

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

Assunto: Exportação de arquivo zipado

Data: 08/06/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/2501>

Dúvida: Há uma exportação de um arquivo geográfico em um banco de dados para um arquivo *shapefile*.

Nesta exportação são pegos os dados de uma propriedade,. Porém são gerados 5 arquivos separados e com dados distintos. Exemplo:

- dados de diagnóstico
- dados de projeto
- dados de edificação
- dados de edificação realocadas
- dados do lote

É gerado um arquivo para cada um dos dados acima. Porém para o usuário não ter que ficar fazendo 5 *links* de *download*, foi criada um rotina para gerar os 5 arquivos e gerar um arquivo zip.

Nesse caso, conta-se tudo como uma CE, porém contando todos os campos dos 5 tipo de dados, ou conta-se 1 CE para cada tipo?

Análise/Conclusão: Nesse caso, os 5 arquivos são necessários, não podendo existir a emissão sem que conste os 5 arquivos. Portanto, conta-se 1 CE (lembrando que CE não deve conter fórmula matemática ou cálculo).

Assunto: Particionamento de arquivos .pdf (com figura)

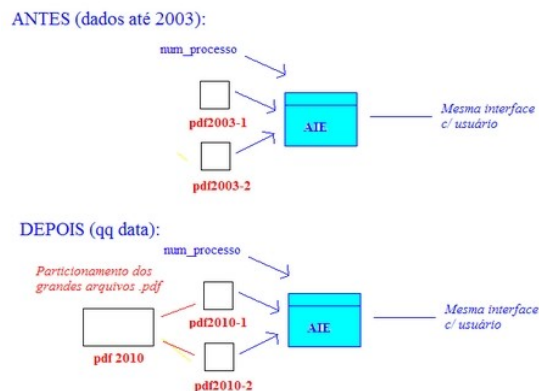
Data: 08/06/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/2503>

Cenário: O sistema de notas taquigráficas do TRF5 estava defasado de forma que os usuários só conseguiam consultar as notas taquigráficas até o ano de 2003. Houve uma alteração no sistema de notas taquigráficas para que o usuário pudesse ter acesso aos votos posteriores a 2003 sem mudar a interface do Sistema (requisito de usabilidade pedido pelos usuários).

Para isso, foi necessária uma mudança na rotina de geração do arquivo de acesso às notas taquigráficas, havendo um particionamento dos arquivos .pdf mais atuais para que pudessem ser pesquisados, pois são maiores que os de antigamente.

Como segue na figura abaixo, não houve mudança de funcionalidade para o usuário. Ele agora pode consultar tudo e, mesmo os .pdf antigos sendo diferentes, pode pesquisar da mesma forma todos os anos.



Houve um esforço do analista em criar a rotina para particionar os .pdf de forma automática e de conversão de todos os dados de 2004 até 2010 para a tabela (AIE).

Deve-se contar a função de conversão de todos os dados até 2010, mesmo esta sendo executada em um AIE (não em um ALI)? Deve-se contar a rotina de particionamento dos arquivos .pdf ?

Análise/Conclusão: É falado em um sistema de notas taquigráficas e colocado os dados de notas taquigráficas como AIE. Qual sistema mantém estes dados? Os dados até 2003 estão em um sistema e os dados após 2003 estão em outro sistema? Por que os dados de 2004 até hoje estão num formato distinto dos anteriores? Por que existe uma rotina de geração destes dados? Por que não lê-los diretamente na fonte? Observe que foi descrito todo um cenário de implementação, mas pouco dos requisitos de negócio envolvidos. É necessário pensar e descrever o cenário abstraindo a implementação, a resposta muitas vezes vem naturalmente.

Assunto: Contagem de novo campo em tela

Data: 10/06/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/2507>

Dúvida: Tem-se um sistema de controle de estoque em ambiente de produção. Nas telas de aprovação do sistema, que trata-se de uma EE, o usuário pediu para exibir mais uma informação, que será a soma de alguns valores. O campo será exibido logo no início da tela de aprovação, antes mesmo do submit da página. Esse cálculo não será armazenado no banco de dados, somente será exibido como informação para o usuário.

Como devemos contar essa alteração no projeto de melhoria?

