

Métricas de Pontos de Função

Alunos:

[Marcio Schinkoeth Cruz](#) DRE:096207234

[Sigmar M. de Siqueira](#) DRE:096206335

[Willian Chu Chang](#) DRE:096223442

O que são pontos de função e como são contados?

No final dos anos 70 a IBM sentiu a necessidade de desenvolver uma abordagem independente da linguagem para estimar o esforço no desenvolvimento de software. Um empregado chamado Allan Albrecht foi designado a desenvolver essa abordagem, o que resultou na técnica de pontos de função.

No início dos anos 80, a técnica de pontos de função e um manual de contagem foi produzido pela organização IBM's GUIDE. O Grupo Internacional de Usuários de Pontos de Função (IFPUG) foi fundado no fim dos anos 80. Essa organização produziu seu próprio Manual de Práticas de Contagem. Em 1994, a IFPUG lançou a Versão 4.0 do seu Manual de Prática de Contagem. Enquanto a publicação da GUIDE e cada versão nova das publicações da IFPUG continham refinamentos para a técnica originalmente apresentada por Albrecht, elas sempre pretendiam ser consistentes com seu pensamento original. De fato, ainda está muito próximo disso considerando as duas décadas desde o lançamento da publicação original de Albrecht.

Nos anos 80 e 90, muitas pessoas que sugeriram técnicas de contagem de pontos de função pretendiam substancialmente estender ou completamente repor o trabalho feito por Albrecht.

Pontos de função são a medida do tamanho das aplicações de computadores e os projetos que os constróem. Esse tamanho é medido de um ponto de vista funcional ou usuário. É independente da linguagem do computador, da metodologia de desenvolvimento, da tecnologia ou da capacidade do grupo de desenvolvimento de desenvolver a aplicação. O fato de Albrecht originalmente usar isto para prever o esforço é simplesmente uma consequência do fato de que tamanho é geralmente o guia para o esforço do desenvolvimento. Os pontos de função mediram o tamanho.

É importante salientar o que os pontos de função não medem. Os pontos de função não são medidas perfeitas de esforço para desenvolver uma aplicação ou seu valor de negócio, apesar de que o tamanho em pontos de função é tipicamente um importante fator na medida de cada um. Isso é geralmente ilustrado com uma analogia às medidas em construções. A construção de uma casa de trezentos metros quadrados quase sempre é mais barata do que a construção de uma casa de seiscentos metros quadrados. No entanto, muitos atributos como banheiros de mármore e chão de ladrilho podem fazer a casa menor mais cara. Outros fatores como localização e número de banheiros podem fazer a casa menor custar ter valor maior como residência.

Bill Hufschmidt, um praticante de pontos de função, sempre afirma, em inglês, que as três primeiras letras em pontos de função são 'FUN' (diversão). As pessoas que gostam da contagem de pontos de função e podem justificar isto nesta base deveriam fazer assim pelas seguintes razões:

Medir produtividade - Muitos executivos têm chegado à conclusão de que qualquer que seja o centro do negócio, eles estão também no negócio de software. Calculando muitas variações nos pontos de função produzidos pelo tema mensal indica a eles como eles estão indo neste assunto.

Estimativa de desenvolvimento e suporte - Desde o começo, pontos de função têm sido usados como técnicas de estimativas. Estimativa é obviamente necessária para a análise custo-benefício que

justifica o desenvolvimento da aplicação. Até mesmo para projetos estratégicos que não precisam de justificação quantitativa, a estimativa precisa é necessária para um quadro de funcionários apropriado.

Monitorar acordos de origem externa - Companhias com uma parte significativa de seus requisitos de IS com origem externa se interessam para que entidades externas dêem os níveis de suporte e produtividade que elas prometeram. Entidades de origem externa, como CSC e IBM Global Services frequentemente usam pontos de função para demonstrar que estão de acordo nessas áreas.

Dirigir IS relacionado às decisões de negócios - Companhias precisam analisar seus portfólios de aplicações e projetos. O tamanho em pontos de função é um atributo que precisa ser percorrido para cada aplicação e projeto. Sempre com outros dados, isso vai permitir decisões levando em conta a retenção, retirada e o redesenho de aplicações que possam ser efetuadas.

