR ou como um ALI ou como os dois? Em qual tipo de contagem? Quando é dito SE, quer dizer SE ou CE, correto?
- 2) A dúvida é se uma transação que apenas referencie um ALI ou AIE, que não mantenha um ALI, deva ter um Arquivo Referenciado contado mesmo que não haja manutenção associada? A resposta é sim. Devemos contar um AR ao avaliar a complexidade de uma EE/SE/CE para qualquer ALI ou AIE lido (independentemente dele ser mantido no caso de um ALI, mas nesse caso assim como em qualquer outro, um ALI apenas pode ser contado uma única vez como um Arquivo Referenciado ao avaliar a complexidade de uma transação).
- 3) Em um projeto de melhoria, apenas as funções incluídas, alteradas ou excluídas devem ser incluídas no escopo da contagem. No caso em tela, apenas as transações cuja lógica de processamento foi alterada entram no escopo da contagem. O ALI/AIE apenas entra se essa for a primeira vez que está sendo referenciado pela aplicação.

**Dúvida:** Então:

- 1) Em um relatório vamos contar o ALI, mesmo que não esteja alterando dados no banco de dados. Ex: Em um NOVO Relatório de Cliente, vamos contar um ALI de Cliente (supondo que esse relatório só acesse a tabela Cliente) e uma SE. Correto?
- 2) Vamos contar o ALI Cliente, sempre que tivermos um Caso de Uso que utiliza a tabela Cliente. Certo? Exemplo: Caso de Uso "Emissão de Nota Fiscal", no qual a NF apresenta os dados do Cliente (nome, CNPJ, etc).
- 3) Só vamos contar novamente o ALI Cliente em uma manutenção, se essa manutenção for incluir, excluir ou alterar algum atributo da tabela Cliente. Certo? Ex: No Caso de Uso manter Cliente, incluir um novo atributo na tela, por exemplo, Nome Fantasia (esse atributo não existia na tabela, mas está sendo criado nessa manutenção).

**Análise/Conclusão:**

- 1) Não. Não devemos criar um modelo mental de CONTA ou NÃO CONTA. Devemos criar um modelo mental de CONTA ou NÃO CONTA COMO. Sem essa qualificação a análise fica sujeita a ruídos e as perguntas ficam sem respostas objetivas. No cenário apresentado, uma nova

SE ou CE será incluída no escopo da contagem. Na avaliação da complexidade desta função, o arquivo com os dados de cliente conta como um Arquivo Referenciado. Para que esse arquivo seja incluído no escopo, ele - o arquivo - deve estar sendo incluído, alterado ou excluído na aplicação; o que não parece ser o caso. Adicionalmente, para que ele seja classificado com um ALI, algum processo da aplicação em análise deve necessariamente mantê-lo, caso contrário, ele é classificado como um AIE.

- 2) Sim, conte o ALI Cliente como um Arquivo Referenciado sempre que uma transação o referenciar ou manter.
- 3) Sim.

**Assunto:** Unicidade do Processo Elementar

**Data:** 10/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3617>

**Dúvida:** Considere a seguinte questão: Quando um usuário adiciona um registro de Cliente na Aplicação A, uma validação de CEP é feita, enviando o código do CEP para uma Aplicação B. Lendo o dado de CEP enviado, a aplicação B valida o código e envia uma mensagem de sucesso ou erro de volta para a aplicação A. Se a mensagem for de sucesso, a aplicação A insere o registro de Cliente. Se a mensagem recebida for de erro, a Aplicação A não insere o registro de Cliente, informa a mensagem de erro ao usuário. Quais são as funções a ser contadas para a Aplicação A?

- A) Uma EE para adicionar o registro de Cliente, um ALI para os dados de Cliente e uma CE para a validação do CEP.
- B) Uma EE para adicionar o registro de Cliente, um ALI para os dados de Cliente e uma AIE para a função de dados do CEP.

Penso que a letra A é a correta. Contudo, o simulado sinaliza a letra B como a correta.

**Análise/Conclusão:** A opção A não pode ser correta, pois a contagem de "uma CE para a validação do CEP" é incorreta, pois a validação do CEP faz parte do processo elementar "Adicionar Cliente". Por isso a opção correta é a B.

