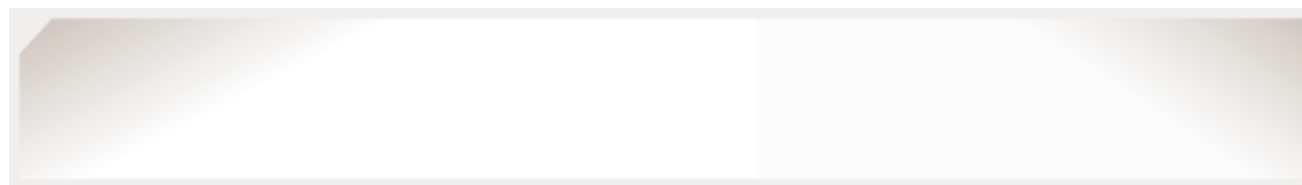


[Fale Conosco](#)[Sala de Imprensa](#)[Suporte](#) [Enviar](#)[> Esqueci minha senha](#)[Home](#)[Mais TI](#)[Artigos](#)[Entrevistas](#)[Notícias](#)[Artigos](#)

Entrevistas

[< Voltar](#)

Medindo a produtividade da equipe de software

Alcançar a máxima produtividade das equipes é o desejo de qualquer empresa que almeja o sucesso de seu negócio. Mas, como atestar que uma equipe de software é de fato produtiva? Como melhorar o desempenho para atingir a alta performance? Essas são algumas das questões esclarecidas por Carlos Eduardo Vazquez, especialista em Pontos de Função.

Data: **12/03/2010** | Entrevistado: **Carlos Eduardo Vazquez**

LG: O que é uma equipe de desenvolvimento de software produtiva?

Carlos: A resposta é simples: É uma equipe que produz. No caso em questão, cabe perguntar: Produz o quê?

O termo software refere-se a uma diversidade de produtos como: programas de computador; especificações de interfaces com o usuário e outros componentes de software; documentos de requisitos; projetos de design etc. Portanto, quando se discute produtividade no contexto do software é necessário, antes de tudo, estabelecer qual a perspectiva em que se deseja avaliar uma equipe como produtiva e uma escala de comparação quanto à maior ou menor produtividade.

Essa escala se faz necessária porque o interesse normalmente não é saber se uma equipe é produtiva, mas sim:

1. O quão produtivo ela é;
2. O quão produtivo ela precisa ser; e
3. Como ela se posiciona em comparação com os outros players no mercado.

Quando a perspectiva escolhida é uma perspectiva econômica que independa dos detalhes técnicos – e são muitos os detalhes técnicos envolvidos no desenvolvimento de software-, uma boa opção é considerar como produto a **funcionalidade** entregue pelo projeto aos usuários. Já o atendimento de demandas de melhoria em um software já existente pode ter como produto as funcionalidades novas incluídas, as funcionalidades existentes excluídas ou alteradas.

LG: Sabemos que existem diferentes atividades pertinentes a software, como: correções, melhorias funcionais, melhorias tecnológicas etc. É possível medir a produtividade em todas elas? De que forma?

Carlos: Sempre que se fala em produtividade, primeiramente, devemos avaliar o que se está comprando. Quando falamos de correções (não cobertas por garantia), suporte de help desk de terceiro nível e outras atividades de sustentação, o que está se comprando é a disponibilidade do serviço (em contraste a compra um produto). Para que se possa falar de produtividade, é necessário haver algum produto ou serviço sendo produzido.

Em uma determinada janela de tempo, nenhum trabalho de sustentação pode ser demandando. Mas, quando o é, o atendimento deve obedecer a certos níveis de serviço como o prazo para o início do atendimento e o prazo para a solução de um problema, por exemplo.

Nesse tipo de demanda, o importante não é a produtividade, mas o nível de serviço. A gerência desse tipo de demanda deve ser realizada pelo planejamento e monitoramento de Acordos de Nível de Serviço e não pela produtividade.

Vamos considerar que uma melhoria tecnológica diga respeito a aspectos que não estejam mapeados para as práticas e os procedimentos de negócio. Ela não afeta funcionalidade alguma. Quando se trata de uma intervenção de impacto amplo, que afete a aplicação como um todo, podemos dizer que o produto entregue são as funcionalidades então existentes em uma nova mídia ou com uma nova infra-estrutura de suporte. Portanto, pode ser encarada como um novo desenvolvimento em termos de quantificação da produção e avaliação da produtividade.

Vale destacar que quando falamos em produtividade, considerar que medimos uma equipe é uma simplificação da coisa. Na verdade, avaliamos um processo produtivo. O que não pode ser feito é colocar um tipo de demanda como essa, do tipo *refactoring*, em um mesmo saco que um desenvolvimento de um novo sistema do zero: cada processo produtivo terá uma produtividade média própria e um conjunto de entradas e saídas diferentes.

