

Síntese das discussões do fórum Livro-APF: Dezembro/2012

Nessa síntese foram abordados, em 149 mensagens, os seguintes assuntos:

- Como contar "Incluir Produto" e "Alterar Produto"
- Instalador de Aplicação
- Contagem Implícita de CE e SE
- Manutenção Corretiva e Garantia em Contratos por PF
- Como medir jogos de computador?
- Projeto de mudança de plataforma
- Aplicação das Regras
- AIE
- Fórmula do calculo dos pontos de Função
- APF Para projetos OO
- Certificação CFPP
- Posicionamento da Fronteira
- Recomendação de versão do Manual de Práticas de Contagem para Desenvolvimento de Sistema
- Contagem de Componentes acessando dados de outra aplicação
- Contagem de relatório para diferentes perfis
- Contratação de serviços de medição

Assunto: Como contar "Incluir Produto" e "Alterar Produto"

Data: 03/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5782>

Dúvida: Como contar a seguinte situação: As funções Incluir e Alterar Produto possuem os TDs "Nome do Produto", "Total do Estoque", "Valor" e "Seção", e o AR "Produto".

Deve-se contar apenas uma função por serem os mesmos TDs e ARs ou duas funções por serem duas tarefas diferentes do usuário?

Análise: Geralmente, as funcionalidades de alterar e incluir vão ter os mesmos TDs e ARs, e o que irá diferencia-los é a lógica de processamento. Logo, conta-se duas funções de transação (EE), desde que as duas "tarefas" estejam especificadas pelo usuário.

Resumindo, quando dois processos elementares são comparados e se determina que eles contêm diferentes DERs, ALRs ou Lógica de Processamento, eles são identificados como processos elementares separados se forem especificados como requisitos funcionais distintos pelo usuário. A condição implica que basta que apenas DERs, ALRs ou Lógicas de Processamento sejam diferentes, para ser considerada a existência de processos elementares distintos (Na maioria das vezes!). Entretanto, podem ocorrer algumas pequenas variações que não caracterizam processos elementares diferentes.

Assunto: Instalador de Aplicação

Data: 03/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5785>

Dúvida: A elaboração de um instalador de uma aplicação pode ser considerada na contagem de APF?

Análise: Qual a classificação do Instalador?

Análise/Complemento: Pode ser considerada se o propósito da contagem for medir o desenvolvimento de um instalador, o que não costuma ser o caso.

Análise/Complemento: Talvez este seja um requisito não funcional relacionado à facilidade de instalação do software. Consulte no CPM esta característica junto às características gerais (Parte 5, página C-23).

Assunto: Contagem Implícita de CE e SE

Data: 30/11/2012

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5780>

Dúvida: Temos, por exemplo, 5 telas:

- 1 - Manter funcionário - com uma CE de funcionário
- 2 - Enviar pedido – com um drop-down com os funcionários do setor de vendas
- 3 - Cadastrar produto – com um drop-down com todos funcionários
- 4 - Solicitar produto – com um drop-down com todos os funcionários

Conta-se 2 CEs e para cada tela um TD?

Análise: O usuário/negócio requer a capacidade de visualizar essas listas? Os dados são recuperados a partir de algum ALI? Deve-se tomar cuidado para verificar de onde os dados estão sendo recuperados, se não são de uma entidade de dados de código. Quando nos requisitos do usuário está claramente explícito que o "combo" irá realizar uma recuperação de dados de uma certa entidade (que é um ALI), deve-se contar.

Análise/Complemento: No caso mencionado, o usuário tem a necessidade da consulta da lista. Os dados vêm de um ALI.

Nesse cenário, seria um filtro diferente para cada situação.

Assunto: Manutenção Corretiva e Garantia em Contratos por PF

Data: 03/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5791>

Dúvida: Em um contrato por PF, paga-se pelo que se recebe, desde que seja de acordo com o solicitado. Isto vale para qualquer relação comercial, não só para contratos por PF.

Se o cliente demanda um projeto em PFs e eventualmente detecta defeitos no projeto entregue, **não cabe remuneração alguma ao fornecedor para executar as correções**, pois o cliente pagou para ter as funcionalidades funcionando (desculpem o pleonasma). Este princípio também vale para qualquer relação comercial e encontra-se consagrado no código de defesa do consumidor. Quem compra um carro zero quilômetro cuja garantia é de 1, 3 ou até 6 anos dependendo do fabricante, sabe que na eventualidade de um defeito surgir neste período, a fábrica obriga-se ao reparo sem custo algum ao cliente.

Voltando ao caso do software, qual deveria ser o período de garantia? Sabemos que software não se desgasta, enferruja ou envelhece. Qual o sentido então em se estabelecer um prazo de garantia inferior à duração do contrato com o fornecedor? Sinceramente não vejo nenhum sentido, pensando como cliente. No entanto vejo frequentemente a seguinte questão aparecendo em contratos de governo:

*"Os produtos e serviços decorrentes da presente contratação deverão ter a **garantia de 06 (seis) meses**, contados a partir da emissão do termo de recebimento definitivo referente a cada ordem de serviço emitida, dentro dos quais a Contratada deverá responsabilizar-se por correções de eventuais defeitos ou er-*

los encontrados em qualquer tipo de serviço, artefato ou produto entregue, sem ônus para a contratante, mesmo aqueles já considerados aceitos."

Ou seja, enquanto tem fabricante de carro que dá 6 anos de garantia, alguns (muitos) órgãos do governo exigem apenas 6 meses de garantia dos serviços prestados pela fábrica de software. Findo este prazo, qualquer defeito que eventualmente surja (e eles aparecem) será remunerado para o fornecedor com o suado dinheiro pago com nossos impostos. E o fornecedor é premiado com a remuneração para correção de falhas de qualidade na sua entrega original.

Então, como cidadão, peço aos colegas que estão no serviço público e estejam em condição de influenciar o processo de contratação de suas empresas, que atentem-se nesta questão da garantia do serviço. Ela deveria se estender por todo o período do contrato.

Quando há situações de manutenções corretivas em sistemas que não foram desenvolvidos pelo fornecedor, mas que este assumiu a manutenção, daí não cabe o raciocínio que citei antes. Quando o cliente mete a mão no código para dar qualquer manutenção no sistema por conta própria, ele está quebrando a garantia do fornecedor e também não cabe o que foi falado.

Análise: Será que não tem algo a ver com a co-responsabilidade do cliente que homologa um sistema com erro? Imagine que a garantia eterna só valeria para erros indetectáveis em tempo de homologação.

Análise/Complemento: Neste raciocínio ganha quem consegue deixar a sujeira bem escondida debaixo do tapete. Não vejo o cliente como co-responsável. Ele só homologa o software porque sabe que vai achar problemas. Não fosse esse o caso, implantava direto. Mas ele paga pelo serviço sendo entregue correto, só que não recebe isso.

Quando se compra um carro, mesmo que você faça antes um test drive e não identifique problemas, a concessionária não está isenta de garantia porque você não relatou problemas no test drive.

Observe que não trata-se de garantia eterna, mas do prazo do contrato. No governo isso representa 5 ou 6 anos.

Análise/Complemento: O X da questão está no tipo de defeito encontrado, sendo cada caso um caso. Se você comprar um carro 0 km e voltar na concessionária poucas semanas depois porque queimou uma lâmpada, a garantia não cobre peças de desgaste, embora o dono de uma concessionária possa decidir não criar caso para não perder o cliente.

Agora imagine um sistema entregue com seus requisitos funcionais bem claros e devidamente testados e homologados. Aí, num determinado momento, o sistema começa a abendar. O cliente demanda uma manutenção corretiva para que o sistema seja corrigido e volte a funcionar como antes. A equipe de desenvolvimento detecta que o programa está abendando porque houve estouro dos limites de uma tabela. Ex: Sistema de Upload de Arquivos funciona adequadamente até que um dia os limites de quantidade de arquivos por diretório ou ainda tamanho máximo de cada arquivo do Sistema de Arquivos (FAT32, NTFS etc) estoura. Do ponto de vista do cliente, é uma manutenção corretiva, pois não há mudanças nos requisitos que estão claros, só que ninguém imaginou (isto é mais comum do que se possa imaginar) que um dia os limites de uma linguagem, ou do sistema de arquivos, ou de uma determinada arquitetura pudessem ser atingidos.

O fornecedor deve trabalhar de graça pra corrigir este problema ou não se trata de manutenção corretiva?

Análise/Complemento: O caso descrito não é um defeito. O sistema foi projetado para operar em um cenário, o tempo passou, o cenário mudou e não houve manutenção preventiva, ocasionando a falha, que não é defeito. Defeito é o funcionamento distinto do especificado. Se não há especificação do que é o correto o cliente não tem como falar em defeito. Se eu compro um carro zero, com garantia de 3 anos e não troco o óleo a cada 6 meses, se o motor fundir com 2 anos, não cabe reclamar garantia.

Análise/Complemento: Se a solução fica obsoleta é porque o contexto mudou. Ou seja, os requisitos teriam que ser alterados para que a solução adapte-se a este contexto. Isto não é manutenção corretiva.

Um cliente que contrata desenvolvimento de software obrigatoriamente tem que ter algum conhecimento técnico, não para contar Pfs, mas para definir o que deseja dos requisitos não funcionais.

Análise/Complemento: Ainda no caso descrito acima, apesar de afetar a aplicação, não diz respeito aos requisitos implementados nela, mas na infraestrutura necessária para rodar o software. Colocar o fornecedor para resolver esse problema é transferir para outrem uma responsabilidade que é do órgão.

Penso que a garantia deve se ater apenas ao software implementado, uma vez que a infraestrutura em que ela vai operar não é de responsabilidade do fornecedor.