**Assunto:** Como contar relatórios com *layouts* únicos, porém com validações distintas?

**Data:** 11/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3627>

**Dúvidas:**

- 1) Considere o seguinte cenário: É solicitado o envio de um arquivo a outro sistema. Este arquivo possui um *layout* único. Entretanto, há um indicativo que informa seu real propósito: (1) Status e (2) Pagamento. Poderíamos separar os 2 arquivos, já que entre eles não há nenhuma co-relação. Entretanto isto não é feito por opção do cliente. Como devemos contar? 1 SE ou 2 SE's ?

Obs.: Todas as validações são feitas de forma independente, e cada modalidade possui suas próprias regras.

- 2) Agora o outro cenário, que por sinal é muito parecido com o primeiro: Um sistema recebe um arquivo de adesão/cancelamento (EE) de um outro sistema. As validações novamente são diferentes, afinal os objetivos são diferentes. Aos olhos do usuário, não é visto como um arquivo e sim como duas funcionalidades distintas (Adesão e Cancelamento). Devemos contar as duas ou apenas uma funcionalidade?

**Análise/Conclusão:** Como a lógica de validação é diferente para status e pagamento, devemos contar 2 saídas.

**Complemento:** A resposta está nos requisitos do usuário: podemos gerar uma saída em separado da outra? A regra de unicidade na 4.3.1 deixa tudo muito mais claro.

**Dúvida:** Para confirmar o entendimento, temos: 1 arquivo com 2 validações distintas, podendo ser gerados de forma independentes, consequentemente gerando 2 ou mais SE's / CE's.

**Análise/Conclusão:** Veja que o PODE ser gerado separadamente, não se refere a um caso de execução em que haja apenas um e não o outro tipo, mas sim o usuário tem o poder de especificar a geração de apenas um tipo, mesmo que haja outros pendentes de geração. Verifique se no plano das regras que governam o negócio em análise há algum rito específico para a geração de um tipo e outro rito específico para a geração do tipo caso; verifique se existe alguma restrição nesse plano que impeça que sejam gerados juntos e, só então, devemos considerar como dois processos únicos.

A decisão quanto a isso não pode estar nas mãos do desenvolvedor, deve estar nas mãos daquele a quem ele serve.

**Assunto:** Tratamento de Não Conformidade pela APF

**Data:** 14/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3643>

**Dúvida:** Considerando que os métodos de desenvolvimento incrementais ou iterativos indicam a entrega de *software* executáveis em 10/14 dias e que um ciclo/iteração/*sprint*, conforme o método, inclui parte da correção dos defeitos do ciclo anterior; como as empresas especialistas em medição estão tratando o *software* executável que parte está conforme e outra parte não está conforme (contém defeitos)?

Existe uma unidade menor de medição do que a função? Poderia ser uma operação ou item semelhante (por exemplo: inclusão/exclusão de um cadastro) que possa delimitar a parte conforme da não conforme e assim contar os Pontos de Função, diminuindo os Pontos de Função da parte não conforme?

**Análise/Conclusão:** No geral a taxa de entrega já deve incluir o rateio do esforço de todo o ciclo de vida do projeto/iteração, e isso é baseado em dados históricos dos projetos. Por exemplo, seja uma taxa de entrega de 10h/PF. Dentro dessas 10 horas, temos rateados os esforços para detalhamento de arquitetura, planejamento de testes, implementação, execução de testes e inclusive a correção desses defeitos (entre outras atividades, dependendo do seu ciclo de vida). Espera-se que dentro de um horizonte estatístico de projetos executados, o percentual de esforço dedicado à correção de defeitos seja razoavelmente projetável. Isso ocorre da mesma forma para as outras atividades do projeto.

O SISP elaborou um método para mensurar o tamanho de manutenções corretivas para sistemas que já estão em produção. Ele considera as seguintes fórmulas:

$$PF = PF\_ALTERADO * 0,6$$

$$PF = PF\_ALTERADO * 0,7$$

$$PF = PF\_ALTERADO * 0,8$$

Onde PF\_ALTERADO indica o tamanho funcional das funções impactadas. No primeiro caso, o método considera que o sistema que deve sofrer a manutenção corretiva possui documentação completa e atualizada. O segundo caso considera ausência de documentação ou documentação desatualizada e, a implementação deve apenas gerar os *builds* de correção, sem qualquer redocumentação. O terceiro caso também considera ausência de documentação ou documentação desatualizada. Entretanto, a implementação deve incluir a redocumentação da funcionalidade.

O intuito do fator utilizado é justamente penalizar as correções com menor documentação, que acabam exigindo um esforço maior de correção. Esse fator é estimado, e pode ser ajustado conforme as especificidades de cada ambiente.

