



P	C	T
110	23	

O PODER DAS MULTIDÕES

POR QUE A FORÇA DA COLETIVIDADE ESTÁ
REMODELANDO O FUTURO DOS NEGÓCIOS

JEFF HOWE

Tradução

Alessandra Mussi Araujo

2

NO INÍCIO ERA TUDO TÃO SIMPLES...

Criando o projeto do crowdsourcing


CAMPUS

Do original: *Crowdsourcing*
Tradução autorizada do idioma inglês da edição publicada por Crown Business
Copyright © 2008 by Jeff Howe

© 2009, Elsevier Editora Ltda.

Todos os direitos reservados e protegidos pela Lei nº 9.610, de 19/02/1998.
Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

Copidesque: Cláudia Amorim
Revisão: Mariflor Brenlla Rial Rocha e Edna Rocha
Edição Eletrônica: Estúdio Castellani

Elsevier Editora Ltda.
Rua Sete de Setembro, 111 – 16ª andar
20050-006 – Centro – Rio de Janeiro-RJ – Brasil
Telefone: (21) 3970-9300 Fax: (21) 2507-1991
E-mail: info@elsevier.com.br
Escritório São Paulo
Rua Quintana, 753/8ª andar
04569-011 – Brooklin – São Paulo – SP
Tel.: (11) 5105-8555

ISBN 978-85-352-3151-9
Edição original: ISBN 978-0-39620-4

Nota: Muito zelo e técnica foram empregados na edição desta obra. No entanto, podem ocorrer erros de digitação, impressão ou dúvida conceitual. Em qualquer das hipóteses, solicitamos a comunicação à nossa Central de Atendimento, para que possamos esclarecer ou encaminhar a questão.

Nem a editora nem o autor assumem qualquer responsabilidade por eventuais danos ou perdas a pessoas ou bens, originados do uso desta publicação.

Central de atendimento
Tel.: 0800-265340
Rua Sete de Setembro, 111, 16º andar – Centro – Rio de Janeiro
e-mail: info@elsevier.com.br
site: www.campus.com.br

CIP-Brasil. Catalogação-na-fonte
Sindicato Nacional dos Editores de Livros, RJ

H845p Howe, Jeff
O poder das multidões : por que a força da coletividade
está remodelando o futuro dos negócios / Jeff Howe ; tradução
Alessandra Mussi Araujo. – Rio de Janeiro : Elsevier, 2009.

Tradução de: *Crowdsourcing*
Inclui índice
ISBN 978-85-352-3151-9

1. Planejamento empresarial. 2. Desenvolvimento
organizacional. 3. Cultura organizacional. 4. Pessoal – Seleção
e admissão. 5. Terceirização. I. Título.

08-4631.

CDD: 658.4
CDU: 658.012.2

NO INÍCIO ERA TUDO TÃO SIMPLES...

Criando o projeto do crowdsourcing

No início, todo código-fonte era aberto. Isso se devia mais às circunstâncias do que ao projeto, mas as conseqüências iriam muito além da esfera da programação de computadores. O código-fonte é formado por comandos em inglês que, uma vez traduzidos em zeros e uns, dizem ao computador o que deve ser feito. O código-fonte aberto é exatamente o que parece: aberto para que qualquer um veja, copie e use, não importa o propósito. Como era aberto, criou-se um espírito de colaboração e livre troca de informações na programação de computadores, e como já havia sido aberto, um pequeno grupo de programadores íntegros determinou que deveria permanecer aberto. Não poderiam obrigar a Microsoft, a Sun Microsystems ou a Apple a revelar seus códigos, mas poderiam criar uma alternativa aberta e livre.

Para fazer isso, os fundadores do movimento do software de código aberto tinham que criar uma maneira nova de fazer as coisas. Não podiam oferecer dinheiro a ninguém, e a tarefa que tinham pela frente – escrever um sistema operacional inteiro, que requeria milhões de horas de trabalho dos programadores – era desanimadora. Será que pessoas altamente capacitadas contribuiriam com seu tempo livre em um projeto que parecia destinado ao fracasso? Na verdade, sim. Muitas e

muitas pessoas contribuiriam. E como muitas pessoas se ofereceram, o fardo acabou não pesando para algumas apenas. No início dos anos 90, a multidão havia realizado seu primeiro trabalho significativo: Linux, um sistema operacional superior, em muitos aspectos, aos melhores produtos de muitas empresas. O código aberto estabeleceu um precedente – uma prova de conceito. Se as pessoas que trabalham em seu tempo livre – químicos, músicos em bares, fotógrafos de festas – fornecem o combustível para o motor do *crowdsourcing*, foi o movimento do software de código aberto que criou o projeto.