Concordo de que a garantia deve se estender até o fim do contrato. Porém, isso traz um problema: se o erro tiver sido introduzido na aplicação já no fim do contrato, essa cláusula de garantia será inócua, pois provavelmente o contrato estará terminado quando o tal erro aparecer. E a probabilidade de esse cenário acontecer é grande, pois os fornecedores costumam desmobilizar suas equipes ao se aproximarem do término de um contrato que não será renovado (por decisão própria, do órgão ou por ter atingido os 5 anos que a lei faculta), o que sobrecarrega os profissionais que ficam no cliente e aumenta o risco da introdução de erros na aplicação.

Talvez uma saída para resolver essa questão fosse vincular a garantia a cada ordem de serviço entregue. Dessa forma, mesmo que o contrato tivesse terminado, os últimos serviços ainda permaneceriam cobertos com a garantia.

Análise/Complemento: Para que realmente exista a “garantia” do automóvel, torna-se necessário o estabelecimento de várias regras indicadas no manual no proprietário, e geralmente em letras tão pequenas que nunca são lidas. Um ponto que desabona a garantia é o uso de combustível “irregular” ou de má qualidade!

Se tratando de software, alguém arriscaria citar algum ponto que poderia afetar a “garantia” apontada como injusta?

Análise/Complemento: O vício oculto é previsto em lei e tem suas implicações que vão além do prazo da garantia, e, se o contratado colocar muito oculto, ainda pode ser caracterizada a má-fé.

Análise/Complemento: Confeção de software é algo absurdamente complexo e o trabalho de quem o desenvolve é exatamente ocultar essa complexidade do usuário. Não vejo necessariamente um defeito como uma sujeira varrida para debaixo do tapete.

Há defeitos que ficam latentes por décadas sem ninguém perceber e há defeitos cuja própria caracterização como tal é objeto de discussão.

É lamentável o desenvolvimento das coisas no sentido da busca por um culpado para pagar por um dano muitas vezes difuso. Almejar software com zero defeito é algo que implica em um custo que se paga? Se paga no patamar de R\$ 300,00 ou R\$ 500,00 por ponto de função?

Não existe almoço grátis e não se pode comparar software a carro nesse contexto; afinal, para gozar dos seis anos de garantia todas as revisões tem que ser feitas na concessionária, no preço que o fornecedor quiser cobrar por elas.

Se um problema não se manifestou em seis meses não cabe onerar que entregou o software. Remover 80% dos problemas não tem o mesmo custo de remover os 20% restantes. Esses 20% restantes custam mais.

Análise/Complemento: O problema chama-se Gestão de Contratos.

Ninguém tem que trabalhar de graça e há corresponsabilidade, sim, do cliente, no processo de desenvolvimento e manutenção de software. A inteligência do negócio é dele.

Em contrapartida, qual é problema de se ter a garantia do sistema até o término do contrato? Não há problema nenhum em se querer um carro com 6 anos de garantia, ao invés de um carro com 1 ano de garantia. O problema é o preço disto, e fornecedor caro o Governo não contrata.

Análise/Complemento: Temos que dividir o problema. A homologação feita pelo cliente - reunião de validação do SCRUM - é indispensável, e tem como objetivo analisar um software, um produto considerado pronto pelo time, avaliando questões de usabilidade e atendimento às regras de negócio (requisitos funcionais), que já devem ter sido testadas anteriormente.

Porém, como o processo de gestão de conhecimento (passagem de tácito para explícito) não é perfeito, sempre há o que ser validado. A questão é o que está combinado como pronto. Pronto deve ser software funcionando, atendendo os requisitos funcionais e não funcionais explícitos e aqueles que são default da instalação. Assim, não é responsabilidade do cliente erros pré-existentes, ou seja, aqueles que não foram introduzidos no sistema por seu mau uso (criar um sistema a prova de usuários seria muito caro). Quando trabalhamos contratando empresas devemos ter uma equipe de interface que proteja a empresa contratada, avaliando a especificação fornecida para ela, verificando se o detalhamento é suficiente para que se desenvolva um software, e guardando os interesses da empresa, realizando os testes automatizados (unitários e de validação) e também os exploratórios, garantindo que as funcionalidades tenham sido construídas como contratado. Isso também significa garantir que a Arquitetura do software seja a combinada e adequada para a solução, que o código esteja legível e simples e que os documentos suplementares suportem futuras manutenções.

Com tudo isso feito, ainda sobram erros, muitos erros (na visão do usuário). Quando compramos uma televisão, o vendedor alerta: "Use nos próximos 7 dias, pois qualquer problema basta trazer na loja que trocamos. Depois de 7 dias somente a assistência técnica do fabricante poderá tomar alguma providência". Nos contratos temos que ter períodos de garantia, ou de operação assistida, em que as contratadas tem a obrigação de corrigir erros (dentro de um ANS previamente combinado) e dar assistência ao uso do software, enquanto ele está sendo passado a equipe que fará sua sustentação. Nos nossos contratos este período é de 3 meses, e nosso ANS prevê: tempo para início do chamado, tempo para que o chamado seja finalizado, indicador de retorno do erro, indicador de densidade de defeito... 5% do valor total pago pelo desenvolvimento fica retido até o final deste período para que se necessário sejam feitos descontos relativos a estes indicadores.

Análise/Complemento: Na situação de fim de contrato citada, não há o que fazer. Se o cliente contrata outro fornecedor ou ele mesmo assume a manutenção, qualquer mexida que ele fizer no sistema irá quebrar a garantia do fornecedor original.

Análise/Complemento: A melhor comparação de software com outras indústrias é a construção civil. Software não é um número indefinido de casos (carros) a partir de um único molde. As construtoras dão cinco anos em alguns itens em um imóvel comprado recentemente... Dá até medo os preços de imóveis em muitas cidades.

Análise/Complemento: O cliente poderia contra-argumentar que contratou a empresa para desenvolver uma solução (hardware, software, SO, BD, storage, infra etc). Como nenhum requisito foi alterado e a solução ficou obsoleta de uma hora pra outra, não seria responsabilidade do fornecedor corrigir o problema? Até que ponto um cliente deveria se meter em discussões técnicas de arquitetura?

Uma coisa é imaginar que o sistema deve suportar milhares de usuários simultâneos, com tempo de resposta inferior a 5 seg, suportando bilhões de registros em bases de dados.

Outra coisa é exigir do cliente que ele saiba que FAT32 suporta somente arquivos de até 4Gb, que NTFS se perde quando o caminho (estrutura de pastas/diretórios) excede algumas centenas de caracteres. Há defeitos cuja caracterização como manutenção corretiva é polêmica.

Peço licença para elocubrar novamente: Será que a motivação da cláusula citada (contratos do governo) não seria uma tentativa de tornar mais prático este assunto controverso?

Análise/Complemento: Temos que ter em mente que uma demanda pode ter requisitos funcionais e não funcionais que devem ser atendidos e a garantia deve cobrir tudo.

Se o cenário e contexto mudam, a garantia não se aplica. Em toda garantia, seja qual for o produto, existem as condições de uso,. Se pegarmos um carro zero e formos fazer Rally, não teremos garantia no caso de algum defeito ou quebra. Porém, se o defeito for oculto e for descoberto por conta de uma situação extrema em que o produto foi sujeitado, a garantia tem que ser dada. Na prática, sabemos que se a garantia envolver ônus o que teremos é um jogo de empurra.

A solução pode ser simples, como a criação de um índice, mas também pode ser complexa, se a causa da queda na performance for uma questão de modelagem mal feita.

É normal os contratos preverem que após seu término a garantia ainda fica vigente por um tempo determinado.

No exemplo da lâmpada queimada no carro, o fornecedor pode te dar uma lâmpada ou até 10 lâmpadas, mas pode não resolver de fato a causa da queima da lâmpada durante a garantia, e na primeira ocasião após o fim da garantia ele te dá uma conta astronômica. Aí começa uma batalha jurídica.

Essa questão da garantia sempre será em todas as áreas uma situação delicada.

Análise/Complemento: De acordo que é necessário qualificar corretamente o que seja problema. Nem todo comportamento indesejado do software é defeito. Na prática a maior parte dos erros se manifesta pouco tempo depois de implantação do projeto.

Dependendo da natureza do software, determinados processos são sazonais, rodam só no final do ano por exemplo. Ou seja, os defeitos latentes nestes processos só irão se manifestar quando eles forem executados, e isto não ocorrerá no período inicial de implantação. Na prática, nem todos os requisitos irão ser usados no período de garantia de 3 ou 6 meses. Ou seja, o teste de cobertura (em operação) abrange apenas um pedaço do projeto entregue. Então não há sentido em ter que pagar para corrigir algo que por uma obra do acaso não se manifestou logo em 3 ou 6 meses. Os ANSs estabelecidos de certa forma induzem o fornecedor a buscar uma entrega de qualidade, mas ainda assim algum defeito vai aparecer e isto pode não impactar o ANS. Digamos que surja um único defeito em determinado mês. Os ANSs continuarão a ser atendidos, mas o cliente irá pagar pela correção do defeito pelo fornecedor e não haverá penalidade alguma, assumindo que ele atendeu as ANS.