Obs.: O Roteiro SISP sugere o uso de percentuais. No entanto, estes devem ser aferidos com base em dados históricos de projetos concluídos, como o próprio Roteiro SISP recomenda: "Os percentuais de multiplicação são estimados, podendo ser reajustados conforme avaliação da base histórica dos serviços realizados no órgão ou entidade"

**Complemento/Análise:**

1. Projetos desse tipo têm um maior risco de retrabalho quanto menores são as interações. Ao medir o valor agregado de uma iteração, devemos considerar uma contingência para isso, digamos 30%.
2. Não devemos realizar o acompanhamento, pagamento ou gestão com bases nas fases de iniciação, elaboração, construção e transição. Devemos considerar uma iteração zero referente à iniciação. O principal produto dessa fase zero deve ser o plano de iteração. Ao final de cada iteração devemos determinar o valor agregado pela mesma em termos das disciplinas pelas quais uma função passou naquela iteração; considerando os artefatos aceitos (ainda que provisoriamente).
3. Se houver retrabalho durante uma iteração n+1 de entregáveis de uma iteração n, isso não gera nova medição em termos de funções incluídas, alteradas ou excluídas. Devemos considerar uma única planilha com as funções da aplicação que é versionada a cada iteração. Ao final do projeto, uma mesma função nunca pode ser medida duas vezes. O que pode acontecer é que numa iteração uma função era de complexidade baixa e na seguinte passou a ser de complexidade alta.
4. A conta do valor a pagar pela interação deve ser feita por diferença. Devemos calcular o valor agregado total do projeto até o final da interação e subtrair do valor agregado total ao final da interação anterior.
5. É possível que haja iterações sem valor agregado em termos de funcionalidades ou mesmo que diminuam esse valor agregado.
6. Devemos criar mecanismos de comparar o valor agregado em termos percentuais com a evolução do prazo também em termos percentuais.

**Assunto:** Manutenção em ALI: é uma alteração nos dados ou na estrutura do ALI?

**Data:** 19/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3651>

**Dúvida:** Quando é dito que em uma manutenção devemos apenas contar o ALI se tiver alteração, inclusão ou exclusão nesse ALI, isso significa alteração nos dados do ALI ou na estrutura do ALI?

Exemplo:

1. Uma aplicação tem um ALI Cliente que foi contado na tela de manutenção de cliente com 4 TDs (CPF, NOME, SEXO e TELEFONE).
2. Essa aplicação também tem um relatório de Clientes que mostra na tela o CPF, NOME e SEXO de todos os clientes da base.

Caso 1: Será feita uma manutenção no relatório de Clientes, onde será necessário incluir um totalizador no final do relatório pra contar quantos clientes o relatório traz.

Caso2: Será feita uma manutenção no relatório de Clientes, onde será necessário acrescentar uma coluna para telefone.

Pelo o que foi dito:

No caso 1, devemos contar uma SE com 3 TDs, ou seja não alterou em nada a contagem de quando o relatório foi feito, correto? Ou o campo calculado também entra como um TD?

No caso 2, como ficaria? Será contado apenas uma SE com 4 TDs, ou devemos considerar também um ALI com 4 TDs, já que o SQL foi alterado para acrescentar o campo TELEFONE e trazer no relatório?

**Análise/Conclusão:** No primeiro caso deve-se contar o campo de total (campo derivado) como 1 TD, ficando assim, 4TD's. No segundo caso, é uma função de transação e por isso será uma SE, se tiver cálculos e campos derivados ou alterar o ALI. Neste caso, é apenas uma consulta ao ALI. Portanto é uma CE com 4 TD's. Se for apenas um relatório vai ser um SE com 5 TD's.

**Assunto:** Mudança de cor de um campo e mensagens enviadas ao usuário: devem ser contados como Tipos de Dados ou são Requisitos de Qualidade?

**Data:** 19/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3654>

**Dúvida:** Duas dúvidas em relação à DETs:

1. O CPM diz que para mensagens enviadas ao usuário, o DET deverá ser contado se a mensagem for de Erro, Sucesso ou Confirmação de prosseguimento da transação. Uma mensagem (solicitada pelo usuário) do tipo "Pesquisando..." após o acionamento de um botão de pesquisa pode ser contado como DET?
2. A mudança de cor de um campo numérico, exibido para o usuário, que varia de acordo com o sinal do número (vermelho para números negativos e branco para números positivos) pode ser considerado como um DET?

