

Aplicando a Análise de Pontos de Função à Completeza dos Requisitos¹

Carol Dekkers
Quality Plus Technologies, Inc.
Mauricio Aguiar
TI Métricas Ltda.

As questões referentes aos requisitos estão sempre presentes no desenvolvimento de sistemas, apesar da existência de muitos modelos e métodos destinados a verificar a completeza dos mesmos. Este artigo mostra como a Análise de Pontos de Função (APF), a técnica de dimensionamento de software, fornece valor como uma revisão estruturada dos requisitos. Embora historicamente sua utilização tenha sido limitada ao dimensionamento de software, a APF vem ganhando popularidade como um método estruturado e útil para revisar requisitos. Ao ser utilizada no desenvolvimento de sistemas para avaliar a completeza dos requisitos, a APF fornece mais do que números representativos do tamanho do software – a documentação gerada pela APF reflete o conjunto completo de requisitos funcionais do usuário.

Há uma grande variedade de abordagens para a definição de requisitos de software, incluindo sessões JAD, técnicas de gerenciamento de requisitos, prototipagem, desenvolvimento rápido de aplicações, *eXtreme programming* e outras. Quando adequadamente executadas, tais técnicas normalmente apresentam como resultado uma forma de documentação dos requisitos dos usuários. Depois de uma série de revisões dos usuários e de *peer reviews*, esses requisitos *formais* são normalmente assumidos como representantes do conjunto total de requisitos dos usuários.

No entanto, o conjunto completo de requisitos do usuário normalmente não estará completo até o final do projeto, continuando a emergir conforme o mesmo progride. Como resultado o projeto enfrentará retrabalho, deslizos de cronograma e estouros de orçamento, a intensidade dos quais dependerá do desconhecimento original dos requisitos. Em alguns projetos do Departamento da Defesa norte-americano este problema é agravado por requisitos baseados em resultados ou desempenho, de modo que requisitos funcionais são criados como parte do processo de *design* do software. Tais projetos, quando surgem, desafiam as abordagens tradicionais para o tratamento dos requisitos. Por exemplo, como são documentados os requisitos quando o requisito de performance é lançar um projétil de uma distância de 200 milhas com um impacto X? Aqui não pretendemos resolver este tipo de problema de requisito.

Este artigo aborda projetos nos quais os requisitos são articulados (ou deveriam ser), delineando como a Análise de Pontos de Função (APF) pode ser uma ferramenta adicional para identificar requisitos omitidos, avaliar a completeza dos requisitos e descobrir defeitos potenciais. Nossa experiência mostra que a APF é frequentemente mais eficaz do que walkthroughs com usuários ou *peer reviews* para identificar o conjunto completo de requisitos funcionais e descobrir defeitos potenciais. De fato, os benefícios obtidos por meio da aplicação da APF aos requisitos funcionais dos usuários podem ser mais valiosos do que o mero tamanho funcional do software.

¹ Adaptado/atualizado pelo segundo autor, a partir do artigo originalmente publicado em Crosstalk em fevereiro de 2001, sob o título “Applying Function Point Analysis to Requirements Completeness”. Também anteriormente publicado sob o título “Double Duty Metrics: Using Functional Sizing to Gauge Requirements Completeness”, por Carol Dekkers e Mauricio Aguiar, Cutter IT Journal, Vol. 13, No. 5, maio de 2000.

Este artigo destina-se a dois tipos de público:

1. Equipes de desenvolvimento que já utilizam, ou estão pensando em utilizar a APF em seus projetos. As informações aqui fornecidas visam melhorar a razão custo-benefício da APF, explorando seu uso como uma verificação de completeza dos requisitos.
2. Equipes de desenvolvimento que não utilizam a APF, mas que gostariam de obter ferramentas adicionais para aumentar a eficácia de seu processo de requisitos. Os conceitos apresentados neste artigo podem ser aplicados seletivamente a qualquer projeto, sem necessidade de incluir todos os passos do método.

Requisitos

Por que será que o conjunto *correto* de requisitos, isto é, completo e corretamente representado, é tão difícil de ser obtido em nossa indústria? Dentre várias razões, o problema consiste em obter corretamente os requisitos, obter os requisitos corretos (o conjunto completo de requisitos funcionais do usuário), frequentemente exigindo que se obtenha mais do que as especificações dos requisitos. Um dos maiores problemas enfrentados pelos desenvolvedores é ser capaz de avaliar quando os requisitos estão completos para então começar o *design* e a codificação do sistema.