Análise/Complemento: Erros fazem parte do negócio, a quantidade de erros e sua severidade é que podem ser controladas. A complexidade de se desenvolver um software, entendendo o código como a explicitação mais detalhada do conhecimento de negócio que o cliente tem, nos leva a supor que erros vão existir. O custo para que não existam pode ser muito alto. Contudo, podemos definir um processo que seja favorável à construção de software com uma densidade de defeitos aceitável para a criticidade do negócio que está sendo automatizado. Não existe almoço gratuito, se não pagamos o suficiente para que os erros sejam detectados e corrigidos em tempo de desenvolvimento, serão descobertos e corrigidos em tempo de produção (com custo dobrado, às vezes mais que isso). O importante é sabermos que densidade de defeito é aceitável para o valor que estamos pagando e que processo envolve o produto para que seja garantida sua qualidade quando aceitamos sua entrega. Testes automatizados, uso de BDD, equipes de interface, iterações curtas, o cliente como parte da equipe, critérios de validação pré-definidas, padrões de programação, uso de componentes, separação em camadas e uso de um framework consistente são boas práticas que ajudam muito a evitar os erros. Nada disso se restringe ao uso de PF, serve para qualquer tipo de contrato e não é só um problema quando o governo contrata, a iniciativa privada também tem este problema. Restringindo esta discussão a responsabilidade pela correção de erros pós entrada em produção, ela deve ser claramente definida à fronteira de entrega, depois disso deve passar a uma equipe de sustentação. Esta fronteira define claramente quanto gastamos para conter os erros e permite que o fornecedor também faça esta conta.

Análise/Complemento: Quando o usuário solicita um serviço de manutenção, seja corretiva, adaptativa ou perfectiva, significa que o sistema está em produção e é útil. Portanto, o sistema está produzindo algum retorno do investimento realizado na sua construção, seja ele em dinheiro ou tempo. O usuário, entretanto, não entende dessa forma. Sua reação típica é emocionalmente negativa, pois considera que se houve algum erro é porque os técnicos não se esforçaram o suficiente para produzir um sistema de qualidade. Se surgiu uma nova necessidade, considera que não foi feita uma análise adequada. Mesmo entre profissionais de tecnologia da informação existe o mito de que é possível construir um sistema sem erro algum. Myers (2004, p. 10) desmente essa crença equivocada ao demonstrar que é impossível testar todas as condições possíveis. Em consequência, o erro é considerado uma característica, e não um defeito do software. Muito se tem discutido no mundo sobre a responsabilidade legal do fornecedor de software, e o entendimento atual é de que não existe responsabilidade naqueles defeitos que não poderiam ter sido previstos (CERQUEIRA, 2004). Portanto, manutenção deve ser encarada como sinal de sucesso de um projeto, seja ela corretiva, perfectiva ou adaptativa. O erro não acontece por imperícia,

imprudência ou negligência do desenvolvedor, nem as necessidades futuras podem ser previstas. Certa vez inseri uma cláusula de manutenção corretiva num contrato de desenvolvimento, a título de garantia. Abri mão da garantia, pois o software continua evoluindo, com manutenções perfectivas (evolutivas) e adaptativas, ficando quase impossível de se identificar quando o erro foi originado.

Análise/Complemento: O ponto não é querer software isento de erros. Isto é uma impossibilidade prática. Assim como é impossível garantir que um carro zero quilômetro ou qualquer produto que se compra seja isento de defeito. Ter um nível de densidade de defeitos baixo implica quase sempre em custo mais alto.

O ponto discutido já parte dessa realidade que o software tem defeitos latentes na sua entrega, mas cujo custo de correção seja de responsabilidade de quem o produziu, não de quem contratou o serviço. O próprio Manual do IFPUG coloca explicitamente esta visão quando aborda manutenção corretiva: o seu custo deve ser alocado ao projeto de desenvolvimento ou melhoria que introduziu o defeito. Se no seu caso você tem equipes distintas para desenvolver e sustentar e que ao final do desenvolvimento uma passa a bola para a outra (com os defeitos juntos), daí fica impraticável cobrar correção de erros de quem desenvolveu sendo que a equipe de sustentação já assumiu a responsabilidade do software. O caso citado é o cliente governo contratando um fornecedor único para executar o desenvolvimento e também a sustentação.

Análise/Complemento: Esse assunto não pode ser avaliado dissociado do contexto em que está inserido e do atual estado da arte.

Não podemos desconsiderar o fenômeno descrito por Pressman:

“Adicionalmente, para que tenhamos 99,99% de eficiência de remoção de defeitos é necessário a adoção sistemática das práticas abaixo (**isso no melhor caso!**)” (Fonte: Applies Software Measurement, Capers Jones).

Os dados do ISBSG indicam progresso: a qualidade das entregas sem defeitos identificados nos primeiros 30 dias é de 50% (em comparação com 12% de 20 anos atrás). Veja que o critério é 01 mês, enquanto hoje se pede entre 03 e 06 (às vezes 12 meses).

Pergunta: Na conjuntura atual, onde se paga R\$ 500,00 em média por PF (levantamento feito pela FATTO com base em 101 editais publicados), é viável esperar que isso aconteça?

Se não conseguirmos instrumentos para fazer essa lógica da contratação de fábrica funcionar de verdade e não uma ficção em que se fabrica PFs para viabilizar contratos.

A publicação do Guia de Boas Práticas em Contratação de Soluções de TI 1.0 pelo TCU e a efetiva fiscalização desses contratos principalmente na medição aparece muito antes disso.

Análise/Complemento: A própria lei de licitações responde:

Art. 69. O contratado é obrigado a reparar, corrigir, remover, reconstruir ou substituir, às suas expensas, no total ou em parte, o objeto do contrato em que se verificarem vícios, defeitos ou incorreções resultantes da execução ou de materiais empregados.

Então pagar manutenção corretiva em um serviço de software que o próprio fornecedor executou é **ilegal**.

Assunto: Como medir jogos de computador?

Data: 23/11/2012

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5759>

Dúvida: Como medir jogos de computador/celular usando PF? Por exemplo, para pontuar um jogo de luta (tipo *Street Fighter/Mortal Kombat*). Devemos contar cada golpe (chutar, socar, pular, abaixar etc) como uma EE de baixa complexidade? Ou só seria uma EE a possibilidade de salvar o jogo, por exemplo?

Análise: Se tivermos a especificação do jogo fica mais fácil fazer a análise. A análise do ponto de vista apenas do jogador é incompleta.

Análise/Complemento: A relação HH x PF é mínima.

Análise/Complemento: Talvez não seja apropriado o uso do método. Seria significativo ao usuário cada ação como chutar, pular, etc.

Também seriam basicamente EEs de baixa complexidade (alterando o comportamento do sistema) e existiriam alguns poucos ALLs como os Parâmetros do Jogo e de Jogos Salvos.

Análise/Complemento: O fator primário de custo de um jogo não é a funcionalidade como medida pelo IFPUG. Seria importante ser feita uma pesquisa para definir a abrangência do esforço que se deseja estimar; por exemplo, o *authoring* do roteiro e da música estão dentro ou fora? As subrotinas de IA estão dentro ou estão fora? E assim por diante.

Talvez se essa abrangência for bem restrita, a quantidade de PF possa até ser uma variável significativa para estimar esse escopo... Mas isso é tudo especulação.

Análise/Complemento: Supondo que o cliente demande transformar jogos simples (caça-palavras, palavras cruzadas, labirinto com perguntas e respostas) e histórias em quadrinhos publicados num PDF em conteúdo desenvolvido em flash (desenho animado), o projeto seria medido usando PF?

Análise/Complemento: Somente se:

- 1) Tivesse dados de esforço e tamanho de projetos realizados numa faixa de pontos de função que se deseja estimar/prescrever...
 - 2) Obtivesse um R2 de pelo menos 75%....
 - 3) A correlação passasse num teste de ANOVA...
 - 4) A normalidade dos resíduos fosse estatisticamente significativa pelo teste de Kolmogorov-Smirnov...
 - 5) A homocedasticidade dos resíduos não fosse negada pelos testes de Breush-Pagan e Koenker...
 - 6) Os resíduos somados se aproximassem de zero.
- Sem qualquer desses passos (talvez com uma maior flexibilidade do item 2), qualquer manifestação é antes de tudo, um ato de FÉ!

Análise/Complemento: É entendido que um jogo muitas vezes extrapola a questão software. Determinados jogos demandam boa parte do esforço associada à elaboração de um filme (roteiro, arte, trilha sonora, fotografia, etc).

Não é sabido se o tamanho funcional do componente software do jogo possui relação com o esforço das atividades, digamos "artísticas", citadas antes. Agora considerando um jogo simples como os citados, é considerado que as atividades "artísticas" tem um peso pequeno no trabalho de desenvolvimento do jogo. Assemelha-se a um projeto de software tradicional, não se vê dificuldade alguma em medir e tem-se absoluta fé de que o tamanho funcional é um item fundamental na estimativa de esforço do projeto.

Assunto: Projeto de mudança de plataforma

Data: 05/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5818>

Dúvida: Como é o cálculo de um projeto para mudar a plataforma (por exemplo, de PHP para Java)? Como isso é quantificado? Consideramos todos os pontos de função do sistema? Ou usa-se outra métrica?

Análise: Qual o propósito da contagem? Qual a razão para execução da contagem de pontos de função?

- Se considerar o que o usuário de negócio vai enxergar nessa alteração, pensando nos requisitos funcionais alterados:

Zero. Se for só mudança de linguagem, *"não existem mudanças para as funcionalidades de negócio associadas com este trabalho"* (upgrade de plataforma, CPM).

- Se considerar um novo sistema, um novo desenvolvimento, independente do sistema anterior e deseja-se saber qual o tamanho funcional do projeto de desenvolvimento deste novo sistema: Conte todos os pontos de função do sistema e, eventualmente alguns PFs das funcionalidades de conversão.

- Se deseja saber qual será o tamanho final do novo sistema, com relação ao sistema anterior: A mesma quantidade de PFs (caso não esteja incluindo, alterando ou excluindo funcionalidades).

Um paralelo: Como é o cálculo do tamanho de uma casa se ela for trocada de tijolo por madeira?