Normalizar outras medidas - Colocá-las em perspectiva, outras medidas frequentemente precisam do tamanho em pontos de função. Por exemplo, 100 defeitos colocados em um sistema de 100 pontos de função não são boas notícias. Os mesmos 100 defeitos postos em um sistema de 10.000 pontos de função são muito mais fáceis de aceitar.

Pontos de função podem ser usados em sistemas com GUI?

Sim. Quando a técnica de pontos de função foi originalmente desenvolvida, ela não estava aplicada a sistemas baseados na Interface Gráfica com o Usuário (GUI - Graphical User Interface). Não havia sistemas baseados na GUIs! Até a pouco tempo atrás, cerca de três anos, existia um certo desacordo entre praticantes de pontos de função quando levavam em conta esses sistemas.

A dois anos atrás, IFPUG completou a Versão 4.0 do seu Manual de Práticas de Contagem de Pontos de Função. Ele contém as regras oficiais e extensos exemplos para sistemas baseados na GUI. Subseqüentemente, IFPUG produziu um estudo de caso que continha a contagem de um sistema baseado na GUI inteiro.

Infelizmente, ainda é possível encontrar praticantes que não estão familiarizados com as novas regras e que não atualizaram seu material de curso para refletir na tecnologia atual.

Podemos usar pontos de função em sistemas cliente/servidor?

Sim. No caso mais simples, o usuário pode não se dar conta de quando ele está usando um programa sendo executado em um computador central ou uma série de programas rodando em vários computadores através de uma rede. De fato, no mundo cliente/servidor suas aplicações podem travar por causa de uma falha em um computador que você nem sabia que existia. Isso apresenta um forte argumento para que sistemas cliente/servidor possam ser contados assim como qualquer outro. Infelizmente, sistemas cliente/servidor introduzem algumas complicações no processo de contagem.

A contagem para o desenvolvimento de uma infraestrutura técnica é geralmente um desafio. A infraestrutura técnica pode ser composta de gerenciadores de banco de dados, middleware, APIs e outros componentes que são usados pelos desenvolvedores. Instalar esta infraestrutura costuma ser um projeto completamente separado. Existem pontos de funções associados com este projeto? Se sim, a contagem é feita como se os desenvolvedores de aplicações fossem usuários da infraestrutura? É claro, é possível que alguma pessoa do ramo de negócios irá interagir diretamente com essa infraestrutura quando estiver fazendo o desenvolvimento de aplicações para o usuário final. Isso mudará as respostas para qualquer uma das questões precedentes?

Até quando confinamos nossas atenções para os programas de aplicação, a identificação dos limites entre projetos de desenvolvimento de sistemas cliente/servidor pode muitas vezes ser complicada. Em um caso simples, aonde uma aplicação é desenvolvida parcialmente em um cliente e parcialmente em um servidor, o limite inclui ambos. No entanto muitos projetos podem cair para o desenvolvimento de aplicações para servidores que irão interagir com outros clientes pré-existentes e servidores somados a

programas de clientes que estão sendo desenvolvidos como parte do projeto. Nesse caso, limites múltiplos talvez necessitem de ser considerados.

Pontos de função podem ser usados em desenvolvimento orientado a objeto?

Sim. Eles podem ser usados para medir essas aplicações, e se seus tamanhos seriam os mesmos se não fosse utilizada a orientação a objeto. Isso acontece porque a contagem por pontos de função para uma aplicação é independente da tecnologia utilizada para desenvolver a aplicação. No entanto, o uso da orientação a objeto tende a causar impacto na contagem de pontos de função do projeto que constrói a aplicação.

No caso mais simples do desenvolvimento de um software, uma aplicação é construída de um rascunho por um projeto simples. Nesse caso, a aplicação e a contagem de pontos de função do projeto são os mesmos. Por exemplo, construir um sistema de 1000 pontos de função daria um projeto de 1000 pontos de função. Orientação a objeto está rapidamente construindo essa aproximação simples para a construção de uma aplicação uma coisa do passado.

No mínimo, usando técnicas de desenvolvimento orientadas a objeto permite ao desenvolvedor utilizar controles de edição em Windows pré-existentes. Quando a orientação a objetos é completamente acolhida, os desenvolvedores podem incorporar pedaços maiores de funcionalidade nas suas aplicações. Por exemplo, a Automação OLE permite que uma aplicação segure a força do Microsoft Project 4.0. A contagem nessas situações requerem bom julgamento em engenharia e experiência.