Conta-se somente o cálculo desse novo campo (como uma SE) ou conta-se a tela toda novamente como uma EE, incluindo esse novo campo?

Análise/Conclusão: Foi informado que a função é uma EE; dessa forma, continua sendo uma EE, sendo que agora aparecerá um valor calculado após a inclusão dos dados, correto?

Se está sendo alterada a EE, então a melhoria será desta EE que alterou sua lógica de processamento e ainda acrescentou um tipo de dado.

Assunto: PCU X APF

Data: 10/06/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/2514>

Dúvida: Ao utilizar o modelo de casos de uso para especificação de requisitos surge sempre a pergunta se a técnica de APF não sofrerá ajustes em função desta representação/paradigma. A técnica de Pontos por Caso de Uso é utilizada atualmente? Há concorrência com a técnica de APF? Como estão os estudos de aplicabilidade da métrica de APF com esta visão?

Análise/Conclusão: A APF independe da tecnologia usada para desenvolver sistemas (inclusive a técnica que será usada para documentar requisitos). Sendo assim, não há nenhum "ajuste" a ser feito na APF para trabalhar com casos de uso.

Há uma falsa "lenda" que alguns acham que PCU é uma técnica mais adequada quando se usa casos de uso e a APF para abordagens mais "estruturadas". Na verdade esta lenda é fruto de desconhecimento da APF por aqueles que pregam isso. Quem já tem algum tempo de estrada com APF sabe que não há dificuldade alguma para se medir a partir de casos de uso.

No FAQ da FATTO (<http://www.fattocs.com.br/faq.asp#P28>) há uma abordagem maior sobre esse assunto.

Dúvida: Então, essa métrica de PCU não "emplacou" e não pode ser considerada concorrente para a APF? Pode-se concluir que não há concorrência no mercado para a APF, pois esta última é, inquestionavelmente, a melhor técnica de medição funcional de sistema ?

Análise/Conclusão: A ideia do PCU é interessante, porém ficou sem um "dono" responsável pela sua evolução. Ela ainda tem um problema de subjetividade em sua aplicação (coisa que também já foi problema da APF no passado) e que inviabiliza seu uso, por exemplo, em contratos de software. O que pode-se dizer é que a APF é uma técnica mais madura que PCU, que tem o suporte do IFPUG para sua manutenção e evolução e uma difusão muito maior no mercado.

Assunto: Banco de dados e *Stored Procedure*

Data: 14/06/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/2524>

Dúvida: Foi feita a seguinte alteração em um sistema: no Banco de Dados foi criado um *job* para que uma *Stored Procedure* execute automaticamente em determinado horário todos os dias. Esse *job* no banco pode ser contado? Quando existem alterações em *Stored Procedures*, deve-se contar essas alterações?

Análise/Conclusão: Nesse caso, a criação do *job* é um requisito técnico porque não envolve nenhuma funcionalidade do usuário e, portanto, não deve ser contado.

Quanto às alterações de *stored procedures*, o que está sendo alterado – algum ARL envolvido, um tratamento de performance, por exemplo (criação de índices, tabelas temporárias, etc), há alguma alteração das funções de transação reconhecidas pelo usuário? Deve-se verificar as funcionalidades que utilizam essas *procedures* e identificar que tipo de alteração foi solicitada.

Se a alteração envolve ARL's que serão alterados, se envolve a alteração da lógica de processamento, isso indica que deve contar os ARL's que serão alterados e/ou as transações envolvidas que serão alteradas. Se for alterações para performance, esta alteração não é considerada para a contagem.

Dúvida: Para tratar uma situação não prevista no início do projeto, ocorrerá uma mudança numa *procedure*, onde será feito um novo *select* para que o resultado dele possa servir de parâmetro para o envio de um *e-mail* que já estava implementado. Note que não haverá mudança em nenhum ALI, mas será necessário uma mudança na *procedure* para ter o e-mail do destinatário que dará sequência no fluxo. Nesse caso deve-se contar esse novo *select* como uma CE?

Análise/Conclusão: O que deve ser observado é a questão do Processo Elementar (PE) ao qual está relacionada este *stored procedure*. Se esta realiza parte ou toda lógica de processamento para algum (PE), logo se para completar este processo, e atender à definição do mesmo, é necessário este e-mail, então a leitura desse ALI ou AIE para obter o e-mail não pode ser considerada um CE à parte.