**Análise/Conclusão:**

1. Quanto à primeira dúvida, a resposta é sim
2. Quanto à segunda dúvida, a resposta é não. O sinal é indissociável do número.

**Dúvida:** A mensagem "Pesquisando..." não seria considerada um componente de *status*, similar a uma barra de processamento? E neste caso, não deveria ser considerado um requisito de qualidade?

**Análise/Conclusão:** Não. A capacidade do sistema de informar o sucesso em uma operação, solicitar uma confirmação ou enviar uma mensagem de erro, é contado como um TD. O "Pesquisando..." em tela cumpre a função de informar que os parâmetros informados foram aceitos com sucesso e que nenhum erro aconteceu.

**Assunto:** Contagem de cadastro dividido em várias telas

**Data:** 20/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3657>

**Dúvida:** Há um sistema que contém um módulo de contratos onde se pode cadastrar o contrato que é composto de vários grupos de informações: Dados básicos do contrato, datas e prazos, participantes, itens do contrato, programação financeira do contrato. Cada grupo desses é uma tela separadamente. Podemos acessar essas telas por meio da tela de cadastro de dados básicos como também podemos ir direto para a tela informando no filtro o campo número do contrato. Sempre que cadastramos um contrato ou parte dos dados de um contrato ele fica com o status "incompleto", até que enviamos o contrato para aprovação pelo gestor.

Uma visão é que todas essas telas fazem parte de um mesmo processo elementar: Cadastrar contrato. Isso porque o fato de o contrato só poder ser enviado para aprovação após preencher todos os dados significa que o contrato não tem sentido para o negócio se tiver só os dados básicos cadastrados. Concordam?

A maior dúvida é sobre como tratar a alteração. Se alterarmos alguns grupos de dados específicos, o contrato volta para o status de "incompleto" e precisará ser aprovado novamente. Outros grupos de dados não. A alteração é feita sem afetar o status do contrato. Sem contar que tem listas dentro desses grupos de dados que podem ser excluídas. Ex: Na tela de datas são incluídas várias datas significantes para o contrato. Essas datas vão sendo exibidas em uma lista de datas. É possível excluir, alterar e incluir uma nova data.

Como devemos tratar isso? A alteração de datas e prazos é um processo elementar ou faz parte do processo elementar alterar contrato? Como devemos contar as inclusões, exclusões e alterações de datas que existem tanto na tela de cadastrar quanto na tela de alterar?

**Análise/Conclusão:** Devemos nos abstrair da quantidade de telas e focarmos nos diferentes atores, desempenhando diferentes papéis em diferentes processos de negócio. O controle de acesso (alinhado ao interesse e visão do usuário) é um bom local para se procurar por evidências disso. Quem pode alterar alguma informação, que quando alterada implica em haver nova aprovação é a mesma pessoa que pode alterar alguma informação que quando alterada implica em não haver nova aprovação? Na documentação de processos de negócio existem processos separados para um tipo de alteração e para o outro tipo? Esses são os lugares que devem ser vistos na nossa análise na busca pelos processos elementares ao identificar as funções de transação.



**Assunto:** Como contar o envio de mensagens automáticas e rotinas executadas automaticamente?

**Data:** 25/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3664>

**Dúvida:** O cliente solicitou que fosse implementado um conjunto de funcionalidades em um sistema que pode ser resumido como um "sistema de e-mail interno simplificado". Existem várias transações, como listar e-mails, ler e-mail, escrever e-mail, etc. Mas o cliente também solicitou que o sistema enviasse algumas mensagens automáticas. O envio dessas mensagens possui uma regra de negócio própria.

Mesmo a lógica de processamento sendo diferente por mensagem enviada, podemos entendê-la como sendo uma única transação de envio de mensagem automática. Entretanto, há uma dúvida quanto à forma de contagem dessa transação. Como é uma mensagem automática, não há informação atravessando a fronteira. O comando para envio da mensagem é interno, e provavelmente será programado para ser executado diariamente em um certo horário.

Outra dúvida: como devemos contar rotinas executadas automaticamente, por exemplo, em bancos? Existem aquelas rotinas que rodam de madrugada para baixar pagamento e outras várias rotinas. Nesse caso, como elas rodam de forma automática, e não havendo informação atravessando a fronteira da aplicação contada, como devemos contá-las?