Uma escola de pensamento foi simplesmente incorporada a todas as funcionalidades na contagem de aplicação e projetos. Isso faz sentido para funcionalidades menores que não seriam reconhecidas separadamente pelo usuário de qualquer forma, como controles de edição. No entanto, isso é geralmente um problema para componentes mais ricos em funções. Se o usuário percebe que muita da funcionalidade que é providenciada na verdade vem de uma aplicação adquirida, irá ele querer dar ao desenvolvedor "crédito" por toda aquela funcionalidade? Se o desenvolvedor clama toda a funcionalidade de ambas as porções escritas pessoais e as porções adquiridas do projeto, a técnica de pontos de função será utilizável para essa estimativa?

Deve-se salientar que essas considerações têm existido desde antes do desenvolvimento da orientação a objeto. No entanto, anos atrás esses eram casos excepcionais. A comunidade de pontos de função ainda precisa codificar sua abordagem para lidar com essas situações.

O IFPUG publicou um estudo de caso mostrando a contagem para uma aplicação que foi desenvolvida utilizando técnicas de orientação a objeto.

Como o uso de pontos de função é comparado ao uso de linhas de código?

O número de linhas de código é o método tradicional de avaliar o tamanho de uma aplicação. Algumas pessoas dizem que esse método ainda é relevante porque isso mede o que os desenvolvedores de software atualmente fazem: escrever linhas de código. Eles irão utilizar linhas de código mesmo ao invés de ou somado aos pontos de função. Outros praticantes dizem que linhas de código são irrelevantes. Caper Jones declarou "o uso de métricas por linhas de código para estudos de produtividade e qualidade para serem levados em conta como más práticas profissionais desde 1995".

No melhor, a contagem de linhas de código medem softwares do ponto de vista do desenvolvedor. Na medida que pontos de função são baseados em telas, relatórios e outros objetos externos, essa medida leva apenas a visão do usuário. Nesses dias de entidades de origem externa e outras confusões levando em conta as regras do IT em uma organização, o entendimento da visão do usuário é de importância crítica.

Relacionado às essas diferentes visões está o fenômeno de que você tem o que você mede.

Quando o número de linhas de código é a sua medida primária de produtividade, você geralmente pega isso pelo fato dos desenvolvedores utilizarem linguagens menos poderosas e ignorarem oportunidades de reuso. Quando o número de pontos de função é a sua medida primária de produtividade, você tende a ter sua produção de funcionalidade aumentada pelos desenvolvedores. É claro que é do usuário a responsabilidade de avaliar quando essa funcionalidade está retornando valores de negócios maiores.

Existem outros problemas técnicos com a medida por linhas de código. A IBM encorajou o desenvolvimento por pontos de função porque estava começando a ver ambientes de desenvolvimento heterogêneos aonde BAL, COBOL e PL/I estavam sendo usados. Agora quase toda oficina está usando uma mistura de linguagens. Como você compara um projeto que tem 100.000 linhas misturadas de C++, SQL e Visual Basic com uma que tem 100.000 linhas de COBOL?

E mais, não existe padrão definido para o que é uma linha de código. Linhas em branco contam? Comentários contam? Dados declarados estão incluídos? O que acontece se uma sentença se estende a várias linhas? Organizações como o Instituto de Engenharia de Software têm publicado alguns guias numa tentativa de padronizar a contagem, mas nenhuma organização seguiu a mesma regra para linhas de código que a IFPUG tem para pontos de função.

O que são pontos característicos (feature points)

Por algum tempo, praticantes de pontos de função reconheceram que algumas classes de aplicações não beneficiavam-se da análise por pontos de função o quanto era esperado. Essas classes de aplicações incluíam controle de processos em tempo real, otimização matemática e sistemas fixos (embedded) variados. Em 1986 a Software Productivity Research (SPR) desenvolveu pontos característicos.

Essa técnica considera o número de algoritmos usados na aplicações e levemente modifica alguns dos pesos constituintes do tradicional ponto de função. Ela foi desenhada de tal maneira que típicas aplicações de negócios, uma área aonde pontos de função tem sido aplicada com sucesso, tenham o mesmo tamanho tanto quando calculadas por pontos de função quanto por pontos característicos. As classes com problemas de aplicações mencionadas acima irão tipicamente ter maiores índices quando medidos os pontos característicos do que com pontos de função. Isso compensa pelo fato da produtividade geralmente parecer estar abaixo para essas aplicações. É claro, pontos característicos são ainda experimentais.