Se a casa tinha 400 m² construída com tijolos, ela vai ter os mesmos 400 m² construída com madeira.

Qual a diferença de tamanho do ponto de vista do morador? Nenhuma. A casa tem a mesma quantidade de m² e cômodos ao final.

Quanto custa trocar a casa de tijolo e reconstruir ela inteirinha em madeira?

Um serviço que será realizado provavelmente em toda a casa, em todos os 400m², no preço que o contrato/edital estabelece que se paga por um serviço desse tipo. Se for o preço de uma casa nova em madeira, paga-se uma casa do zero. Se o contrato/edital estabelece algum redutor já que os requisitos funcionais e documentação anteriores serão aproveitados, paga-se o valor do redutor estabelecido.

Análise/Complemento: Uma mudança de plataforma significa que será desenvolvido um outro sistema para substituir o atual. Trata-se de um projeto de desenvolvimento e por isso devem ser contadas todas funções de dados e de transação, bem como o tamanho dos possíveis utilitários para conversão de dados.

Análise/Complemento: Substituir um sistema significa construir um outro a partir do zero, numa abordagem de reengenharia, estilo big-bang.

Modernizar, por outro lado, significa adaptar o sistema a mudanças significativas para adicionar flexibilidade, isolamento, extensibilidade ou abertura. É a abordagem da melhoria contínua. Pode-se, por exemplo, manter a tecnologia atual e acrescentar uma nova camada com estrutura mais modernas.

Na modernização se aplicam técnicas de manutenção de software, enquanto que na substituição são aplicadas técnicas de desenvolvimento e um novo projeto deve ser aberto.

Bisbal (1997) explica que a substituição deve ser escolhida em último caso. Sempre que possível recomenda-se uma estratégia de modernização devido aos seguintes fatores:

a) O sistema atual já foi amplamente testado ao longo do tempo, enquanto que o novo precisará de testes extensivos e mesmo assim não haverá garantia de ausência de erros que não existiam no atual;

b) Possivelmente o novo sistema passará por um período de maturação em que as funcionalidades terão problemas de integração e desempenho que deixará os usuários insatisfeitos; e

c) Não existe garantia de que o novo sistema será tão robusto ou funcional quanto o antigo.

Além dos fatores considerados por Bisbal, é interessante também levar em conta fatores de gerenciamento de projetos. Quando se trata de desenvolvimento, Ian Sommerville, no seu livro "Engenharia de Software", destaca que só por acaso o projeto termina no tempo, prazo e orçamento previstos. Segundo pesquisa do Standish Group (The Chaos Report, 1995), 31% dos projetos de software são cancelados antes do final; 53% são completados, mas têm falhas em escopo, prazo e custo planejado; apenas 16% são completados dentro do escopo, prazo e custo planejado. Portanto, o risco de desenvolver um sistema é alto. E quanto maior o tamanho do projeto, maior o risco: projetos orçados em mais de US\$ 10.000.000,00 ou com prazos superiores a três anos possuem taxa de sucesso próxima ao zero (STANDISH GROUP, A recipe for success, 1999).

Serviços de manutenção, por outro lado, raramente são mal sucedidos. Portanto, antes de abrir um projeto de desenvolvimento, recomenda-se analisar a viabilidade de modernização, num estudo que deve ser elaborado por especialistas em manutenção de software com experiência em diversas tecnologias, tanto modernas quanto antigas. A decisão de abrir um novo projeto é fácil e muitas vezes comemorada; vencer os riscos, concluir e implantar um novo sistema, entretanto, só se consegue com muito esforço e depende de aspectos que nem sempre são passíveis de controle.

Análise/Complemento: Pode-se avaliar qual o nível de reuso que pode ser aplicado nesse novo projeto. Se todos os requisitos estão bem escritos, os mesmos serão utilizados na nova plataforma. Talvez seja razoável deduzir da produtividade o percentual de requisitos. Podemos ter em alguns casos reaproveitamento da base de dados e etc.

Assunto: Aplicação das Regras

Data: 05/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5819>

Dúvida: Na parte 2 do capítulo 11 do livro, a questão 43 tem o seguinte enunciado: *“O usuário da aplicação do caso anterior construída uma tela no sistema para que alguns usuários com acesso privilegiado pudessem alterar eles próprios o texto das mensagens do sistema. Como essa tela é contada?”* Sendo que o enunciado da questão anterior é: *“Uma aplicação foi desenvolvida de tal forma que todas as mensagens emitidas para o usuário são lidas de uma tabela criada pelo programador. Sempre que o usuário solicita a mudança de uma mensagem, o programador apenas altera o texto na tabela, sem a necessidade de alteração e recompilação do programa. Como essa tabela é contada na APF?”*

Qual a resposta correta?

Análise/Complemento: A resposta correta é: *“A Tela não é contada, pois o arquivo é Dados de Código”*, pois a tabela citada na questão 42 é Dado de Código. No enunciado, é citado que o programador cria a tabela (requisito técnico), e requisitos técnicos não devem influenciar no tamanho funcional da aplicação. Então, as transações que fazem apenas manutenção dos dados de código não devem ser considerados processos elementares.

Assunto: AIE

Data: 06/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5829>

Dúvida: Será desenvolvido um sistema de pedido de venda que será integrado com um ERP já existente. Nesse sistema a ser desenvolvido, será gerado um relatório de Cliente, em que as informações para gerar o relatório vem do cadastro do cliente do ERP integrado. A dúvida é: esse relatório de Cliente vai ser AIE ou ALI?

Análise: O relatório em si será uma consulta ou saída externa. Valide contra as regras dessas funções afim de determinar qual seria exatamente. A questão apontada deve ser sobre a função de dados CLIENTE. Tudo depende do posicionamento da fronteira da aplicação. Se a fronteira do pedido de venda for a mesma do ERP existente, então, temos na verdade uma contagem de projeto de melhoria e, se não houverem modificações em CLIENTE, o mesmo não é escopo da contagem. Se forem duas fronteiras distintas, então CLIENTE será escopo da contagem e será um AIE.

Assunto: Fórmula do calculo dos pontos de Função

Data: 12/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5836>

Dúvida: No Manual de práticas de contagem de pontos de função. 4. ed. BRA: IFPUG, 2005 está descrito na parte 1 pag. 9-5 a fórmula: $DFP = (UFP + CFP) * VAF$ sendo UFP os pontos que ficarão na aplicação (os que são disponibilizados). Este é o mesmo de ADD descrito na cartilha do IFPUG e no Livro?

Análise: Esta versão do manual não é a mais recente. As fórmulas mudaram. Mas o sentido do UFP é o mesmo do ADD no livro. A versão atual do manual está em www.ifpug.org

Assunto: APF Para projetos OO

Data: 13/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5840>

Dúvida: A versão 4.3.1 do Manual de práticas de contagem de pontos de função aborda o processo de contagem de pontos de função com aspectos da orientação a objeto? Há alguma referência para esse tipo de contagem? Onde se consegue encontrar de forma gratuita a versão 4.3.1 do manual?

Análise: Para usar a APF em um projeto OO, esqueça que houve a orientação a objetos e considere os pontos de interface entre o sistema de software e seus usuários humanos ou não. Nessas interfaces há troca de informações que instiga a investigação de entradas, saídas e consultas externas. A partir dessas funcionalidades de transação, procure identificar os requisitos de armazenamento independentemente de qualquer consideração de projeto.

Complemento: Em tempo, os casos de uso concretos descrevem o comportamento do sistema em direção a um objetivo. Os casos de uso que estão no nível de objetivo do usuário (não em um nível de objetivo agregador ou de subfunção) deveriam ser mapeados para uma função.

Já um caso de uso em um nível de objetivo agregador (manter) deve ser mapeado para várias funções. Casos de uso abstratos que não são realizados e não entregam um objetivo do usuário (que não requer passo subsequente ou anterior para a realização de uma de suas tarefas no negócio), não devem ser contados isoladamente como funções. Muitas vezes usa-se um caso de uso de extensão para descrever um incluir, outro para descrever um alterar a partir de um caso de uso concreto listar. Não é considerada uma boa prática; mas em se encontrando, esses casos de uso de extensão entregam um objetivo do usuário e devem ser avaliados na contagem de transações.

Os diagramas de seqüência descrevem melhor o comportamento de um caso de uso nas interações e troca de informações entre os atores; no RUP ele é parte da disciplina de análise e projeto e pode ajudar, associado a um dicionário de dados, a contar os TD.

Análise/Complemento: Não é à toa que não se encontrará muita coisa a este respeito. A Orientação a Objeto é irrelevante do ponto de vista da medição dos pontos de função. Da UML, os casos de uso e o seu diagrama são os artefatos mais adequados para se trabalhar a APF. Os demais diagramas são em geral voltados ao projeto interno do sistema, tem caráter portanto mais técnico. A APF é justamente o contrário disso: abstração total da implementação. Temos uma pergunta do nosso FAQ a este respeito:

[12. É possível utilizar a APF em um projeto Orientado a Objeto?](#)

Sim. APF é uma técnica independente da tecnologia utilizada para modelar ou implementar um software. Portanto um software terá o mesmo tamanho em pontos de função quer venha a ser desenvolvido utilizando tecnologia OO ou uma outra abordagem.

O que poderá diferenciar as duas abordagens é que no projeto OO a produtividade (hora/PF) poderá ser melhor devido ao reuso. Fazendo uma analogia com a construção civil: pode-se construir uma casa de 100m² da maneira convencional ou utilizando módulos pré-fabricados. Em ambos os casos o tamanho da casa será o mesmo, apenas o tempo de construção ou o custo poderá variar.

Assunto: Certificação CFPP

Data: 14/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5843>

Dúvida: Gostaria de informações de como o mercado está encarando a certificação CFPP, se está ocorrendo muita diferença entre a CFPS ou é somente um título?