Antes de discutir como aplicar a APF à completeza dos requisitos, vale a pena identificar os três grandes tipos de requisitos de software. Em conjunto, esses três tipos de requisitos abrangem todos os requisitos dos usuários. O primeiro grupo é constituído pelos **requisitos funcionais** dos usuários, que são as funções lógicas do negócio ou do usuário que o software precisa executar. Qualquer tipo de software, desde sistemas direcionadores de mísseis em tempo real até sistemas de contabilidade possuem requisitos funcionais que precisam ser executados. Tais incluem processos elementares que precisam ser suportados para receber a entrada, processar, manipular, dar saída e repassar dados de e para o software. A APF contempla especificamente esses tipos de requisitos.

Em segundo lugar estão os **requisitos não-funcionais** dos usuários. Essas são as restrições, independentes de tecnologia, que o negócio precisa satisfazer. Os requisitos não-funcionais incluem requisitos de qualidade e performance tais como portabilidade, usabilidade, segurança, dependabilidade, confiabilidade e velocidade. Parte da técnica de Pontos de Função pode ser útil para lidar com esses requisitos.

Finalmente, os **requisitos técnicos** são os requisitos do usuário relativos a uma configuração de hardware/software específica, ou a uma configuração técnica específica que deve ser satisfeita. Como exemplo, os requisitos técnicos podem exigir um banco de dados Oracle ou uma solução de hardware com várias camadas. Embora estas especificações sejam tão importantes quanto as demais classes, a APF não contempla este tipo de requisito.

O restante deste artigo refere-se especificamente a requisitos funcionais e não-funcionais.

Verificações de Completeza Tradicionais

Ainda que se tenha como meta que os requisitos funcionais e não-funcionais devam ser não-ambíguos, corretos e completos, é fácil escrever regras e checar com um usuário a

ambiguidade e correção dos requisitos. O problema, no entanto, é garantir que o conjunto completo de requisitos funcionais tenha sido identificado. São necessárias uma ou duas estruturas de referência para isso. Tal estrutura é normalmente obtida por meio de duas técnicas existentes:

1. **Modelo Baseado em Teoria.** Uma estrutura de referência baseada em teoria pode ser utilizada, tal como análise estruturada, engenharia da informação, ou modelagem de dados. O analista irá decompor o problema e procurar estruturas abstratas tais como fluxos de dados, processos e depósitos de dados. Os requisitos serão considerados completos quando as estruturas abstratas fizerem sentido para o analista, por exemplo, quando os depósitos de dados tiverem fluxos de dados entrando e saindo dos mesmos.
2. **Experiência Pessoal.** O analista pode ter trabalhado com outros sistemas semelhantes ao que está sendo analisado. Nesse caso, ele ou ela possuirá uma estrutura de referência subjetiva composta por todas as estruturas e regras de negócio anteriormente encontradas. O analista irá decompor o problema e procurar estruturas e regras conhecidas, em conformidade com seu modelo da realidade. Os requisitos serão considerados completos quando as estruturas identificadas corresponderem ao modelo de completude subjetivo do analista, por exemplo, contas a receber vencidas terão que ser classificadas como recebidas ou como pendentes.

Normalmente o analista trabalhará com uma mistura das duas estruturas de referência, a fim de aumentar a clareza e qualidade do conjunto documentado de requisitos conhecidos e, ainda, aumentar o percentual de requisitos conhecidos em relação ao total de requisitos. Ao longo do projeto ele ou ela irá integrar sua experiência pessoal ao conhecimento teórico. No entanto, cada vez mais isto será insuficiente para obter a necessária cobertura de completude. Interessa ao analista utilizar tantas estruturas de referência quanto for possível.

Idealmente, as estruturas de referência deveriam ser ortogonais, isto é, não deveriam se sobrepor. Cada estrutura de referência deveria prover informações únicas, não disponíveis nos outros modelos.

Por Que a Análise de Pontos de Função?

A Análise de Pontos de Função oferece uma estrutura de referência adicional para verificar a completude dos requisitos funcionais e de alguns não-funcionais. A APF é diferente das duas primeiras estruturas de referência mencionadas porque provê uma perspectiva única, centrada na visão do usuário. A APF examina o conjunto de requisitos funcionais do usuário em termos de dados e movimentos/manipulações (transações), conforme entendidos e expressos pelos usuários; com base nisso, a APF determina o tamanho funcional do software. Dessa forma, a APF pode ser utilizada em acréscimo aos modelos baseados em teoria e experiência pessoal anteriormente mencionados, a fim de garantir que os requisitos funcionais estejam completos.