Pontos característicos não estão no mesmo nível de aceitação que os pontos de função estão. Não existe nenhuma organização de padrões como o IFPUG. No entanto, um requerimento de um padrão é que este esteja disponível para as pessoas estudarem e usarem. Caper Jones documentou a técnica em seu livro *Applied Software Measurement*. SPR faz uma descrição da técnica disponível na internet em um artigo chamado ["What Are Function Points?"](#). Esse artigo descreve ambos pontos de função e característicos, e discute como escolher o método apropriado para usar em uma determinada aplicação.

O que são pontos de função Mk II?

Nolan, Norton & Co., parte da KPMG Management Consulting, contrataram Charles Symons em 1984 para aconselhar clientes sobre o método apropriado para melhorar a performance do desenvolvimento de seus sistemas. No período em que fez isso, ele clama por ter descoberto um ponto fraco na abordagem de Albrecht para as análises de pontos de função e desenvolveu a abordagem MK II para substituí-las. Por volta de 1987 este se tornou um produto licenciado e é muito vendido.

Symons diz que a abordagem de Albrecht sofre das seguintes fraquezas:

- É geralmente difícil identificar os componentes de uma aplicação. Por exemplo, o que é um arquivo lógico? Em 1984 não havia guia suficiente para fazer essas determinações.

- Albrecht determinou pesos para os pontos de função baseados em "discussão e tentativa".
- As duas críticas acima foram também niveladas na identificação e determinação dos pesos dos Fatores de Ajustamento de Valores.
- Albrecht não providenciou meios de justificar a complexidade interna. Isso é o mesmo problema levando em conta algoritmos cuja técnica de pontos característicos foi desenvolvida para endereçar.
- Quando sistemas pequenos são combinados em aplicações grandes, a abordagem de Albrecht faz a contagem de pontos de função total ser menor do que a soma dos componentes.

A diferença conceitual principal entre os dois métodos é o cálculo do Tamanho do Processamento da Informação, que corresponde aos Unadjusted Function Points (UFP) de Albrecht. Symons decompõe a aplicação que está sendo contada em uma coleção de transações lógicas. Cada transação consiste em uma entrada, um processo e uma saída. Para cada transação, os UFP se tornou uma função do número de entradas, tipos-elementos de dados, tipos-entidades referenciadas e tipos-elementos de dados de saída. Os UFP para o sistema inteiro são então somados.

Depreciadores do trabalho de Symons dizem que Mk II é simplesmente uma maneira de exagerar o valor de sistemas monolíticos que os clientes de Nolan, Norton & Co.'s estavam construindo na época. Eles consideraram isso uma concessão política que irá minar a competitividade dos clientes durante a corrida. Críticos mais moderados acreditam que Symon identificou algumas questões dos pontos de funções que os praticantes já tinham conhecimento. No entanto, ele escolheu lidar com eles de uma forma que levaria a um proprietário de um produto Nolan, Norton & Co., ao invés de reforçar o trabalho inicial de Albrecht.

Grant Rule forneceu informação no estado atual do Mk II. Ele relatou que a técnica tem sido de domínio público desde 1991. A autoria do design é da United Kingdom Function Point Users Group, 147 Knowsley Road, Norwich, Norfolk, NR3_4PT, United Kingdom; telephone/fax: +44-1603-219-263.

Ele foi até um estado em que Mk II era muito utilizado no UK, e aonde seu uso estava crescendo, como na Índia, Singapura, Hong Kong e Europa. Mais ou menos 50% dos membros do UKFPUG usavam o Mk II. Usuários incluem muitas organizações governamentais, financeiras, seguradoras, varejo e manufatureiras.

O que são pontos de função 3D?

Entre 1989 e 1992, a Boeing Company considerou o uso de pontos de função para medir a produtividade. Um documento interno da Boeing Company foi publicado nos procedimentos da Pacific Northwest Software Quality Conference em 1992. Ele definia uma técnica chamada pontos de função 3D.