O que pode ser feito é contar o Arquivo Referenciado relacionado a esta leitura, mas somente caso ainda não tenha contado este ALI ou AIE. O *e-mail* não pode ser contado com TD deste PE por não atravessar a fronteira da aplicação.

Assunto: Alteração de Transação - Inclusão de TD's

Data: 16/06/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/2534>

Dúvida : Em um sistema há as seguintes transações:

* Incluir Contrato - EE - 2 FTR - 30 TD = 6 PF

Foi identificado mais 1 ALI que será utilizado nesta transação e mais 3 TD. Neste caso, na contagem de melhoria deve-se contar a transação novamente somando os itens da transação atual mais os novos itens, ou somente os novos itens:

Incluir Contrato - EE - 3 FTR - 33 TD = 6 PF ou Incluir Contrato - EE - 1 FTR - 3 TD = 6 PF ?

Análise/Conclusão: Deve-se contar a transação toda. Como ela já era complexa, continuará complexa. Deve-se contar a transação com todos os TD e AR, tanto os novos quanto os antigos, ou seja, esta EE vai contribuir com 6PF para seu projeto de melhoria.

Assunto: Gravação de log

Data: 22/06/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/2537>

Cenário: Existe um sistema de *workflow* onde para cada etapa terá a aprovação de um departamento, e toda vez que um usuário deste departamento aprovar a requisição, será gravado o nome deste usuário que realizou a aprovação, assim como data da gravação em uma tabela de auditoria.

No sistema existirá uma tela para que o administrador visualize a tramitação desta requisição, assim como os atores responsável pela aprovação de cada requisição.

Como deve-se contar esta gravação de log ? Pode-se contar da seguinte forma:

Transação	Tipo	AR's	TD's	PF
Aprovar Requisição	EE	Requisição / Usuário / Log	5	6

Onde toda a transação que irá gravar *log*, serão referenciados os ALI's Usuário e Log, Usuário que utilizará o ID do usuário e *Log* será a tabela onde serão gravados os dados de aprovação ?

Ou deve-se contar uma transação somente de gravar log ?

Análise/Conclusão: Nesse cenário não tem porque contar "gravar log" separadamente. Ele faz parte do processo elementar "Aprovar requisição" em cada etapa. Se a transação de "Aprovar Requisição" for a mesma em cada departamento (mesmos arquivos referenciados, mesmos campos, mesma lógica de processamento), devemos contar a transação "Aprovar requisição" uma única vez.

Assunto: Orientação legal da utilização da APF em órgãos públicos.

Data: 23/06/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/2542>

Dúvida: Existe algum instrumento legal que oriente a utilização de PONTOS DE FUNÇÃO em contratos de entidades governamentais com fábricas de software ? Onde podemos obter o acórdão do TCU ?

Análise/Conclusão: Pode-se obter mais conhecimento sobre esse assunto no site do TCU. Há vários acórdãos : item 9.4.1.1, TC-006.030/2007-4, Acórdão nº 1.999/2007 – TCU - Plenário e item 9.2.2.2, TC-019.998/2007-7, Acórdão nº 2.024/2007-TCU-Plenário. E também no *link* que segue:

<http://encosep.ccasj.aer.mil.br/wp-content/uploads/2009/12/IN04xMPSBR.ppt>

Assunto: Alteração do Fator de Ajuste

Data: 23/06/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/2548>

Cenário : Foi desenvolvido um projeto que percorreu todo seu ciclo de vida e foi disponibilizado ao usuário. Depois de entregue o cliente solicitou algumas alterações de requisitos não funcionais do sistema, o que impactou no valor do Fator de Ajuste e alterou seu valor de acordo com suas características.

Cenário Antigo

Total de PF Não Ajustados - 1031

Fator de Ajuste - 1

Cenário Atual

Total de PF Não Ajustados - 1031

Fator de Ajuste - 1,2

Total PF Ajustado = 1237,2

Existe alguma forma de cobrar essas modificações? Foi realizado um estudo e verificou-se que o fator de ajuste passou de 1 para 1,2. Pode-se cobrar 206,2 PF que corresponde diferença do total de PF entre o cenário antigo e o atual?