Fundamentos de Pontos de Função

Os Pontos de Função (PF) medem o tamanho de um projeto de software considerando a funcionalidade lógica percebida pelo usuário, ao invés de medir a implementação física dessas funções, a qual pode ser medida em Linhas de Código (LOC – *Lines of Code*). A APF examina os requisitos funcionais do usuário a serem suportados ou entregues pelo software. Em seguida, atribui um número ponderado de PF a cada função lógica do

usuário, conforme estabelecido no Manual de Práticas de Contagem de Pontos de Função, calculando em seguida o tamanho do software em PF.

Em termos simples, a APF mede o que o software deve fazer de um ponto de vista externo, do usuário, independentemente de como o software seja construído. Embora analogias com outras indústrias tais como a de construção e de manufatura tentem descrever como a APF trabalha com o software, nenhuma consegue um ajuste perfeito. Em termos básicos, a APF reflete o tamanho funcional do software, independentemente da linguagem de desenvolvimento e da implementação física.

Os PF podem ser comparados à área funcional de um prédio, conforme obtida somando-se as áreas da planta baixa. Os PF quantificam os requisitos funcionais do usuário (planta baixa), através da soma de seus *componentes* funcionais. Conforme ocorre na construção civil, o gerenciamento do projeto não é possível se for conhecida apenas a área em metros quadrados. O desenvolvimento de sistemas não pode ser gerenciado com base apenas no tamanho em PF.

Utilizando a APF para Avaliar a Completeza dos Requisitos

Consulte a revista Crosstalk de fevereiro de 1999 para ler um artigo introdutório sobre APF (<http://stsc.hill.af.mil/crosstalk>). Durante a execução de uma contagem de PF, todos os requisitos funcionais do software são analisados, ponderados e contados com base no método de identificação padrão. É durante a análise dos requisitos funcionais do usuário que a maioria dos erros e omissões nos requisitos são descobertos, conforme descrito abaixo. A seguir, os passos do processo de contagem de PF:

1. Determinar o escopo do projeto e o objetivo da contagem de PF. Por exemplo, os PF podem ser contados para quantificar o tamanho de um projeto de novo desenvolvimento ou de melhoria/renovação, ou para medir o tamanho de uma aplicação existente.

Neste passo é proveitoso registrar o título e a data dos documentos-fonte utilizados como base para a contagem (p.ex.: Documento de Requisitos do Sistema ABC, V1, 22 de março de 2009). Isto provê rastreabilidade para as funções lógicas incluídas nos requisitos funcionais em um momento específico no tempo, sendo útil para avaliar o aumento descontrolado do escopo (*scope creep*) do projeto. Também pode contribuir para a base histórica que irá ajudar a avaliar projetos futuros, conforme mostrado abaixo.

Através da documentação do escopo do projeto e objetivo da contagem de PF, ainda que expressa apenas em algumas linhas, as suposições do projeto são esclarecidas e esquecimentos nos requisitos podem ser identificados. Por exemplo, se o objetivo da contagem de PF for medir a quantidade de customização necessária à implementação de um pacote adquirido no mercado (COTS – *Commercial-Off-The-Shelf*), o escopo deverá incluir apenas as funções customizadas e não o pacote inteiro. Isto oferece uma idéia do que estará incluído no projeto.

2. Identificar a fronteira lógica da aplicação. Este passo identifica as funções que o software deverá executar, assim como interfaces com os usuários externos, departamentos e outras aplicações. A fronteira da aplicação para fins da contagem de pontos de função não coincide com a fronteira física da aplicação. Ao invés disso, aquela fronteira é a fronteira lógica que envelopa as funções auto-contidas do usuário, que precisam existir para atender os respectivos requisitos. Esta fronteira separa o software do domínio dos usuários (os usuários podem ser pessoas, coisas, outras aplicações de

software, departamentos e outras organizações). O software pode se espalhar por diversas plataformas físicas e incluir processos on-line e batch (por lote) – todos eles incluídos na fronteira lógica da aplicação. Como exemplo, um sistema de contas a pagar seria normalmente considerado como uma aplicação na APF, mesmo que possa residir em múltiplas plataformas de hardware em sua instalação física.