Análise: Quem tirou a CFPP está mais para algum problema ou imprevisto na hora da prova, nervosismo ou frio na barriga. A prova é a mesma, o CFPS está no aproveitamento da prova, CFPS no mínimo 90 % e CFPP entre 80 e 89%, ninguém estuda para CFPP. Tratar APF com uma pessoa que é pelo menos CFPP é muito mais produtivo do que com alguns analistas que acham que conhecem muito e só não se submeteram a prova porque acham que a certificação é apenas um papel.

Dúvida/Complemento: Em termos de contratação, as vagas anunciadas no mercado quase todas solicitam CFPS, pode ser por causa dos editais governamentais que exigem CFPS ou o fato da CFPP ser nova e desconhecida para outros.

Análise/Complemento: Há um tempo houve um momento em que os editais não podiam mais cobrar a existência de profissionais com títulos nas empresas concorrentes, muitas empresas colocam no edital "profissional habilitado". Isso se reforça quando se trabalha por produto, não por serviço ou hora. Mas uma coisa é certa, independente da certificação, o mercado está carente desses profissionais.

Assunto: Posicionamento da Fronteira

Data: 14/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5845>

Dúvida: Está sendo realizada a contagem de uma aplicação que controla operações TEF (Transferências Eletrônica de Fundos) que possui uma interface de integração com vários sistemas que realizam venda. A fronteira é apenas a aplicação que controla operações de TEF. Ocorre que para possibilitar os testes de desenvolvimento, existe um módulo que simula os sistemas que realizam a venda. Como o objetivo é medir os projetos de melhoria, a dúvida é se podemos considerar o módulo simulador fora da fronteira, visto que ele é apenas utilizado para simulação dos usuários da aplicação, mas deve compor o tamanho do projeto de melhoria.

Análise: O simulador é outra App considerando os propósitos destacados. Um TEF é um meio de campo com um monte de SE e EE atualizando um log ao longo do dia e passagens de movimento para legados tipicamente.

Assunto: Recomendação de versão do Manual de Práticas de Contagem para Desenvolvimento de Sistema

Data: 17/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5848>

Dúvida: Em um termo de referência para desenvolvimento de sistema, a versão 4.2.1 é a mais adequada para ser colocada como referência do Manual de Práticas de Contagem de Pontos de Função – CPM (IFPUG), para fins de dimensionamento do tamanho dos sistemas e esforço necessário para realização dos serviços? Ou a versão 4.3.1 é mais apropriada?

Análise: A versão 4.3.1 é a mais apropriada.

Isto porque é uma evolução da 4.2.1. Não houve mudanças substanciais de conceitos da 4.2.1 para 4.3.1. Houve uma melhoria na padronização dos métodos de medição e uma melhor organização dos conceitos nos capítulos do CPM.

Não há nenhuma justificativa para montar o termo de referencia com a versão 4.2.1, até porque o mercado já trabalha com a versão 4.3.1.

Complemento: O uso da versão 4.2.1, para pessoas mal intencionadas, pode ocasionar um aumento na contagem em função de uma regra que não adequadamente implementa o princípio da APF em medir os requisitos funcionais do usuário. Esse defeito na regra foi corrigido a partir da versão 4.3.1.

Assunto: Contagem de Componentes acessando dados de outra aplicação

Data: 17/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5851>

Dúvida: Há um componente que busca dados de outra aplicação, exclusivamente para uma consulta. Esse componente foi implementado para atender várias aplicações na empresa e a consulta já é de entendimento comum de todos que deve ser contada. Porém há uma divergência sobre contar ou não o AIE.

Análise: Se os dados consultados por essa consulta são referenciados mas não mantidos pela aplicação sendo contada, então estão contidos em um AIE que deverá ser considerado na contagem.

Para a APF importa os dados negociais, sob o ponto de vista do usuário da aplicação. Ou seja, se o negócio concorda que os dados são mantidos em outra aplicação, isto é, fora das fronteiras da aplicação sendo contada, então deverá ser contado como AIE na aplicação que faz a referencia ao mesmo.

Complemento: Mas, para efeitos de pagamento para a contratada por utilizar um componente pronto em um sistema?

Será uma funcionalidade que não será implementada pelo projeto? Cobrar somente a função transaccional?

Análise/Complemento: Nesse caso de remuneração de contratada, deve ter uma convenção local de contagem na qual o uso de componentes externos seja tratado. Se não tiver, é uma boa hora de pensar em construir um.

Análise/Complemento: Concordamos que o tamanho do sistema em PFs é uma coisa e o pagamento pelo esforço de implementação é outra coisa. Provavelmente há um contrato entre a empresa que desenvolve e o usuário. Tem que ver se o contrato prevê esta situação. Isto poderia ser compensado nas Características Gerais do Sistema com o Valor de Fator de Ajuste. Temos que tomar cuidado para não errar na contagem dos PFs para compensar um esforço que realmente existe.

Assunto: Contagem de relatório para diferentes perfis

Data: 18/12/12

Link: <http://br.groups.yahoo.com/group/livro-apf/message/5854>

Dúvida: Um sistema tem 4 perfis. Para cada perfil logado, os campos para gerar o relatório são diferentes. Dúvida: É contado apenas um relatório ou 4 relatórios?

Análise: Tendo como base essa simples descrição, são 4 transações diferentes já que os campos (DER) são diferentes. Caso fosse o mesmo relatório sendo executado por autores diferentes, aí sim contaria apenas uma transação. É importante lembrar que a documentação do usuário deve refletir claramente os relatórios e a diferença entre eles.

Complemento: Não é o fato de haver variação na apresentação dos DERs que caracterizaria ser uma outra função transacional. O que confirmaria ser uma transação diferente seria o fluxo da transação. Pela descrição acima, o relatório é o mesmo, o fluxo negocial também e os campos mudam conforme o perfil que está logado. Sendo assim, é apenas uma função transacional. Se tivesse, por exemplo, um nome diferente para cada relatório e um item de menu para chamar cada relatório, então seriam fluxos diferentes e funções transacionais diferentes.

Análise/Complemento: Então se tivesse o relatório "faltas do funcionário no mês" para o perfil X que mostrasse (recuperasse) somente os dias em que o funcionário X faltou naquele mês, e para o perfil Y tivesse relatório "faltas do funcionário no mês", que além dos dias que ele faltou, mostrasse também um gráfico do percentual que isso representa do total de dias trabalhados naquele mês, seria contado os relatórios como uma única transação?

Análise/Complemento: O ideal seria perguntar aos usuários (de cada um dos 4 perfis citados) como é que cada um deles vê esse relatório. É a visão do usuário que determinará como o tal relatório será contado: se um ou vários.

Análise/Complemento: O que caracteriza processos elementares diferentes basicamente são: Lógica de processamento distintas, conjunto de tipos de dados distintos e arquivos referenciados diferentes. Pelo descrito, o conjunto de dados muda conforme o perfil logado, então temos diferença do conjunto de dados exibidos, logo temos relatórios diferente. O CPM cita inclusive um exemplo parecido no capítulo 2: "quando o requisito funcional do usuário especificar a necessidade para dois relatórios semelhantes (tal como o Relatório 1 que contém Nome do Consumidor, Identidade do Consumidor, e Endereço e Relatório 2 que contém Nome do Consumidor, Identidade do Consumidor, Endereço e Telefone), os relatórios são identificados como PEs separados, uma vez que o requisito funcional do usuário especifica a necessidade para diferentes DERs. Os relatórios não são combinados em um PE único apenas porque têm DERs semelhantes"

Análise/Complemento: Se em uma consulta simples de pessoas, para um determinado perfil é exibido CPF e para outro não por requisitos funcional do usuário, são relatórios distintos.

O que pode ser investigado é qual a necessidade do negócio de apresentar variação de dados para cada perfil, qual o impacto para o negócio gerar um único relatório com todos os dados para todos os perfis, mesmo que para um deles não contenha dados desnecessários.

Análise/Complemento: Foi informado que a variação do relatório não é por requisito funcional do usuário, mas por uma variação do perfil que está utilizando a função. Claro que isto muda o resultado da contagem.

Análise/Complemento: Os relatórios podem ter variações mas serem a mesma função transacional. Vai depender da forma como estiver escrito o requisito, partindo da premissa que está trabalhando com documentação. Então o fato de ter um gráfico a mais sendo impresso para o funcionário Y não é suficiente para caracterizar como uma outra função transacional. Onde começa e onde termina a função transacional (fluxo)? No caso, a variação do relatório seria em função da validação do perfil, mas o relatório seria o mesmo. Claro, se a documentação fornecer mais detalhes, então a análise pode mudar.

Análise/Complemento: Qual o motivo para existir perfis diferentes acessando dados diferentes em um mesmo sistema senão um requisito funcional?

Análise/Complemento: Talvez devido a algum requisito de segurança relacionado a uma política organizacional.

Análise/Complemento: Vamos detalhar melhor a situação: O administrador está logado.

Na aba de Relatórios tem uma lista com todos perfis do sistema.

Se selecionado o perfil A, os domínios dos campos serão preenchidos de acordo com uma regra de negócio que busca os domínios dos demais campos do relatório.

Se selecionado o perfil B, os domínios dos campos serão diferentes dos outros perfis.

E assim por diante.

Esta diferença no domínio dos campos é por questão de segurança.

Importante: Os campos apresentados na tela de geração não mudam. Muda apenas o domínio desses campos de acordo com o perfil selecionado.