A nova técnica foi desenhada para endereçar dois problemas clássicos associados com a abordagem de Albrecht. Primeiro, a abordagem é considerada difícil de usar. Segundo, ela não mede propriamente sistemas científicos e de tempo real.

As três dimensões em pontos de função 3D são dados, funções e controle. A dimensão de dados é similar ao de pontos de função de Albrecht. A dimensão função adiciona transformações, que são similares aos algoritmos. A dimensão controle adiciona transições, que enumeram mudanças no estado da aplicação. Foi chamado pontos de função em escala decrescente 3D para os níveis de classes usadas em conjunto com desenvolvimento orientado a objeto.

É discutível quando pontos de função 3D são mais fáceis de contar do que pontos de função tradicionais. Também é discutível quando a técnica endereça trabalhos científicos melhor do que pontos característicos. É um fato que pontos de função 3D não são utilizados vastamente hoje.

Quando é possível contar pontos de função?

Como votar, você deve contar mais cedo e com frequência. Quanto mais cedo for quantificado o que um projeto está entregando, mais cedo ele estará sob controle. Tradicionalmente, muitos praticantes acreditam que a contagem de pontos de função não podia ocorrer até que o design do produto esteja completo. Membros do IFPUG perceberam a importância de poderem contar pontos de função mais cedo no ciclo de vida do desenvolvimento. Na versão 4.0 do IFPUG, existem regras e guias que tornam possível contar pontos de função uma vez que os requerimentos estejam finalizados. Muitos praticantes usam heurísticas que os permitem estimar o tamanho do ponto de função e a contagem que pode ocorrer em vários pontos no ciclo de vida. Na medida que todos usam diferentes ciclos de vida, os estágios irão ser descritos em termos do progresso do projeto em liberações relatadas.

Se o ciclo de vida começa com alguma coisa como um estudo possível, é geralmente impossível contar pontos de função a essa altura. No entanto, o número de pontos de função pode ser estimado usando a mesma técnica atualmente utilizada para estimar linhas de código ou esforço. Se um projeto similar tem 2.000 pontos de função, o melhor palpite a essa altura é que esse projeto também terá 2.000 pontos de função.

Durante a aquisição dos requerimentos a estimativa do tamanho em pontos de função pode ser refinada continuamente. Para muitos projetos um modelo de dados lógico é desenvolvido. A definição das pessoas de um modelo de dados lógico varia, mas no mesmo ponto esse modelo não é atribuído nem na sua terceira forma normal. No entanto, nesse ponto o analista pode começar a identificar quais entidades irão precisar do processo para adicionar, mudar, deletar e mostrá-los. Se o equivalente de um diagrama de contexto Yourdon foi produzido, então as interações com o usuário e com sistemas externos estão especificados. Nesse ponto, uma boa estimativa do tamanho dos pontos de função pode ser feita assumindo que as transações identificadas são todas de complexidade média. Até nesse ponto do ciclo de vida, é geralmente possível prever os Fatores de Ajuste de Valores com uma acurácia justa.

Uma vez que os requerimentos do negócio foram completados, é possível contar com acurácia os pontos de função para a aplicação. Neste ponto, o modelo de dados lógicos deve ser completamente atribuído. Todas as interações do sistema, incluindo telas e relatórios, devem ser identificados e os itens de dados associados com eles especificados. Não há necessidade para que as telas ou os relatórios serem formatados nessa hora.

Como já foi dito antes, seguindo um design externo ou design de produto é possível alcançar uma acurácia para a contagem dos pontos de função. É claro, você deve ter tido uma contagem precisa ao completar os requerimentos. Agora é hora de começar a considerar a volatilidade dos requerimentos e o impacto no projeto. O tipo mais simples de volatilidade de requerimentos é avançar lentamente pelo escopo. Se, no final dos requerimentos, as medidas mostraram que o projeto tem 1.000 pontos de função e depois do design do produto estar completo, as medidas mostram 1.500 pontos de função, você tem pelo menos 50% do escopo pesquisado lentamente.

Outras formas de volatilidade dos requerimentos podem ser mais perigosos sem serem notados. Por exemplo, você pode medir 1.000 pontos de função no fim dos requerimentos e 1.000 pontos de função no fim do design do projeto. Infelizmente, muitos desses pontos de função tardios são funções que são completamente diferentes das que foram identificadas nos requerimentos. Esses problemas devem ser pesquisados. Quanto mais profundo durante o ciclo de vida isso ocorrer, mais cara essa mudança pode se tornar mais tarde. No entanto, desde que o tamanho dos pontos de função não tenha mudado, o impacto dessas mudanças é geralmente ignorado. Essas mudanças precisam ser pesquisadas como pontos de função aprimorados. Nesse caminho, os impactos podem ser propriamente considerados quando o projeto precisar ser estimado novamente.