Poderíamos considerar o caso de que o que executa o gatilho de verificação e envio dessa mensagem pudesse ser uma aplicação externa (por exemplo, o TOMCAT), mas seria bem possível ser implementado dentro da própria aplicação. Assim, isso seria uma condição técnica, o que não deveria ser levada em conta.

#### **Análise/Conclusão:**

- 1) Podemos considerar que a mensagem atravessa a fronteira sim, pois em algum momento um usuário irá ler este e-mail, correto?
- 2) Alguns sistemas, por serem basicamente *batch*, ficam difíceis de serem dimensionados através de PF. Devemos tentar ao máximo identificar as fronteiras por sistema: Exemplo: Sistema de Conta Corrente é uma fronteira, Sistema de Conta Poupança outra, Sistema de Pagamentos outra, e assim por diante. Outra dica é considerar o sistema de agendamentos (exemplo OPCON) como um usuário, por que é ele quem sempre irá startar as rotinas conforme horário programado.

#### **Dúvida/ Complemento:**

- 1) Todo usuário pode receber mensagens e pode lê-las. Já existe a transação de leitura de mensagem, independente de quem enviou, ou seja, a transação "Mensagem - Ler" é capaz de ler qualquer mensagem, seja ela enviada por um usuário ou pelo próprio sistema. Será necessário implementarmos o envio automático das mensagens e será uma transação diferente, mas a dúvida persiste em como contá-la.
- 2) No caso de sistemas de bancos, a fronteira realmente vai ser determinante, porém pensando em outros sistemas que simplesmente possuem um *batch*, por exemplo, que é rodado a noite para que sejam desativados alguns registros dependendo de alguma RN específica. Neste caso definitivamente não há informações atravessando a fronteira, portanto este caso continua se parecendo com o de mensagens citado anteriormente.

O arquiteto do projeto foi consultado sobre quem executa o gatilho para rotinas automáticas, e no caso citado é um componente interno do Grails que faz isso. Considerar por exemplo o OPCON ou então algo interno pode ser visto como uma solução técnica. Portanto é necessário contar a rotina em si. Uma exceção é quando, por exemplo, o cliente já possui um sistema que vai comunicar com o que está sendo desenvolvido e ele deixa claro que tem que ser assim. Dessa forma, o sistema existente será um usuário da aplicação contada e atravessará pela fronteira o comando de execução da rotina

#### **Análise/Conclusão:**

- 1) Deve ser contado como uma Consulta Externa ou como uma Saída Externa, dependendo da lógica de processamento (se possui algum cálculo, ou altera o comportamento do sistema).
- 2) Deve ser contado como uma Entrada Externa, considerando que o sistema de agendamentos é um usuário que passa parâmetros de entrada (um xml ou um arquivo txt) para rotina executar um

processamento que será mantido em banco (atualizará um ALI ou altera o comportamento do sistema).

**Dúvida/Complemento:** No 1º caso, não é um e-mail que é enviado. É uma mensagem interna do sistema. Neste caso o sistema mantém um ALI criando a mensagem e relacionando-a com o(s) destinatário(s). A dúvida é se não seria uma EE, já que a leitura da mensagem será feita pela transação de leitura.

Esse envio de mensagens automáticas acaba sendo uma rotina como outra qualquer que é executada de forma programada. Sendo assim, os 2 casos poderiam ser tratados da mesma forma, já que o que eles fazem é alimentar ALIs, não exibindo nada.

A dúvida é o fato de que esse agendamento e acionamento da rotina podem ser feitos tanto por um aplicativo externo quanto algo interno, e desta forma não há informações atravessando a fronteira.

Existe algo quanto a isso no CPM ou já houve discussões sobre isso no IFPUG?

**Assunto:** Contagem de *Upload* de Arquivo

**Data:** 29/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3675>

**Dúvida:** Como devemos contar *upload* de Arquivo?

**Análise/ Conclusão:** Conte um "*upload*" de arquivo como uma EE se, e somente se, ela for a implementação de um processo de negócio completo e reconhecido pelo usuário; também dissociado de qualquer outra atualização.

Não conte um "*upload*" como uma EE quando ela está em separado da atualização de outros dados como resultado do projeto; porque dificulta isso ser feito em conjunto com a atualização de outros campos ou fica mais bonito desse jeito.

Devemos pensar no conteúdo do arquivo processado como um campo e fazer a avaliação nos mesmos moldes que fazemos com qualquer campo. Não se conta cada "Alterar um campo" como uma EE.