Análise/Complemento: A situação descrita é uma única função transacional (podendo ser uma saída externa ou uma consulta externa). Isto porque não está claro se os campos do relatório são apenas consultados ou campos com informações derivadas. Para enquadrar desta forma eu entendo que os passos da função transacional são:

1) selecionar o perfil

2) exibir o relatório, de acordo com o perfil escolhido.

Observação: A exibição dos domínios específicos, vinculados a cada perfil, variam conforme regras de negócio (lógicas de processamento da função).

Enfim trata-se de apenas uma função transacional.

Análise/Complemento:

Esses perfis não seriam indicativos de diferentes tarefas no plano da divisão do trabalho em funções (nada a ver com o design da solução)? Não seriam dois diferentes requisitos funcionais onde um poderia ser aprovado e outro não aprovado?

A unificação dessas necessidades de um perfil e do outro perfil em uma única "unidade" não é uma decisão de projeto?

Análise/Complemento: Qual é a necessidade do usuário em realizar essas separações? Foi falado em segurança, mas em que sentido? O que cada perfil pode realizar no seu negócio tendo em mão estes relatórios? Enfim, qual intenção do usuário com estes diferentes relatórios?

Análise/Complemento: A unificação das necessidades em um perfil e de outro perfil em uma única "unidade" pode ser uma decisão de projeto, mas também pode ser uma decisão da função comercial. Apenas entrevistando o "dono" da funcionalidade podemos ter certeza, que é o usuário definidor. Outra coisa que poderia matar e solucionar de vez a questão seria um bom caso de uso (UML).

Análise/Complemento: Sendo uma decisão do negócio, isso implica em duas tarefas e, portanto, dois requisitos funcionais diferentes. Para nossos propósitos de medição, mesmo que esses dois requisitos funcionais do usuário estejam, equivocadamente, em uma única especificação técnica, ainda assim haverá duas funções. Uma função não pode ser subdividida considerando como referencial exatamente as tarefas e serviços no negócio.

Análise/Complemento: Parece que estamos fazendo análises diferentes porque estamos partindo de premissas desiguais. É o requisito que direciona todo o processo de desenvolvimento. Quando identificamos situações onde o requisito está descrevendo uma única atividade, então consideramos assim como está descrito. Havendo reclamações no resultado da contagem, alteramos a contagem somente se o requisito for corrigido e passe então a descrever mais de uma atividade.

Nesse processo direcionado pelo requisito, tem que ficar clara a existência de mais de uma atividade. Por exemplo, em um caso de uso, quando um fluxo alternativo não traz um passo de encerramento mas

retorna ao passo do Fluxo Principal que o chamou, então consideramos apenas uma atividade. Por outro lado, se há um passo de encerramento no fluxo alternativo e outro no Fluxo principal então consideramos duas atividades e, portanto duas funções transacionais na medição.

Análise/Complemento: O conceito aplicado acima nas contagens em que o fluxo alternativo que retorna ao fluxo principal representa uma atividade não ficou claro. Se há uma consulta que chama uma consulta detalhada são duas atividades. Não é correto criar regras dentro da UML para contagens de pontos de função. Cada caso de uso deve ser analisado sem regras para o correto entendimento da funcionalidade.

Pedir para o fornecedor escrever o que a Caixa pode não estar recebendo também é extremamente perigoso. A obrigação do analista de Métricas é investigar e entender corretamente a demanda e somente desta forma iniciar uma contagem.

Requisitos ruins e casos de uso péssimos fazem parte da nossa realidade. A metodologia de pontos de função é clara, o difícil é entendermos corretamente o que foi solicitado pela Caixa e entregue pelo fornecedor. Para esta atividade, precisamos entender o negócio e os principais conceitos de Engenharia de Software.

Análise/Complemento: É a lógica da fábrica aplicada à medição. Se são tarefas no negócio diferentes e apenas um RFU, a granularidade deste último não é a mesma granularidade da descrição de uma funcionalidade de acordo com a APF. O analista de métrica deve certificar-se quanto à isso; caso contrário, basta automatizar a contagem dos requisitos... Será uma aproximação, muitas vezes válidas, mas não uma medição. A contagem é feita exatamente no plano das tarefas e serviços do usuário. Se temos diferentes tipos de usuário que compartilham um programa ou uma especificação, não é isso que levará a apenas uma função. O bom de ter um padrão é esse: não basta a vontade de um cliente gritar, conte-se um. Nesse sentido, o COSMIC é melhor definido que a APF no sentido de mapear o objetivo da ISO 14143 de medir requisitos funcionais do usuário em um método de medição. Afinal, não é porque há uma única especificação chamada APLICAÇÃO que apenas uma função será contada. Um dos motivos de termos entrado no ramo também da engenharia de requisitos, foi o de ilustrar de maneira pragmática qual o critério para identificar um RFU. Milhões de reais estão sendo jogados no lixo pela lógica da fábrica... Principalmente se a remuneração de quem conta é proporcional aos pontos que conta.

Análise/Complemento: Única atividade quer dizer único processo elementar? Quando uma Instituição utiliza algum processo, seja qual for, e que ainda um dia possa "sofrer" auditoria, deve-se ter ao menos requisitos/casos de usos documentados e bem arquivados: assinados por quem fez (da contratada ou funcionário público) e digitalizados num servidor e com espelho, o mesmo vale para as contagens, de acordo com a última auditoria no TCU no Governo.

O que se deve atentar é que descrever um requisito funcional de negócio com tudo que foi solicitado e bem escrito e ainda for implementado conforme solicitado, é bem difícil de acontecer. Apesar disso, o analista de métricas deve analisar minuciosamente para entender o processo elementar: normalmente é analisado a solicitação da área de negócio, os requisitos, modelo de dados, e todos os documentos que estão num repositório, e se for uma contagem detalhada, normalmente já houve implementação do código e é verificado com o desenvolvedor se tudo foi feito e se há compatibilidade com o que foi pedido e documentado.

A contagem começa após tudo isso, aí sim podemos mudar o requisito caso não esteja claro, faltando algo ou sobrando, após todos esses batimentos. Um requisito pode ter uma, duas ou várias funcionalidades, independente as vezes.

Quanto a esta regra, utilizada em caso de uso no fluxo alternativo quando não há encerramento, é apenas uma forma de escrever que pode ser errada ou certa, devemos analisar e entender se o fluxo alternativo é um processo elementar ou não, dois ou mais, se foi encerrado ou se retorna ou fluxo principal. Isso não é conclusão para se determinar um processo elementar, não é uma técnica profissional ou metodológica, mas sim uma má prática sendo expandida dentro de uma Instituição. O cliente ou um certificado não pode dizer que é apenas uma função transacional sem argumentos, simplesmente para dizer que manda ou que é autoridade no assunto, isso tudo é muito sério para ser tratado assim.

A contagem deve ser conforme as regras do CPM, e não para beneficiar o cliente ou a contratada. Tudo que envolve dinheiro público devemos estar muito atentos já que a medição é um passo para tal, mas se existe uma técnica que foi escolhida seja em licitação ou pregão ela deve ser seguida de forma correta. Infelizmente estão mudando o objetivo, o conceito e o propósito da análise de pontos de função nas Instituições, não o IFPUG, mas talvez infelizmente por briga de poder, quem sabe mais ou quem manda mais, isso não é ético nem mesmo técnico.

Análise/Complemento: Não se trata de criar regras dentro da UML para contagens de ponto de função. Mas sim a utilização de um artefato da UML, o caso de uso, como insumo principal para a identificação dos requisitos funcionais do usuário. Isto é uma combinação entre área de negócio, área de TI e empresa contratada para desenvolver o software. Se não é uma possível trabalhar desta forma então a utilização do conceito fica descartada.

O caso de uso, pela definição acima, contempla na sua estrutura um fluxo principal e fluxos alternativos. Os fluxos alternativos podem mapear uma função independente da função tratada no fluxo principal. Isto ficaria evidenciado se o fluxo alternativo possuir um passo de encerramento. Assim teríamos dois pontos distintos de encerramento, um no fluxo alternativo, outro no fluxo principal, com lógicas totalmente diferentes. Isto é uma evidência muito forte de que teríamos duas funções transacionais, do ponto de vista do usuário e no conceito estabelecido pelo CPM. Mas ainda sim, poderia haver alguma exceção para este conceito. (Exceção).

Análise/Complemento: Sim, na discussão anterior a expressão “única atividade” poderá ser entendida como único processo elementar.

Todo analista de métrica deve analisar minuciosamente para entender o processo elementar, analisando a solicitação da área de negócio, os requisitos, modelo de dados, e todos os documentos que estão no repositório.

Sem desconsiderar tudo isto, há situações em que é necessário estabelecer um acordo de serviço, juntamente com um processo de trabalho. Existem casos com um volume tal de demandas onde sem processo e sem acordo de trabalho, simplesmente as coisas não evoluem. É necessário firmar um acordo entre Área de negócios, Área de TI e Empresa desenvolvedora do SW, segundo o qual o Caso de Uso pode ser a “palavra final” para dirimir quaisquer dúvidas inclusive as questões dos pontos de função das funcionalidades, por exemplo. Neste sentido, muitas vezes o que está escrito no caso de uso e o resultado da contagem não fica correto se o caso de uso estiver mal escrito. Neste acordo consensual entre as partes uma nova contagem poderia ser solicitada, desta vez com o caso de uso alterado e assim refletindo corretamente o real ponto de função da funcionalidade. Lembrando sempre que nada substitui o bom senso e que, nos casos de Administração Pública, não podemos fugir dos contratos firmados e do que neles ficou estabelecido.

Análise/Complemento: Com relação à dúvida inicial, a princípio poderia ser 4 processos elementares (4 CEs ou SEs), já que são 4 relatórios diferentes. Se fôssemos imaginar uma tela de relatório para cada perfil seriam contados 4 PEs, certo? Mas nesse caso, como o requisito aglutinou os 4 relatórios em uma tela só, é possível que seja apenas 1 PE.