Análise/Conclusão: O contrato com o cliente deveria prever como seria cobrado um serviço que alterasse apenas requisitos não funcionais. A mudança no Fator de Ajuste não é um bom critério para este fim pois, em muitas situações há alteração em requisitos funcionais e o Fator de Ajuste não se altera. E também há situações em que a manutenção diminui o Fator de Ajuste. Se for seguir o raciocínio do exemplo acima, teria que pagar ao cliente (os PF seriam negativos).

Assunto: SCRUM + APF

Data: 29/06/2010

Link: <http://br.groups.yahoo.com/group/livro-apf/message/2555>

Dúvida: Para os sistemas desenvolvidos através de metodologias ágeis como o SCRUM, é possível a utilização da técnica de APF ou existem restrições?

Análise/Conclusão: Sim, é possível. O que não é adequado é considerar cada *sprint* como um projeto. Em algum momento o projeto de desenvolvimento é concluído e a aplicação entra em manutenção. Esse momento não é ao fim do primeiro *sprint*. Estimar é mais difícil na medida em que o escopo vai se definindo de maneira bastante radical. Contudo, para medir, basta estabelecer o marco do final do desenvolvimento.

Dúvida: Como estabelecer o marco do final do desenvolvimento se o escopo inicial sofrer uma mudança radical? Neste caso há benefícios na utilização da APF? Ou existe uma outra métrica que melhor estime os sistemas modelados com os métodos ágeis como SCRUM?
A única métrica madura e reconhecida pela ISO é a APF?

Análise /Conclusão: Normalmente em projetos usando essa abordagem (SCRUM) esse marco é definido pelo prazo e não pelo escopo. Isso é denominado como "*time-box*". Quanto à aplicação da APF nesse contexto, o principal benefício é estabelecer uma meta de produtividade para o PROJETO, acompanhar essa meta durante o PROJETO e avaliar os resultados obtidos ao final do PROJETO. Ela pode ser usada como somente equacionar o quanto de escopo "cabe" na janela de tempo estabelecida. A APF independe de metodologia. Não fosse assim como comparar o quão melhor é o desempenho em projetos usando a abordagem do SCRUM em comparação a outras abordagens? Existem outras métricas em conformidade a ISO14143, que podem ser encontradas no capítulo 2 do livro APF.

Complemento: Numa empresa que trabalha com SCRUM foi verificado na prática, que perdia-se muito tempo re-contando funcionalidades e isso encarecia barbaicamente o custo do projeto, por não existir um escopo definido. Outro ponto levantado foi que a APF algumas vezes inviabiliza alguma alteração de funcionalidade em virtude do número de pontos detectados algumas vezes serem maiores para alterar do que para desenvolver a funcionalidade. E como os principais clientes não se baseavam em APF, essa foi abolida em projetos SCRUM.

Análise/Conclusão: Nesse caso o problema não foi a medição, e sim como foi feita. Não deve-se contar como funções incluídas, alteradas ou excluídas o resultado do refinamento dos requisitos conforme se realizam novos *sprints*. Isso está errado. Tomando como referência de certo o objetivo de medir o produto entregue ao final do projeto. Um *sprint* não é um projeto considerando os típicos objetivos empresariais.

Tratá-lo como tal revela-se um desalinhamento da tática utilizada para resolver um problema inserido numa estratégia mais ampla.

Complemento: Esse tipo de erro na "tática", ocorre normalmente também em qualquer projeto iterativo. Como foi colocado "isso encarecia barbaramente o custo do projeto, por não existir um escopo definido." Onde está o *Product Backlog*, nesse contexto?

Cada *sprint* não deve ser tratada como um projeto, mesmo, às vezes, possuindo o *time box* e um produto único.

Dúvida: Ao final de cada *sprint* o *product backlog* é atualizado e novas funcionalidades são incluídas, outras são excluídas e algumas outras são alteradas e isso exige uma recontagem de pontos. Como é implementada APF + SCRUM?