A “melhoria tecnológica”, em uma escala menos abrangente – não se trata de um *refactoring* –, também pode ter a sua produção quantificada, contudo, o melhor caminho é estabelecer um peso relativo comparado ao que denominou “manutenção funcional” de forma que se possa trabalhar com uma unidade de medição de software única (UMS).

Nesse cenário, uma unidade de medição funcional (UMF) equivale a uma UMS e existe uma “tabela de preços” expressa em UMS para tipos de melhoria tecnológica não mensurável pela UMF. Ficou confuso, não é? Vamos tentar com um exemplo:

Um determinado processo demora hoje dois dias de processamento e é necessário reestruturar os programas para que esse processamento aconteça em apenas um dia. Nesse caso, nos resta apenas considerar os artefatos técnicos que devem sofrer intervenção. Afinal, não houve função alterada.

Em tempo de definição de políticas e premissas, o ideal é comparar em ensaios controlados o tipo de trabalho envolvido em média com cada programa e criar uma escala relativa com a UMF. Digamos que cada programa tome o equivalente a 5 horas de trabalho e que a taxa de entrega estabelecida em média para o contexto em análise de 10 horas por UMF. Podemos dizer que o produto referente a cada programa será “equivalente a 0,5 UMF”. Como a coisa de funcional não tem nada, seriam 0,5 UMS.

Em tempo de medição das demandas, havendo a necessidade de intervenção desse tipo em 10 programas, serão consideradas para fins de medição, 05 UMS.

O que chamamos de “melhorias funcionais” não existem em um sentido restrito. Existem manutenções que modificam as regras e procedimentos de negócio mapeados como funções da aplicação (requisitos funcionais). Contudo, não há função que não necessite de um suporte tecnológico, uma infra-estrutura técnica. As intervenções relativas ao atendimento desses novos requisitos funcionais não devem passar pela ponderação descrita no parágrafo anterior e utilizar apenas as unidades de medição funcional para ponderação.

LG: Quais as vantagens de utilizar a APF na medição da produtividade de uma equipe de software?

Carlos: A APF como unidade de medição funcional é o método mais maduro que existe e com maior suporte em termos de profissionais e organizações com a expertise na mesma. Adicionalmente, existe o IFPUG – *International Function Point Users Group* – que é responsável pela manutenção e evolução da técnica desde 1986.

Tomo a liberdade de estender a sua pergunta: Por que utilizar a Medição Funcional? Porque ela considera os requisitos funcionais, mapeados para as práticas e procedimentos de negócio, externos à função de desenvolvimento e, portanto, na esfera de controle do cliente, daquele que demanda os serviços de desenvolvimento. Se usássemos alguma métrica interna (como infelizmente é inevitável naquelas demandas descritas no item anterior), não seria possível ao cliente questionar aquilo que é apresentado pelo desenvolvimento. Isso por si só, em minha opinião já é motivo suficiente.

Isso habilita a criação de um processo de gestão em que tendências opostas, porém complementares, interagem na busca de um equilíbrio. Quais tendências? O desenvolvimento tem uma tendência de inflar a sua produção enquanto o seu cliente tem a tendência oposta de diminuir a medição de seu pedido. O interessante da APF é que não cabe uma discussão apaixonada, existe todo um corpo de regras e procedimentos que cumpre o papel de moderar essas interações.

LG: Uma pessoa que não seja da área de TI consegue utilizar esta técnica?

Carlos: Perfeitamente! Já treinei mais de 2.000 pessoas nesse assunto e posso falar com tranquilidade: mais da metade do tempo é gasto apoiando os profissionais de TI a desaprender uma série de detalhes técnicos que apenas dificultam a contagem. Entre essas pessoas, ainda que uma minoria por uma questão de cultura, havia aquelas que eram da área de negócio e que tive a oportunidade de assistir aplicando a técnica de maneira adequada.

LG: Por qual motivo você acha que a produtividade deve ser medida?

Carlos: Porque é impossível gerenciar aquilo que você não consegue medir. Qualquer reunião de trabalho vira conversa de botequim. Falando em botequim, vale citar Noel Rosa que compôs Feitio de Oração em 1932. Nela há um trecho que cito sempre que posso: “Quem acha, vive se perdendo”. Se não houver métrica de produto e de produtividade, qualquer reunião para discutir o processo de trabalho, o desempenho da equipe, as consequências da adoção de uma nova ferramenta etc., vira uma feira de achados e perdidos!

LG: Depois de medida, como fazer para melhorar a produtividade?