Análise/Complemento: Não é a redação de uma especificação em uma unidade ou várias que é o referencial da contagem; são as tarefas e serviços do usuário. Se são 4 tipos de usuário diferentes e uma especificação "aglutinando" diferentes tarefas, não deve mudar a análise.

A métrica funcional deve ser feita na perspectiva das tarefas e serviços do usuário, caso contrário não é aderente à ISO/IEC 14143. O CPM deve amadurecer mais na implementação desse princípio. A mudança na 4.3.1 no sentido de caracterizar as "pequenas variações" foi um passo importante nesse sentido. Mas há espaço para evoluir...

Assunto: Contratação de serviços de medição

Data: 28/12/12

Link: <http://br.groups.yahoo.com/group/livro-afp/message/5905>

Dúvida: Tem sido uma prática de alguns órgãos públicos ao contratar serviços de fábrica de software, contratar também uma outra empresa (distinta da fábrica) para fazer as medições e estimativas de pontos de função dos projetos da fábrica de software. No sítio do Governo Eletrônico são dadas três sugestões para se contratar serviços de medição

(<http://www.governoeletronico.gov.br/sisconteudo/nucleo-de-contratacoes-de-ti/perguntas-frequentes/contratacao-de-ti/#13779986>), cuja cópia está abaixo.

P: Quais formas eu posso utilizar para contratar uma empresa para fazer a contagem de pontos de função de um software? (12/08/2011)

R: Existem várias formas de se contratar uma empresa para contagem de Pontos de Função, cada uma com suas vantagens e desvantagens. A seguir as formas mais comuns:

1) Contratação por projeto de contagem (valor fixo): A empresa candidata analisa a documentação do sistema e estipula um preço fixo para contagem de PF. A empresa candidata precifica com base nas horas estimadas para execução do serviço. É uma das formas adotadas pelo mercado. Neste tipo de contratação é necessário expor o sistema ou parte da documentação para os fornecedores terem condições de elaborar a sua proposta.

2) Contrato por hora: Paga-se pelas horas despendidas na contagem. Um desvantagem clara deste forma de remuneração é que quanto mais tempo a contagem demora, maior a remuneração.

3) Contrato por PF Contado: Paga-se pelo total de PFs contados. O preço varia pelo tipo de contagem (estimada ou detalhada). Uma desvantagem desta forma de remuneração é que o fornecedor pode aumentar a contagem de PF para ganhar mais, onerando duplamente o órgão (na contagem e no pagamento do desenvolvimento).

Salientamos que, para se fazer a contagem detalhada de forma mais precisa, a contratada necessitará da documentação do sistema (requisitos, modelo de dados, etc). Caso não possua toda documentação a contagem perderá um pouco de precisão e terá de ser feita com base nas telas, relatórios e tabelas do banco de dados do sistema. Lembramos que, em caso de redesenvolvimento, é boa prática considerar um percentual adicional sobre o total de pontos de função contados do sistema legado, para ajustar o total de PF previsto para desenvolvimento do novo sistema, contemplando, assim, a evolução de requisitos.

No contexto de uma contratação vinculada a um contrato de fábrica de software, a abordagem 1 é inviável de se aplicar pois não se conhece a priori quais projetos serão demandados para os serviços de medição.

A opção 2 é ruim pois na prática é uma alocação de recursos, sendo que o pagamento é desvinculado de resultados produzidos.

A opção 3 tem sido mais frequente porém incorre em um conflito de interesse; quanto mais pontos de função o fornecedor de métricas encontrar, mais ele receberá.

Há alguma abordagem diferente destas ou tem alguma ideia nova para esta questão?

Análise: As 3 sugestões propostas pelo governo são ruins. O melhor modelo é a contratação de alocação **por hora, porém com produtividade mínima associada** (taxa de entrega da contagem), ou seja, em um determinado contexto organizacional, é estimado que seja possível em condições normais contar 100 PF detalhados/dia, então paga-se por hora porém observando-se este limite.

Complemento: No TST foi realizado um pregão no modelo 3. Entretanto, para diminuir o risco da contratada realizar contagens maiores que o real, introduzimos um acordo de nível de serviço, no qual uma amostra das contagens será verificada por servidores e, caso estejam erradas, a contratada será descontada nos pagamentos.

O modelo 2 parece que também poderia funcionar, mas precisaríamos pelo menos de medidas confiáveis de produtividade de contagem de pontos de função.

Análise/Complemento: A sugestão acima mistura as opções 2 e 3. O conflito de interesse continua presente, pois se o fornecedor contar mais PFs atinge mais facilmente a produtividade mínima que citou. Além disso, se o cliente pagar por hora, derivado dessa produtividade, não tem como ele prever de antemão quantas horas seriam necessárias para o trabalho. No fim das contas continua igual à opção 3.

Análise/Complemento: As ações citadas são fundamentais; não só para a opção 3, mas para todas as abordagens. Mas ainda assim o conflito de interesse não é eliminado.

Análise/Complemento: O conflito de interesse vai existir sempre, a menos que o próprio órgão tenha uma equipe de servidores especialistas no assunto. Os 3 modelos são provisórios, enquanto se monta uma equipe interna para fazer o trabalho. A questão é que a coisa vira (ou) um provisório-eterno, pois parece que os gestores públicos não querem o desgaste de ter de brigar com o Ministério do Planejamento para conseguir as vagas efetivas necessárias à composição dessa equipe, principalmente porque há muito mais interesses "diversos" por trás dessa cortina do que pode supor nossa vã ingenuidade.

Análise/Complemento: Por que não medir em função do número de páginas que deve ser analisado?

Análise/Complemento: Contar ponto de função hoje no Brasil é contar dinheiro. Pensando como cliente que contrata fábrica de software ou como dono de fábrica de software ninguém gostaria de terceirizar uma atividade tão crítica como essa. Optaria por formar gente da casa para ter essa competência e ficar responsável por isso. Mas a realidade de muitos é que não há gente em casa suficiente para assumir esta missão e contratar mais gente nem sempre é uma opção viável. Então resta a terceirização.

O cliente não tem que sequer ver a contagem produzida pela fábrica. Ele tem que produzir a sua própria medição. Isso obriga-o a também indiretamente verificar e validar requisitos. Uma prática comum em alguns lugares é o cliente receber a contagem da fábrica e "validá-la". Mas muitos não fazem isso. E às vezes o cliente acaba sendo induzido ao erro quando analisa a contagem da fábrica, que muitas vezes tem o viés da implementação e não da visão do usuário.

Mas voltando ao assunto da terceirização; é a realidade instalada. Falta encontrar o jeito mais correto de lidar com ela.

Análise/Complemento: Ao final das contas continua mesmo igual à opção 3 em relação ao conflito de interesses. Não sendo conveniente pagar por preço fechado, por hora ou por PF, e nem mesclando ambos, não é possível uma solução que resolva integralmente o problema proposto.

Análise/Complemento: A melhor opção é a empresa contratante ter pessoas e fazer a sua própria contagem. O modelo fábrica de métricas sempre será falho, além de ter um custo muito maior do que fazer as suas próprias contagens.

Análise/Complemento: No modelo no TST, não eliminou-se completamente o conflito de interesses, mas reduzimos bastante ao penalizar erros grosseiros.

É difícil definir o esforço de contagem pelo número de páginas de documentação a ser analisada já que: 1) o mesmo requisito escrito por pessoas diferentes terá tamanhos diferentes; 2) temos necessidades de contar o tamanho de sistemas que já estão prontos e não tem documentação; e 3) estamos caminhando para um modelo de desenvolvimento ágil onde se valoriza mais software funcionando do que documentação.

Análise/Complemento: O melhor seria o próprio cliente fazer as medições. Mas não acredito que isso seja possível de fazer em todas as organizações, por isso temos que amadurecer o modelo de contratação dos serviços de medição. Não acredito que ele sempre será falho, acho que apenas é pouco tempo de experiência para se dizer que temos uma resposta definitiva. O próprio modelo de contratação de fábricas de software por ponto de função ainda está evoluindo. Minha provocação foi no sentido de fomentar ideias para melhorar o cenário atual.

Análise/Complemento: No modelo de contratação do início da discussão, o fornecedor do serviço de medição está restrito ao trabalho de medição, sem agir na disciplina de requisitos. Na situação de ter que medir um legado sem documentação, o fornecedor terá um trabalho muito maior para medir o sistema pois terá que realizar também elicitação de requisitos, entrevistar usuários, rodar o sistema em ambiente de testes, etc. No seu próprio contrato não seria possível acionar o fornecedor para medir um sistema sem documentação, pois o documento de requisitos é uma das entradas do processo. E essa é a regra de todos os contratos de serviços de medição. Não cabe ao fornecedor entrar na disciplina de requisitos, apenas receber a documentação de requisitos, avaliar se ela é suficiente para a medição e seguir em frente no trabalho ou devolver a ordem de serviço. A disciplina de requisitos é responsabilidade da fábrica de software ou do cliente.

Com relação a contar pelo número de páginas, o volume de documentação é um elemento que implica em mais ou menos trabalho para quem vai fazer a medição. O fornecedor do serviço de medição não tem como influenciar na produção de mais ou menos documentos de requisitos. A fábrica de software não tem interesse em gerar mais ou menos documentação, apenas atender à metodologia do cliente. Então, esta hipótese de usar a quantidade de páginas como métrica para remunerar o serviço de medição elimina o conflito de interesse que citei no início da discussão.