Análise/Conclusão: "Tomando como referencia de certo o objetivo de medir o produto entregue ao final do projeto". Deve-se entender que em um novo desenvolvimento o escopo da contagem tange as funcionalidades incluídas e (quando houver) funções de conversão. As alterações e exclusões fazem parte do processo natural de concepção do projeto lógico. Erro está em recontar. Considerar que houve um projeto de melhoria após o primeiro *sprint*.

O "desafio" é a atualização da contagem. De *sprint* em *sprint* deve ser revisto as funcionalidades incluídas, excluídas e alteradas para que a sua contagem de desenvolvimento reflita apenas o *product backlog*.

Genericamente citando: Não existe recontagem e sim atualização. Inclusão de funcionalidades na sua planilha, alteração para refletir a complexidade dos processos elementares e retirar funções excluídas do projeto.

SCRUM e APF funcionam perfeitamente, ainda mais, quando esta metodologia baseou-se também no *FDD - Feature Driven Development*. Quando o *FDD* direciona para o desenvolvimento de funcionalidades.

Complemento: Como ser iterativo e continuar "cobrando" em pontos de função?

Como negociar cada "*sprint*" com o cliente? Uma solução usada foi a da NESMA!

- Cada *sprint* é um "micro projeto"
- Cada *sprint* tem um escopo bem definido
- Cada contagem de *sprint* é feita sobre a linha de base do tamanho funcional do sistema entregue na versão anterior.
- Os *sprints* não podem ser muito curtos. Nem muito pequenos (em Pontos de Função). Em geral, entre 3 a 4 semanas. Isto por que temos um "custo fixo" para executar cada *sprint*.

Considerações ao executar desta forma:

- Só foi possível fazer assim, por que o cliente entrou em acordo.
- Há *sprint* que o cliente ganha. Há *sprint* que o fornecedor ganha. Em geral, acompanha-se a média dos *sprints*.
- Há situações que alguns *sprints* geram uma forte tendência para cima ou para baixo.
- O esforço e disciplina para manter a linha de base do tamanho funcional do sistema é bastante considerado. A planilha é sofisticada.

Existem outras formas de executar esse processo que possa ser citada?

Análise/Conclusão: Situação hipotética: Uma empresa X possui um projeto XPTO e adota o SCRUM como ferramenta ágil para gerenciamento de projetos. Define-se inicialmente as necessidades do cliente. Os requisitos são armazenados em um documento chamado *Product Backlog*. Define-se que o *time-box* de cada *sprint* será de 3 a 4 semanas.

Para o desenvolvimento estima-se que serão necessários 5 *sprints* para a entrega do produto.

Após a 1ª *Sprint* obteve-se, durante a análise e medição, 500 PF.

Perto da entrega da 2ª *Sprint* inicia-se a contagem para identificar o impacto no tamanho funcional.

Histórico da 2ª *Sprint*:

- Foram identificadas alterações em funcionalidades já existentes, realizadas na 1ª *Sprint*.
- Funções foram suprimidas devido a erros na concepção dos requisitos. Outras foram excluídas por vontade do usuário.
- Novos módulos/sub-sistemas foram incluídos.

Por definição, o escopo da contagem de um projeto de desenvolvimento inclui:

- Novas funcionalidade e (quando existir) funções de conversão.

Sabe-se que é natural em abordagens iterativas que haja alterações em funcionalidades. Novas funções sendo vislumbradas e outras simplesmente suprimidas.

O produto do projeto da empresa X é a necessidade do cliente, e somente será suprida após o fim da 5ª *Sprint*.

Após o final da 2ª *sprint* e com a devida contagem deve-se:

- Incluir na contagem as novas funcionalidades identificadas;
- Atualizar as funções alteradas, para refletir a devida complexidade;
- Excluir da contagem as funções suprimidas devido a erros e/ou por vontade do usuário.

Ao final a 2ª *Sprint* obteve-se um total de 970 PF. Acréscimo de 470 PF.

Para o 1º mês, *sprint*, fatura: 500 PF

Para o 2º mês, *sprint*, fatura: 470 PF

Nesse caso não há recontagem e sim atualização a planilha de contagem de acordo com o decorrer do projeto que irá terminar após a 5ª *Sprint*.

E ao final, a planilha de contagem estará aderente ao *Product Backlog*!