Na teoria, não deveria haver mudanças na contagem de pontos de funções entre o fim do design do projeto e o fim do teste de aceitação. É claro, em teoria, não existe diferença entre a teoria e prática; na prática, existe uma diferença muito grande. É durante o tempo de implementação e teste que as mudanças se tornam progressivamente mais caras de fazer. Algumas vezes os usuários e os gerentes de projeto trocam os requerimentos, características, custos e prazos durante esse tempo. Pontos de função podem ser usados para quantificar essas negociações. No entanto, elas devem ser usadas propriamente. Não é sábio trocar 100 melhorias de pontos de função por uma redução de 100 pontos de função na funcionabilidade do sistema. O trabalho já expandido aos 100 pontos de função a serem postos devem também ser considerados. Mais uma vez, usando o aperfeiçoamento dos pontos de função pode fazer esse processo mais fácil para todos os participantes.

Faz sentido atualizar a contagem de pontos de função no final do projeto. O projeto deve ser revisto com a idéia de fazer estimativas de projetos futuros mais precisas. O tamanho das aplicações também é um fator que determina o nível de requerimento do suporte. A comunidade de usuários precisa decidir quando o valor do negócio associado com uma aplicação vale o esforço que será requisitado para suportar isso.

O que fazer com as contas do ponto de função?

Esta pergunta deve ser feita e respondida antes da primeira conta do ponto de função. Aqui estão algumas das respostas:

- para medir a produtividade de sua equipe, seu outsourcer ou até mesmo você
- para calcular o esforço de projeto e o tempo gasto;
- para medir sua produtividade e comparar os resultados com outras organizações
- para usar os dados com o intuito de guiar para uma melhor decisão. Decisões empresariais como reter, aposentar ou redesenhar as aplicações.
- Se você enquadra em um ou mais resposta, então você precisa de algum tipo de programa de métrica em sua organização.

Como você mede produtividade com pontos de função?

Devemos ter mais atenção em dois tipos de métricas de produtividades: pontos de função implementados por pessoa/mês e calendario/mês. Ao medir atividades de manutenção de software, pontos de função de realce podem ser substituídas nesta mesma métrica. O número de pontos de função suportadas por pessoas membros é facilmente achado.

Pontos de função implementados por pessoa/mês indicam a quantidade de esforço que será necessária para o desenvolvimento de um sistema. Por exemplo: Uma aplicação que possui 500 ponto de função , desenvolvida utilizando 10 pessoas em 10 meses, obtendo uma taxa de produtividade de 5 pontos de função por pessoa/mês.

Frequentes perguntas devem ser consideram, como: que pessoas devemos contar?. Se os usuários

participarem no processo de construção, então o tempo dessas pessoas contam? Quando calculamos um departamento IS inteiro, os executivo e o pessoal administrativo contam?

Pontos de função implementados por calendário/mês indicam a velocidade com que o sistema pode ser desenvolvido. A velocidade não aumenta simplesmente, caso adicione novas pessoas, pois a adição de pessoas, geralmente, não traz nenhum impacto positivo no prazo de finalização. Organizações que possuem um grande time de desenvolvedores para um curto prazo de entrega, possui muito mais do que pessoas, possui um elevado numero de pontos de funcao. No exemplo acima, foram desenvolvidos 50 pontos de função por calendario/mês.

Como você calcula com pontos de função?

Há basicamente dois métodos para transformar contas de ponto de função em estimativas de desenvolvimento. O primeiro é assumir um certo nível de produtividade em pontos de função por pessoa/mês e calendario/mês. O segundo é converter a conta de ponto de função em um número equivalente de linhas de código e usar um macro modelo de estimativa igual ao COCOMO.

Este método posterior permite uma variedade de outros fatores, além de pontos de função que são considerados na estimativa. Alguns destes fatores são considerados na próxima pergunta.