Como cada aplicação ou sistema de software possui uma fronteira de aplicação separada (isto é, contas a pagar seria normalmente uma aplicação, ativo fixo uma outra aplicação), o processo de medição funcional de software frequentemente inclui um diagrama de contexto do projeto, constituído por vários círculos representando várias fronteiras de aplicação. Nos casos em que um projeto de melhoria renova uma aplicação com pouca documentação, este passo provê um diagrama de contexto que pode ser posteriormente utilizado na comunicação com os usuários a respeito do sistema de software. No caso de um cliente específico, esta representação visual de várias fronteiras de aplicação e suas interfaces abriu uma discussão a respeito da migração de certas aplicações para uma plataforma cliente/servidor, pois nosso diagrama mostrava quais aplicações seriam afetadas pela migração de uma aplicação central. Como esses diagramas são de natureza visual e independentes de tecnologia, sua revisão muitas vezes leva à descoberta de interfaces anteriormente discutidas, porém ausentes dos requisitos escritos.

Além do exposto acima, estes passos demarcam claramente a fronteira lógica entre as aplicações dos usuários. Esclarecendo-se quais funções pertencem a quais aplicações, é menos provável que um conjunto de requisitos venha a ser esquecido. Por exemplo, se uma equipe de projeto estiver supondo – erradamente – que uma outra aplicação atualizará um conjunto de dados comuns, uma revisão do diagrama de contexto demonstrando a interface entre as aplicações poderá revelar eventuais falhas de comunicação.

3. Contar as funções de dados. Este passo considera as entidades de dados internas e externas. Consiste em:

- Identificar, ponderar e contar os Arquivos Lógicos Internos (ALI). Estas são as entidades lógicas persistentes ou grupos de dados a serem mantidos (“mantidos” aqui significa incluídos, alterados ou excluídos) por meio de uma função padrão do software.
- Identificar, ponderar e contar os Arquivos de Interface Externa (AIE), que são pertinentes a outras aplicações e referenciados, porém não mantidos. Normalmente estes dados são utilizados para editar, validar ou reportar tipos de processos de software.

Ao identificar e classificar as entidades lógicas persistentes como internas (mantidas), externas (apenas referenciadas), é útil desenhar círculos ao redor das entidades e respectivas subentidades, em um modelo de dados ou diagrama de entidades-relacionamentos. Se não existir um modelo de dados ou diagrama de entidades-relacionamentos, um será basicamente criado neste passo, a partir do diagrama de contexto elaborado no passo anterior.

Notar que a APF não conta dados *hard-coded* (incorporados ao código dos programas), ou quaisquer tabelas/arquivos criados exclusivamente em função implementação técnica ou física. Este passo, referente a dados, registra a quantidade e o tipo dos elementos de dados quando conhecidos, se já não estiverem identificados nos requisitos. Isto provê um

checklist (lista de verificação) de entidades de dados para avaliar a consistência e completude das funções transacionais (que manipulam os dados).

A revisão das entidades frequentemente fornece respostas para várias perguntas, seja a revisão efetuada sobre um modelo de dados ou mesmo sobre um diagrama desenhado a mão. Esta revisão aplica-se igualmente a entidades internas à fronteira da aplicação (i.e., a serem mantidas pelo software) e externas (i.e., para serem apenas referenciadas). As perguntas respondidas pela revisão podem ser do tipo: “Por que esta entidade é externa? Pensei que fosse necessário atualizá-la.”. Este tipo de situação levará a uma discussão que confirmará os requisitos originais, ou revelará uma inconsistência no entendimento e a uma mudança no diagrama. Quando esta revisão é combinada com as transações consideradas no passo seguinte, a maioria das inconsistências (potenciais) nos requisitos é identificada.

4. Contar funções transacionais. Utilizar o seguinte:

- Entradas Externas (EE) são os processos elementares cuja intenção primária é manter os dados em uma ou mais entidades lógicas persistentes ou controlar o comportamento do sistema. Notar que essas EE são processos funcionais unitários e não fluxos de dados físicos ou estruturas de dados.

- Saídas Externas (SE) são os processos elementares cuja intenção primária é levar dados para fora da fronteira da aplicação, incluindo pelo menos um dos seguintes itens: cálculos matemáticos, derivação de novos elementos de dados, atualização de um ALI, ou alteração no comportamento do sistema.

- Consultas Externas (CE) são os processos elementares cuja intenção primária é levar dados para fora da fronteira da aplicação, por meio de recuperação de dados de um ou mais ALI ou AIE.