Carlos: Uma organização opera em diferentes níveis. O que discutimos até agora diz respeito a uma perspectiva tática e estratégica. Há necessidade de supervisão cotidiana numa escala operacional, há necessidade de liderança efetiva. Há necessidade de profissionais técnicos com habilidades não só de desempenhar o seu trabalho, mas também de atuar como multiplicadores de competências num cenário em que a demanda por serviços só faz aumentar e o volume de novatos é grande. Há necessidade de processos que não engessem a criatividade e que não sejam apenas folhas de papel para manter uma certificação, mas uma coisa viva que cumpra um propósito de orientar a sequência de trabalho e os padrões dos produtos entregues entre os envolvidos. Há necessidade de talentos, pessoas que sejam apaixonadas pelo que fazem e que não se sintam sufocadas por um processo que ao invés de potencializar as suas capacidades, as tolham.

A vantagem de um modelo de acompanhamento da produtividade bem estruturado desenvolvido é a liberdade que ele traz à equipe de desenvolvimento (e, portanto, uma grande responsabilidade!). Não se pretende definir a produtividade de uma ordem de serviço em particular, não se pretende criar um modelo automático de gerência, pretende-se definir uma meta de produtividade para cada processo ou versão do produto desenvolvido. A equipe de desenvolvimento tem a liberdade de realizar toda a gestão de seu próprio trabalho desde que, ao final do projeto, quando se apura o total entregue e o total investido em termos das horas da equipe de desenvolvimento, se respeitem as metas de produtividade e qualidade.

LG: Na sua opinião, o que mais prejudica a produtividade de uma equipe?

Carlos: Eu vejo dois mundos: um mundo do desenvolvimento de software e outro da manutenção. No primeiro, vejo que o principal inimigo da produtividade está no plano dos requisitos. Imagine uma obra onde se começa o acabamento enquanto ainda está sendo colocada a alvenaria. Imagine como seria se as modificações feitas em uma planta no AutoCAD fossem feitas na mesma intensidade e frequência no canteiro de obras. Hoje existem duas principais estratégias para o desenvolvimento de sistemas: o desenvolvimento em cascata, no qual apenas depois de esgotado o trabalho de requisitos, se inicia os demais trabalhos. Qual o problema? O usuário não consegue materializar as suas necessidades e desejos sem materializar melhor a solução e, como consequência, apenas quando recebe o produto em um estágio bem mais avançado, obtém êxito na elicitação e análise dos requisitos. Abordando também esse problema, surgem abordagens iterativas e incrementais nas quais não há mais coincidência entre as fases de acompanhamento gerencial do projeto que se descolam das disciplinas da engenharia de software. Em termos mais simples, em uma fase de concepção da solução, existe não só o exercício das disciplinas de requisitos, mas também se exercitam disciplinas de análise e projeto ou implementação. A intenção é abordar o mais cedo possível aspectos de maior risco e materializar para os usuários por meio de protótipos, que prefiro chamar bonecos, aquilo que receberá quando da entrega do produto. O problema é que apesar de já não ser uma novidade, poucos operacionalizam essa estratégia de desenvolvimento de fato, em minha opinião. Esse assunto é bastante extenso e temo fugir do foco.

Já no mundo da manutenção, tive contato com sistemas de missão crítica com trinta anos de idade em produção e constante manutenção. Isso é um problema? Não necessariamente. Mas, uma coisa é um ambiente em que se investe na manutenção de forma que não se degrade a integridade e estrutura do software, outra coisa é o contexto do “qual a forma mais rápida e barata para fazer o sistema funcionar desse jeito?”. Eu acredito que quem trabalha com manutenção entenda perfeitamente o que digo. Essa perda da estrutura introduz tanta complexidade no software, que muitas vezes parece tornar-se com vida própria e isso aumenta exponencialmente o trabalho necessário para atender a inclusão, alteração ou exclusão de uma funcionalidade.

LG: Na sua opinião, como podemos motivar uma equipe de desenvolvimento de software para que ela seja cada vez mais produtiva?

Carlos: O manifesto ágil não deve ser ignorado na avaliação do seu questionamento. Acredito que não sejam apenas processos e ferramentas, mas sim as pessoas. Mas chamo a atenção para um ponto crítico: vejo os praticantes do SCRUM alardeando os ganhos de produtividade, mas nunca vi isso sendo medido de forma consistente.



Sócio-fundador da Fatto Consultoria e Sistemas, Carlos Eduardo Vazquez é especialista na técnica de Análise de Pontos de Função, certificado pelo International Function Point Users Group – IFPUG. Graduado pela PUC-Rio, com mais de 20 anos de experiência em desenvolvimento, manutenção e gestão de sistemas corporativos, atua como instrutor e consultor de empresas. É também autor do livro "Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software".

 Esta página O site da LG

Uma seleção dos assuntos mais relevantes de TI e RH publicados nos principais veículos de comunicação.

Goiás
(62) 3545-9000

Minas Gerais
(31) 2105-0800

Paraná
(41) 3343-9418

Pernambuco
(81) 3463-1343

Rio de Janeiro
(21) 3257-0800

São Paulo
(11) 3841-2244