Capers Jones fez um trabalho que é aplicável a ambas as aproximações. O livro dele Applied Software Measurement dá taxas de produtividade comuns para aplicações comercial, MIS e militares.

No mesmo livro, Jones dá tabelas que indicam quantas declarações de código são necessárias para implementar um ponto de função em várias linguagens diferentes. Como foi em "What is Backfiring?," muitos fazem a suposição que declarações e linhas de código são equivalentes. Então o resultado, pode ser inserido em qualquer modelo de macro, incluindo um de domínio público COCOMO. Um produto comercial, Before You Leap, provê isto em um único produto integrado

. Categorizando um projeto simplesmente como comercial, MIS ou exército não é suficiente. Até mesmo entre projetos comerciais, podem haver grandes variações em produtividade por causa dos vários atributos do projeto. Estes incluem tamanho de projeto, ferramentas de desenvolvimento e outros fatores. Um projeto verdadeiramente similar deve ser achado em ordem para adquirir taxas de produtividade que garanta o uso com confiança.

Há áreas de debate relativo à segunda aproximação. Uma dessas areas é se fundar no total de linha derivada no total de ponto de função não adaptada ou usar o total depois que for ajustado pelo Fator de Ajuste de Valor. Em um dos primeiros trabalhos em COCOMO2, Barry Boehm, afirma com convicção resultados bons são gerados usando contas não adaptadas. Porém, outros têm usado conta ajustado.

Estas mesmas aproximações são válidas para calcular atividades de manutenção. Porém, há duas razões principais por que estas estimativas podem ser mais difíceis de chegar . Primeiro, atividades de manutenção tendem a ser menor em tamanho. Sempre é perigoso usar um modelo para predizer projetos pequenos. O tempo é muito dependente de uma pessoa ou uma pequena equipe que faz o trabalho. Super-homens (e super-mulheres) existe na comunidade de desenvolvimento de software. Porém, ninguém parece ser hábil para coordenar as atividades de 30 super pessoas .

O segundo desafio da estimativa de manutenção é a natureza do ponto de função de realce. Adicionando uns 100 pontos de função de funcionalidade ou apagando 100 pontos de função resultam em custo de 100 pontos de função. Porém, a estimativa de esforço seria bastante diferente. Isto deve ser considerado ao usar esta técnica.

Além de pontos de função, quais fatores influênciam uma estimativa?

O esforço e horário exigidos para desenvolver ou melhorar um produto de software são influenciados por vários fatores além do tamanho do produto. Muitos destes fatores são discutidos por Barry Boehm junto com seu novo modelo COCOMO2. Eles podem ser resumidos em quatro categorias:

1. outros fatores do Produto - Há fatores do produto que não são refletidos no tamanho. Estes podem causar impacto no custo de desenvolvimento do produto. Eles também alteram o valor do produto.

- Precedentedness é uma medida da experiência das organizações com sistemas similares. Flexibilidade mede a habilidade das organizações para acomodar requisitos com o intuito de facilitar o esforço do desenvolvimento de software. Trabalhos são mais difíceis quando uma aplicação é moderna e os requisitos são inalteráveis.

- Confiabilidade de requisitos é relacionada com o impacto a falha do sistema. Banco de dados aumenta a complexidade do produto. Há outros fatores de complexidade do produto.

- A documentação de desenvolvimento do sistema pode variar de insuficiente para as necessidades da fases de ciclo de vida até muito excessivo para as necessidades do ciclo de vida. No ultimo caso, a documentação se torna parte do produto. Além de software em desenvolvimento, a equipe está desenvolvendo algum tipo de material de leitura para outras pessoas.

2. Dificuldade de plataforma - Plataformas com limitações de memória, armazenamento e cpu fazem o trabalho mais custoso.

Plataforma, ou Máquina Virtual, refere-se ao complexo de hardware e software que a aplicação usa. Inclui os compiladores, DBMSs e qualquer outro componente. Normalmente, há uma grande mudança duas vezes por ano. Estabilidade de Máquina virtual pode cortar o esforço de desenvolvimento em até 13%. Ambientes que experimentam grandes mudanças a cada duas semanas podem esperar um aumento de 30% no esforço de desenvolvimento.