Este passo é onde a maioria dos requisitos esquecidos, incompletos ou inconsistentes são identificados. A lista a seguir oferece alguns exemplos dos tipos de descoberta que podem ser obtidas utilizando-se a APF:

- Se uma entidade lógica persistente foi identificada como um ALI, isto é, mantida por uma função padrão de manutenção da aplicação, mas não existirem processos de Entrada Externa a ela associados, haverá um ou mais requisitos com problemas:

- Ou a entidade é apenas uma entidade de referência (caso em que poderia ser um AIE), ou
- Há pelo menos um requisito faltando para manter a entidade, do tipo incluir entidade, alterar entidade, ou excluir entidade.

- Se existirem funções de manutenção de dados (ou de administração de dados) identificadas para os dados, mas não existir uma entidade lógica persistente para armazenar os dados (ALI), o modelo de dados pode estar incompleto. Isto indicaria a necessidade de revisar os requisitos de dados da aplicação.

- Se houver uma função de atualização de dados presente para uma entidade identificada como apenas de referência (AIE), isto poderia indicar que a mesma na verdade é um ALI. Os requisitos de dados estariam inconsistentes e mereceriam revisão.

- Se existirem entidades de dados que precisam ser referenciadas por uma ou mais funções de entrada, saída ou consulta, e não existir a correspondente fonte de dados no modelo de dados, diagrama de entidades-relacionamentos ou diagrama de contexto, os requisitos de dados estarão incompletos e precisarão ser revistos.
- Se existirem funções de saída ou consulta que especifiquem campos de dados que precisem ser exibidos ou transferidos para a saída para os quais não existam fontes de dados (i.e., nem ALI nem AIE), não estando os dados incluídos no código dos programas, haverá uma incongruência entre o modelo de dados e as funções dos usuários. Isto indicaria necessidade de revisão dos requisitos de dados.

A maioria das entidades mantidas segue a convenção AUDIO (*Add, Update, Delete, Inquiry, Output*); cada entidade lógica persistente normalmente terá associado a si um conjunto padrão de funções. Nem todas as entidades seguirão este padrão, mas AUDIO é um bom *checklist* para utilizar com os ALI.

5. Avaliar a complexidade das restrições não-funcionais dos usuários utilizando um fator de ajuste. Por meio de uma avaliação das 14 Características Gerais dos Sistemas (CGS) da APF (por exemplo, as CGS incluem *performance*, eficiência do usuário final, volumes de transação, etc.) pode ser obtida uma avaliação da complexidade do software. O impacto das restrições do usuário nesta área é frequentemente não formalizado e não considerado até tarde no ciclo de vida do desenvolvimento de software, embora sua influência possa ser muito grande no projeto como um todo.

O exame dos requisitos do usuário a partir destas restrições não-funcionais do negócio pode prover as seguintes informações valiosas:

- O requisito não-funcional relativo à carga de transações no período de pico pode precisar de disponibilidade 24x7. Isto poderá ter um impacto crítico sobre o projeto.
- Proteção especial contra a perda de dados pode ter importância crítica para o negócio e precisar ser especialmente desenhada com essa finalidade. Isto deve ser identificado logo, para prevenir impactos não desejados.

A APF fornece um tamanho objetivo do projeto para utilização em equações de estimativas de software (juntamente com outros fatores), ou para normalizar razões de mensuração. O processo verifica se o conjunto completo de requisitos funcionais dos usuários foi identificado, podendo descobrir requisitos faltando ou defeituosos. A Tabela 1 resume como utilizar a APF para descobrir defeitos nos requisitos. A coluna mais à direita da Tabela 1 qual tipo de problema potencial de requisitos pode ocorrer e onde.

Tabela 1 – Utilizando a APF para Descobrir Defeitos nos Requisitos

Função de Dados	ALI	AIE	Função Transacional	EE	SE	CE	Indicador de Problema em Requisitos	Problemas Potenciais em Requisitos
Entidade Empregado (mantida)	X	X					X	1. A mesma entidade não pode ser mantida e apenas referenciada. 2. Faltam funções de manutenção.
Vendas Mensais	X		Incluir Venda Excluir Venda	X X			X	1. Como corrigir os erros (falta atualização)? 2. Não há relatórios com esses dados?
			Relatório de Conta				X	1. Não foi identificada nenhuma fonte de dados sobre "Conta".
% de Componentes da APF Identificados	50	0		10		7	X	Com base no % padrão: 1. Por que não há AIE? 2. Por que não há SE? 3. Todas as funções transacionais foram identificadas?
% de Componentes da APF – Padrão	30	10		40	10	10		

Nota – Os percentuais de componentes da APF acima são fictícios e apenas indicativos de que percentuais como esses poderiam ser utilizados, a partir de valores obtidos de uma amostra de projetos semelhantes ao que estivesse sendo avaliado.