Dúvida: Como é possível atualizar a planilha de contagem sem recontar o projeto como um todo?

Análise/Conclusão: Ao olhar para o carro como máquina, ele utiliza como insumo o combustível e nos entrega deslocamento. Contudo, também produz algo que de nada interessa nesse deslocamento, pelo contrário, drena energia que poderia ser usada nele: Calor.

Iniciativas como o SCRUM visam potencializar em produção o que é gasto como "calor" na produção de software. De certo modo, o "calor" é inevitável (de acordo com a termodinâmica... entropia...). O que muda é como e em que grau esse calor é gerado. Quanto o SCRUM gera de calor em relação aos processos tradicionais? Saber responder essa pergunta passa por definir o produto do projeto de *software*. Qual o produto de um projeto de *software* para seus clientes? O produto final... As mudanças que acontecem ao longo do processo em função dos refinamentos dos requisitos dos usuários e o esforço na consolidação e especificação da solução são parte intrínseca e fundamental do processo de desenvolvimento. Optar por empreender essas atividades já desenvolvendo o *software* como na abordagem do SCRUM é acreditar que isso seja mais produtivo que fazer isso em especificações.

As funções que são incluídas, alteradas ou mesmo excluídas durante um *sprint*, as atualizações no *product backlog* NÃO DEVEM SER MEDIDAS em termos de pontos de função seja do IFPUG, seja da NESMA (EFP+TFP). Considerando como objetivo de medição acompanhar a produtividade (como é o caso em contratos com fábricas de *software*), a métrica que deve ser coletada é a medição do produto como em sua configuração final quando da entrega. O indicador é o total de horas/R\$ investidas versus o total de PF medidos ao final do projeto de desenvolvimento (como citado no próprio manual do IFPUG).

Considerando como objetivo de medição o planejamento do escopo/prazo/custo, devem ser feitas estimativas em termos dos conceitos principais do sistema (coisa que não muda conforme se elicitam e

analisam os requisitos) e considerar uma contingência técnica. Essa contingência técnica é o resultado das medições dos erros entre este momento preliminar e a entrega ao longo de um conjunto de projetos.

Para o acompanhamento "micro" do projeto, a FATTO está conduzindo uma pesquisa no assunto. Se o ponto de função ou o EFP+TFP pode ser usado adequadamente nessa micro estimativa traz alguma vantagem em relação a estimativas diretas de esforço (Story Points, por exemplo), é um dos objetos dessa pesquisa.

Em resumo, deve-se ter em perspectiva os objetivos de medição, para fins de avaliar o PROJETO, nesse caso: A APF é a melhor opção. Para avaliar cada SPRINT (interação), não se sabe. Contudo, tem-se uma certeza: Sem instrumentos para garantir o sucesso dos SPRINTS, não haverá sucesso no projeto.

Dúvida: Considerando como objetivo acompanhar o andamento e a produtividade, não poderíamos utilizar a APF e aplicar em *burndown chart*?

Poderia dizer que o “calor” gerado pelo SCRUM será o que chamamos de RISCO?

Uma organização ao escolher o SCRUM como processo padrão de desenvolvimento de *software* deve estar ciente de estar assumindo o risco de não ter “tempo” para compreender as necessidades dos clientes antes de codificar.

A contingência técnica refere-se aos riscos inerentes, retrabalho e má concepção, por exemplo?

Análise/Conclusão: *“Poderia dizer que o “calor” gerado pelo SCRUM será o que chamamos de RISCO?”*

O "calor" gerado é materialização de um risco. O risco de escopo presente em qualquer iniciativa que envolva o desenvolvimento (concepção) de um produto é resultado do refinamento dos requisitos (de todos os tipos) conforme se avança na criação. Seja usando o SCRUM, ou qualquer outra estratégia de desenvolvimento.

Ações para contingenciar esse risco podem ser feitas pela manutenção da média da magnitude do erro relativo da medição em comparação à estimativa em momentos preliminares do processo, preferencialmente segmentada pelo grau do conhecimento do problema quando da estimativa e de outros fatores como por exemplo, a maturidade da visão do usuário quanto ao produto que deseja, o conhecimento dos analistas sobre o negócio, etc.