Certamente o volume de documentação para um projeto varia de empresa para empresa, mas dentro da mesma empresa, que possui um processo de desenvolvimento formal e instituído, não há variação significativa quando produzido por pessoas diferentes. O que influi mais fortemente nisso é a metodologia. Alguns órgãos públicos exigem até 30 artefatos a serem produzidos pela fábrica de software. Mais papel, mais trabalho. No modelo ágil, embora a documentação não seja prioridade, ainda que se fique restrito a histórias de usuário, isto já permite produzir estimativas de pontos de função. Para a medição final pode ser usado o próprio sistema ou seu manual.

Essa ideia da quantidade de páginas já fez (e ainda faz) muita gente torcer o nariz; mas pode ser melhor investigada. E também temos que pensar em outras hipóteses. Enfim, abrir a cabeça.

Análise/Complemento: O raciocínio de contar pela quantidade de páginas é falho, e por uma razão simples de ser antecipada. A afirmação de que "a fábrica de software não tem interesse em gerar mais ou menos documentação, apenas atender à metodologia do cliente" pode até ser verdadeira atualmente, mas certamente não o será em um cenário em que a fábrica passe a ser remunerada com base no volume dessa documentação. Ciente disso, a fábrica passaria a ter interesse, sim, em gerar mais documentação, tendo o poder de influenciar positivamente na própria remuneração.

Por mais que a solução do TST também não seja perfeita, me parece o melhor dos cenários, dentre os propostos. No Cenário 3 (medição paga por PF contado) se paga tanto a contratada para medir quanto a fábrica, efetivamente, pela quantidade exata de PF. Nada mais justo. Com um bom acordo de nível de serviço no contrato da contagem, e uma equipe própria mínima para realizar o controle dessas contagens, é possível minimizar o conflito de interesses no sentido de a contratada inflar as contagens.

Ressalte-se que tal ANS deve ser elaborado tendo-se em mente que a sobrecontagem, por parte da contratada, implicará em duplo prejuízo à administração (no pagamento da própria contagem e no pagamento pelo desenvolvimento). Esse ANS deve, igualmente, prever uma penalização gradativa da contratada, maior à medida em que a contagem que ela gerou se distancia, para cima, da contagem real (teoricamente), produzida pela equipe de controle. Se essa contratada, mesmo com as penalizações monetárias (glosas, multas etc.), reiteradamente extrapola na medição, daí talvez seja melhor partir pra rescisão, pois há forte indício de má-fé nas suas contagens. Esse cenário também já pode ser previsto no próprio ANS.

O problema dessa abordagem é que ela demanda a existência dessa equipe de contagem interna. Nos órgãos que a possuem (como o TST), ok. Mas e naqueles que não a possuem, pelo menos ainda não? (o ideal seria que possuíssem, trata-se de contar dinheiro).

Como a pergunta inicial pedia soluções alternativas, segue uma: A própria fábrica também fica responsável por fazer a contagem, tendo a segunda contratada o trabalho de verificação das contagens produzidas pela fábrica. O ANS, aqui no contrato da fábrica, a penalizaria no caso de extrapolação da contagem com relação àquela realizada pela segunda contratada (teoricamente, a correta). Para funcionar, o pagamento dessa segunda contratada (a que verifica as contagens) não pode ser por PF contado, pois ela não pode ter interesse em inflar as contagens.

Assim sendo, o melhor cenário seria da TST, apenas com essa ressalva, de que ele demanda a existência de uma equipe interna com expertise de contagem, o que pode não ser o caso, dependendo do órgão.

Análise/Complemento: A responsabilidade é de quem está contratando. É estranho uma equipe de desenvolvimento contratar uma contagem para saber o tamanho daquilo que vai ser construído e pago. Como é feita a homologação desta contagem para pagamento se o contratante não entende o que está recebendo?

Deveria ser obrigatório em todas as Organizações que trabalham com esta forma de contratação ter especialistas de métricas com capacidade de verificar todos os faturamentos.

O processo deve ser criado de uma forma clara e que a equipe de métricas da Organização tenha total visibilidade de tudo que está sendo contratado e medido.

A responsabilidade é de quem assina a fatura de pagamento.

Análise/Complemento: Quem gera a documentação é um; quem mede é outro. Se há mais páginas, há mais trabalho para as ler e interpretá-las. Medir o trabalho de medição pela quantidade de páginas que devem ser lidas não muda o volume de páginas a ser lido. Fosse quem produzisse a documentação aquele que fosse medido por ela, sem dúvida, uma coisa afetaria a outra... O processo de desenvolvimento seria medido normalmente em pontos de função. Não há porque um serviço ser medido por uma métrica e outro serviço (a medição) se medido por a mesma métrica.

Análise/Complemento: Em todo caso, mesmo nesse cenário, ainda vejo como necessária a implementação de um controle de contagem por parte do órgão. Isso para evitar a ocorrência de conluio entre as empresas. A fábrica poderia inflar a documentação, fazendo a contadora receber mais, e esta última, por sua vez, inflar a contagem, fazendo a fábrica receber mais. Perceba que o controle na contagem evita essa segunda inflação, apenas. Uma boa forma de se evitar a primeira seria remunerar a contratada para a contagem não com base em uma função linear do volume de documentação, mas sim com base em uma função logarítmica decrescente, na qual, atendida a documentação mínima prevista na metodologia de desenvolvimento de software do órgão, quaisquer documentos extras acrescentassem cada vez menos nessa remuneração.

A solução proposta não elimina a necessidade do controle na contagem, por parte do órgão.

Análise/Complemento: Algum tipo de conflito de interesses sempre vai existir. Em qualquer situação, até mesmo se o Órgão possuir uma equipe interna de contagem. Contar PFs é contar dinheiro. E, como sabemos, a contagem é uma atividade extremamente analítica, cabendo sempre uma discussão sobre a análise feita. Em todas as situações, é possível expor as diversas abordagens de análise, considerar os contextos específicos e chegar um consenso que seja fundamentado no CPM.

Um exemplo seria uma empresa pública que desde 2008 trabalha com um modelo de fábrica de software totalmente baseado em PFs. Há um contrato para a fábrica de desenvolvimento e outro contrato para a empresa de métricas. Uma empresa desenvolve, outra mede e a empresa pública gerencia todo o processo que segue uma metodologia forte. O volume de serviços é bem grande (já foi licitado mais de 340.000 PFs).

No geral, não há conflito. Isto não significa que não tenhamos outros pontos de atenção constante e sempre passíveis de aprimoramento.

O contrato com a fábrica de métricas é uma especialização do tipo 3, abaixo. Há três tabelas para enquadrar os serviços de mensuração.

Uma tabela onde ficou contratado o prazo máximo para a conclusão do serviço de mensuração, conforme o exemplo abaixo:

De 3 a 100 PFs: 1 dia.

De 101 a 500: 2 dias.

De 501 a 900: 3 dias.

De 901 a 1200: 4 dias...

De 1501 a 1800: 6 dias...

De 4801 a 5100: 17 dias .

De 5101 a 5400 18 dias.

Para a remuneração é seguido a fórmula contratual abaixo:

REMUNERAÇÃO = Tamanho equivalente em PF X Valor do Preço Unitário do Serviço X Ponderador Financeiro

O Tamanho equivalente segue um tabelamento

De 3PFs a 50 = 45

De 51 a 60 = 55

De 101 a 125 = 113

De 126 a 150 = 138

De 701 a 750 = 72

A terceira tabela é o ponderador financeiro conforme se segue:

de 3 a 100 = 55,50

de 101 a 600 = 26.6

> 600 = 28,1

A equipe que elaborou este contrato mitigou, através da fórmula e das tabelas, a questão da produtividade ser maior à medida que o Volume de PFs seja maior. Isto é um fato que está em todas as contagens. Por outro lado, quanto a um modelo de remuneração baseado em nº de páginas de documentação, é válido o estudo como uma alternativa a mais. Mas soa estranho que a unidade de medida gerada pelo método APF não possa balizar o próprio trabalho da analista de APF. Os Pontos de Função devem balizar todo o processo de desenvolvimento.

Análise/Complemento: A fiscalização pelo cliente deve existir em qualquer abordagem que se imagine para contratar este tipo de serviço. Bem como acordos de nível de serviço balizando qualidade do serviço e prazos. Quanto à ideia do fornecedor do serviço de medição, apenas "validar" a contagem da fábrica é temerário. Se já existe CFPS hoje que assina a contagem feita por terceiros sem sequer analisá-la, que dirá de uma empresa que pega uma contagem pronta e precisa apenas dar um carimbo OK ou NÃO OK. Para dar apenas este carimbo, esse pessoal cobra muito barato o R\$/PF. Carimbar não é validar; mas é o que tenho presenciado com alguma frequência.

Análise/Complemento: Não se trata de o fornecedor do serviço de medição apenas carimbar OK ou NÃO OK na contagem da fábrica, o que seria altamente temerário. Trata-se de este fornecedor refazer as contagens (ou algumas contagens, pelo menos), *sem saber quanto deu a contagem da fábrica*, atuando como um ponto de controle, como se fosse o controle do próprio órgão, para os casos em que este não possui em seus quadros pessoal com suficiente expertise para tanto.

Nesses casos (órgão sem ninguém com conhecimento em contagem de PF), ou se atua assim ou contrata uma empresa para contagem, que repassa os PF para a fábrica desenvolver. E, entre essas duas alternativas, a segunda é ainda mais temerária. Há alguma outra saída para o caso em que o órgão é incapaz de contar com pessoal próprio?

Análise/Complemento: Se estamos falando do fornecedor fazer a contagem do zero, independente da contagem da fábrica de software, estamos na mesma língua.

Para quem não tem condição de formar equipe da casa para assumir as contagens, só resta a terceirização. Mas ainda assim demanda gente com competência interna para poder fazer o trabalho de fiscalização corretamente.