Benefícios Posteriores à Fase de Requisitos

Disponer de um conjunto documentado de requisitos funcionais do usuário (e dos requisitos não-funcionais que a APF contempla) como provido pelo processo da APF vai muito além da fase de requisitos. Os *Material Systems Group* (MSG) das bases de Hill e Tinker da Força Aérea Norte-Americana concluíram ser este o caso. Um exemplo da base Hill serve para ilustrar este ponto: O MSG anexou uma lista completa dos requisitos funcionais (utilizando a documentação da APF com os componentes contados) à estimativa do projeto enviada ao quartel-general. Posteriormente, quando perguntas surgiram sobre um conjunto de funcionalidades e se as mesmas tinham sido incluídas na estimativa, o grupo consultou a listagem da APF para verificar se as referidas funcionalidades estavam listadas. Se não estava, ficava claro que a mesma não tinha sido incluída. Nesse ponto, uma decisão era tomada a respeito de incluir ou não as funcionalidades e ajustar as estimativas.

Este simples conjunto de funções documentadas minimizou as discussões do tipo “quem disse o quê e quando”, reduzindo as interações a se as funções tinham ou não sido incluídas nas especificações submetidas. Além disso, quando mudanças no escopo emergiram posteriormente no projeto, como sempre acontece, os dois grupos estavam em uma posição de poder ajustar os tamanhos em PF e rapidamente avaliar o impacto da mudança de escopo sobre o projeto.

Embora outras técnicas de revisão e acompanhamento dos requisitos possam prover valor, a APF é um método simples que fornece tanto o tamanho funcional do software (útil para estimativas) e pode ajudar no processo de requisitos.

Resumo

Hoje em dia o analista de software precisa de toda a ajuda que puder encontrar na busca de requisitos do usuário completos e conhecidos. A estrutura (*framework*) fornecida pela APF dá ao analista mais uma estrutura de referência para avaliar a completude dos requisitos conhecidos do usuário. Defeitos de requisitos ainda irão ocorrer, não importando quantas estruturas de referência forem utilizadas. Contudo, a utilização da APF para estender as estruturas de referência tradicionais baseadas em teoria e experiência pessoal aumentarão a habilidade do analista de garantir que os requisitos de software estejam completos.

Sua organização deve considerar a utilização da APF? A resposta dependerá da estrutura, metas e objetivos quanto à mensuração. A APF é uma ferramenta que pode ajudar nos seus processos de requisitos e prover um valor quantitativo para dimensionar seu software. Para aqueles que vem usando a APF apenas para obter o tamanho do software, valiosos benefícios podem ser obtidos por meio da aplicação da APF à revisão estruturada dos requisitos, especialmente no que se refere à *completude* dos mesmos.

Referências

1. Requisitos de qualidade podem ser encontrados no conjunto de normas ISO/IEC 9126:2000, que contempla as restrições do tipo *ility* (em inglês), tais como portabilidade, segurança, usabilidade, confiabilidade, etc. (*portability, security, usability, reliability, etc.*). Contatar a ISO para mais detalhes.
2. O Manual de Práticas de Contagem (Counting Practices Manual – CPM) é mantido pelo International Function Point Users Group (IFPUG) e estava, à época em que este artigo foi escrito, em sua versão 4.1 (1999). Hoje (2009) encontra-se em sua versão 4.2.
3. O *checklist* AUDIO foi obtido por Carol Dekkers através de conversa com John Van Ordem, à época consultor da Quality Plus Technologies e anteriormente do Gartner Group.

Nota

1. Na discussão de questões relativas a estimativas, muitos outros fatores são envolvidos além do tamanho funcional do software, tais como o tipo de software, requisitos técnicos, número de usuários, localizações geográficas, etc.

Sobre os Autores

Informações sobre os autores podem ser obtidas em:

Carol Dekkers – <http://www.qualityplustech.com>

Mauricio Aguiar – <http://www.metrics.com.br>