Obs.: Quando esse arquivo para "*upload*" for a implementação de várias entradas externas, devemos contar dessa forma e considerar não o conteúdo do arquivo como um TD, mas cada menor unidade de informação individualmente reconhecida no(s) processamento(s) de entrada ali implementados.

**Assunto:** Determinação da fronteira na contagem de Projetos SOA

**Data:** 29/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3679>

**Dúvida:** Hoje em dia é comum em projetos SOA termos que trocar informações entre sistemas.

Pois bem, estudemos o caso: O sistema A deve atualizar no sistema B um determinado conjunto de tabelas. O sistema B deve retornar a confirmação da transação e alterar o *status* da operação em A.

Consideremos serem sistemas de plataformas diferentes, e assim, precisamos de um barramento (broker) que faça a comutação entre os sistemas para a troca desses dados.

Pergunta: O barramento como um todo (suas funções e tabelas) deve ser contado como um AIE? Se não, como devem ser contadas as funções setadas no barramento?

**Análise/Conclusão:** Sugerimos que leia o artigo do IFPUG que aborda a contagem de *Middleware*. Há uma versão em Português. *Function Points & Counting Middleware Software Applications – Portuguese*

Alerta: o artigo está disponível apenas para os filiados.

Além disso, há uma *thread* nesse grupo: <http://br.groups.yahoo.com/group/livro-apf/message/2568>

**Complemento:** Para aplicar a Análise de Pontos de Função, a diferença em utilizar SOA ou uma arquitetura convencional está basicamente na divisão das fronteiras (em SOA temos sistemas especialistas e não conglomerados de negócio - como em um ERP, por exemplo). Na prática, devemos contar um AIE num cenário ou no outro, somente quando houver necessidade de validação ou referência de uma

informação específica na perspectiva do negócio do sistema sendo contado, ou seja, a regra não muda, nem precisa de adequação.

Para facilitar a contagem de sistemas SOA, devemos imaginar que cada sistema é uma pessoa com uma "profissão", a qual pode ser entendida como um conjunto de serviços prestados a outros "profissionais" da "rede corporativa" (leia-se sistemas). Devemos imaginar estes especialistas trocando informações para a execução dos serviços e veremos que os serviços são praticamente os processos elementares de cada aplicação sendo analisada.

Devemos tomar cuidado com as fronteiras técnicas envolvidas, as quais normalmente são consideradas componentes visando o reuso de funções.

**Assunto:** Contagem da manutenção de cadastros que só possuem Código e Descrição

**Data:** 29/04/2011

**Link:** <http://br.groups.yahoo.com/group/livro-apf/message/3683>

**Dúvida:** Os cadastros que só possuem código e descrição, mantidos pelo usuário (ou seja, o usuário faz o cadastro através do sistema), são contados como ALI's ou são contados de alguma outra forma?

**Análise/Conclusão:** Dados de substituição, que cumprem o papel de substituir uma descrição textual longa por um código, são classificados como Dados de Código no jargão do IFPUG e devem ser desconsiderados na análise como ALI; assim como as transações existem exclusivamente para a sua manutenção ou consulta devem ser desconsideradas como EE, SE ou CE.

Existem organizações que usam regras complementares aos padrões internacionais definidos nos sentido de que tais transações/arquivos sejam pontuadas em (equivocados) pontos de função. Por exemplo, cada transação de inclusão de dados em uma tabela como essa seria pontuada com 0,3 "pontos de função".

Críticas:

1. Do nome: Ao invés de usar o nome Pontos de Função, deveria usar o nome "Unidade de Medição de Serviço";
2. Da utilização: O princípio da APF é medir aquilo que agrega valor ao negócio em termos da funcionalidade de negócio fornecida pelo *software* sendo medido. Ao medir uma transação que existe para tornar mais fácil a manutenção do sistema, economizar espaço em disco, tornar mais fácil a utilização do sistema pela presença de listas suspensas, etc; desviamos a lógica da análise de pontos de função de uma perspectiva funcional para uma perspectiva não funcional. Caso em que pode ser utilizada: Quando o objeto da manutenção é exclusivamente esse - criar e manter dados de código dissociados de qualquer funcionalidade do usuário.
3. Do poder: Na Análise de Pontos de Função o poder de "parir" funções está no domínio do usuário. Essas tabelas são resultados de projeto em poder do desenvolvedor. Quando os requisitos do usuário estabelecem esse tipo de Dado de Código no domínio funcional, cabe um olhar crítico sobre esses requisitos.