3. Fatores pessoais - Estes tendem a ser o segundo grupo de maior custo. Diversos livros foram escritos falando sobre isto. Os principais fatores pessoais serão apresentados. · Team Cohesion é normalmente presente quando stakeholders compartilham os mesmos objetivos e cultura. Porém, diferenças podem ser superadas quando stakeholders estiverem dispostos a reconciliar objetivos. A experiencia e familiaridade dos stakeholders no comando de uma equipe também é relevante.

Mudança de pessoal 12% por ano em um projeto é considerados normal. O valor pode variar entre 3% e 48%.

- Experiência em Aplicações é o principal requisito que os analistas devem ter com o domínio do problema para o qual um sistema está sendo construído. Pode levar 10 anos para alguém se tornar um perito na área empresarial. A maioria dos analistas têm muito menos experiência do que isto.

- Experiência da equipe com a Plataforma, Linguagem e Experiência com Ferramenta têm impacto na estimativa. Por causa da velocidade da mudança tecnológica, a maioria dos profissionais tem apenas um ano de experiência com as ferramentas que estão sendo usadas para um determinado projeto. Pode levar três anos para dominar qualquer linguagem de computador específica ou se especializar com ferramentas de desenvolvimento de software.

4. outros Fatores de Projeto - O projeto é igual ao produto mais a plataforma da equipe mais o processo. · Architecture/Risk Resolution envolve a habilidade dos melhores projetistas. A presença de risco de administração também é considerada. Em caso normal, serão dedicados 17% do desenvolvimento a arquitetura (projeção).

- Maturidade do Processo (Process Maturity) é um novo guia de custo. O procedimento para determinar esse novo guia de custo é organizado em relação as 18 Áreas de Processo Chave do SEI Capability Maturity Model. Teoricamente, organizações mais maduras são mais eficientes.

- Schedule já foi um guia de custo para COCOMO81 e continua sendo uma área comum de dificuldade. Se um projeto é planejado para 10 pessoas durante um ano, então é necessário 120 pessoas meses de esforço. Se uma nova decisão é tomada. Sendo necessarias agora 9 meses para desenvolver uma aplicação, de quantas pessoas são necessárias?. Algumas pessoas poderiam dizer que 14 (120/9) pessoas seriam o bastante. Porém, tal diminuição do Schedule aumentaria o esforço em 22%. Se o esforço fosse aumentado em 22%, então o projeto deveria requerer 16 (120*1.22/9) pessoas.

O que é benchmarking?

Benchmarking ficaram populares como parte de Total Quality Management (TQM). Os empregados de uma companhia visitariam outras companhias e comparariam vários processos similares com as suas. Algumas pessoas sentiam que estas interações davam valiosos discernimento que os ajudariam a aperfeiçoar os seus processos empresariais.

Como a maioria das companhias estão envolvidas em desenvolvimento de software, benchmark se tornou um processo natural. Além disso, técnicas como análises de ponto de função permitem as pessoas fazerem comparações quantitativas entre companhias.

As vantagens destas comparações são óbvias. Departamentos IS podem adquirir um visão avançada sobre o impacto de se

mover para novas tecnologias, definindo os possíveis problemas que possam surgir em seus processos.

Várias organizações que estão envolvidas no benchmarking IS. A maioria deles fazem algum uso de pontos de função. As melhores organizações conhecidas são:

- IFPUG recentemente começou um serviço de benchmarking para seus sócios. É modelado depois de um programa australiano que teve êxito por algum tempo. No futuro, esta pode se tornar uma valiosa fonte de informação.
- Compass América, Inc e Reais Decisões Corp. são uns dos mais conhecidos benchmark vendidos na América.
- CSC Research and Advisory Services Performance Enhancement Programme (PEP) é outro serviço que tenta usar dados comparativos para acelerar o processo de melhoria IS

Conclusão

Há muitos usos para pontos de função além de estimar o cronograma, esforço e custo. Muitos gerentes de projeto não acreditam que os pontos de função possuem qualquer utilidade. De certa forma, eles estão certos. Muitas organizações estão utilizando pontos de função e métricas de software para reportar tendências a nível organizacional. Muitas equipes de projeto enviam dados a um grupo central de métricas e nunca mais tornam a ver seus dados. Isto é análogo a enviar os dados a um buraco negro. Se os gerentes de projetos começarem a entender como os pontos de função podem ser utilizados para estimar casos de teste, calcular custos de manutenção e assim por diante, eles provavelmente investirão na contagem de pontos de função.