“Uma organização ao escolher o SCRUM como processo padrão de desenvolvimento de software deve estar ciente de estar assumindo o risco de não ter “tempo” para compreender as necessidades dos clientes antes de codificar.”

Sim, assim como em qualquer estratégia de desenvolvimento iterativa incrementar! Escolher o SCRUM é apostar que o "calor" gerado pelo retrabalho(?) que surge para resolver os problemas da comunicação técnico-usuário seja menor quando comparado a outras estratégias.

Quanto maior o grau de inovação envolvido mais agudo são os efeitos disso. Adotar o SCRUM como um processo padrão é adequado para empresas cujas soluções sempre envolvam um grande e intenso componente de inovação. O mesmo fenômeno (em menor intensidade) acontece com quem usa processos baseados no RUP que prevê que já na concepção os cenários arquiteturais de maior risco sejam resolvidos logo de início o que não extraordinário envolve programar.

Quanto a não se ter tempo para compreender as necessidades dos usuários antes de codificar, isso não é um risco. Risco é um evento que tem uma possibilidade de acontecer. No caso, essa é a própria essência. No SCRUM a especificação dos requisitos do usuário, pode se dizer, está escrita em Java.

“ A contingência técnica refere-se aos riscos inerentes, retrabalho e má concepção, por exemplo?”

Má concepção é um julgamento de valor, assim como retrabalho. A contingência técnica é referente às incertezas originadas de três fontes:

- 1) "Visão do Problema" pouco madura no entendimento dos usuários e que se consolida no processo de desenvolvimento (Revelação? observe a palavra em inglês para revelar fotos);
- 2) "Visão da Solução" sempre envolve alguns componentes inéditos, pesquisa e desenvolvimento; também amadurece durante a revelação;
- 3) Em função da divisão do trabalho, ambos os itens anteriores sofre por problemas de comunicação entre desenvolvedor e usuário;

“ Considerando como objetivo acompanhar o andamento e a produtividade, não poderíamos utilizar a APF e aplicar em burndown chart? ”

Hoje a APF talvez não seja a melhor opção para isso. A menos que se descole das medições e estimativas cujo objetivo de medição seja planejar e monitorar a produtividade . Em outras palavras: um projeto teria 500 PF para o objetivo de medição Planejar e Monitorar Produtividade e 50.000 PF' para o objetivo de medição Acompanhar Progresso).

O *burndown chart* em um projeto SCRUM apresenta em seu eixo horizontal os *sprints* e em seu eixo vertical o "TRABALHO RESTANTE" - ou seja, é um instrumento para Acompanhar o Progresso. A unidade PF' linha poderia ser uma combinação de EFP+TFT ou mesmo algo mais simples;

Dúvida: Nesse contexto, este raciocínio está correto ?

- 1) Existe um *Product Backlog* (que é uma lista de coisas a serem feitas).
- 2) Os Métodos Ágeis assim como o RUP são baseados em Iterações (SCRUM chama a Iteração de SPRINT).
- 3) A cada SPRINT deve ser entregue um código funcionando.

Então:

Product Backlog

A = 5PF , B = 12PF , C = 3PF , D = 20PF

Sprint atual

A

C

D (parcialmente)

No final do *Sprint* a empresa deverá receber o equivalente a $8PF + (20PF * 0,5) = 18PF$.

A contagem do Ponto de Função será sempre no Início do *Sprint*. Não no *Product Backlog*.

O Valor do PF é combinado em contrato, suponhamos que é R\$ 1.000.

O valor a ser recebido pelo *Sprint* é de 18.000 nesse caso.

Análise/Conclusão: A filosofia do SCRUM é que um segundo *sprint* pode ter 0 PF se visto como um projeto de melhoria. Por outro lado, não raro, um projeto usando SCRUM entrega ao final 200 PF e se cada *sprint* fosse medido e somadas essas medições, teríamos 2.000 PF. Para o objetivo de medição de avaliar e planejar a produtividade, PF é uma medida adequada e, portanto, pode ser usado em cenários com esse objetivo de medição como em contratos.

Se o objetivo de medição é acompanhar o progresso, PF não é uma boa opção. Para entender como é um projeto SCRUM é importante ter vivido um. É iterativo incremental como o RUP, sim. É igual ao RUP, não.