É lógico que ninguém sabia disso em 1969, quando Ken Thompson, um programador de computadores do Bell Laboratories, de repente ficou no olho da rua. Thompson havia trabalhado durante cinco anos em um projeto ambicioso, em parceria com o MIT, a General Electric e o Bell. O objetivo do projeto era criar um sistema operacional mais eficiente para os computadores mainframe da época, um sistema que fosse capaz de realizar mais de uma tarefa por vez – uma limitação que tornava incrivelmente lenta a velocidade de processamento até mesmo dos computadores mais avançados. Mas em vez de eficiência, o esforço criou um caos. O manual do sistema operacional tinha mais de 3 mil páginas. Nessa época, o Bell pulou fora.

Tendo que tirar quatro semanas de férias forçadas, Thompson decidiu começar do zero, desta vez sozinho. Em vez de estabelecer uma meta muito alta, estabeleceu uma meta pequena. Passou a escrever quatro componentes de um sistema operacional, dedicando uma semana a cada um deles. Steven Weber, autor e cientista político, escreveu, em *The Success of Open Source*, “com apenas um homem-mês e uma estrutura básica de hardware, Thompson tinha de deixar para trás a mentalidade dos sistemas grandes e fazer algo simples”. Ou como mencionou na época um dos colaboradores de Thompson, “criar coisas pequenas mas bem-feitas em vez de algo grandioso”.

Até o fim do mês, Thompson já havia escrito o esboço do Unix, que se tornaria o sistema operacional de maior sucesso e o mais duradouro já criado. Mas não foi só a popularidade do Unix que fez as decisões de Thompson tornarem-se históricas; foi sua decisão de criar o Unix a partir de algo pequeno, de programas discretos que tinham o objetivo de fazer uma única coisa, mas fazê-la bem. Essa arquitetura permiti-

ria finalmente que centenas de programadores trabalhassem juntos de uma forma totalmente descentralizada – muito parecida com a maneira como milhares de colaboradores trabalham juntos atualmente para realizar um singular trabalho de referência, a Wikipedia. Dividir o trabalho em pequenas unidades, ou módulos, é uma das marcas registradas do *crowdsourcing*. Nesse caso, isso facilita a abordagem “vem um, vêm todos” da programação de código aberto.

Não que houvesse muitas partes interessadas na época. O campo da ciência da computação desenvolveu-se a partir de laboratórios acadêmicos e semi-acadêmicos de pesquisa, como os do Bell e do MIT. Compartilhar o código do computador estava de acordo com a tradição acadêmica da livre troca de informações, mas era também um expediente comum: era a única maneira de se conseguir alguma coisa. Um dos primeiros computadores comerciais, o IBM 705, custou \$1,6 milhão em 1953, ano em que foi lançado. (Isso corresponde a mais de \$12 milhões em 2008.) Além de ser caro e ocupar muito espaço em uma sala, gastava-se um tempo extraordinário para escrever os códigos que diziam a ele o que fazer. Para maximizar seus recursos, as poucas pessoas capacitadas a realizar tal tarefa uniram seu tempo e talento.

Uma cultura de programadores surgiu durante esses anos de formação e caracterizou-se por interações competitivas, brincalhonas e também extremamente colaborativas entre os especialistas dedicados. O espírito resultante desse grupo valorizava a originalidade e a criatividade e, acima de tudo, o acesso irrestrito às informações – o código do computador principalmente. Não era possível separar os usuários dos programadores porque as únicas pessoas que usavam os computadores eram as mesmas pessoas que os programavam. Esses eram os hackers originais. Embora o termo tenha adquirido conotação negativa, originalmente significava alguém cujo domínio da computação alcançava o estado da arte.

Somente com a introdução do computador pessoal apareceu a necessidade de um software patenteado. Isso gerou um conflito. Em 1976, Bill Gates e Paul Allen – co-fundadores e os únicos empregados da então chamada “Micro-Soft” – escreveram “An Open Letter to Hobbyists” (Uma carta aberta aos entusiastas por hardware). Não mediram palavras: “Como a maioria dos ‘hobbistas’ deve saber, vocês roubam seu

software.” A carta era uma condenação inequívoca do que havia se tornado o hábito dos hackers – o livre compartilhamento de software e do código-fonte no qual estava baseado. Gates destacou que os royalties que ele e Allen pagaram pelo lançamento da primeira versão comercial da Micro-Soft foi de, em média, \$2 por hora. “Quem escreveria um software com tamanho incentivo?”, Gates se perguntou. Os hobbistas precisavam de programadores profissionais porque, antes de qualquer coisa, “qual hobbista conseguiria alocar o equivalente a anos de trabalho de três homens para programar, encontrar todos os bugs, documentar seu produto e distribuí-lo de graça?”. Gates jamais poderia prever a resposta a essa pergunta, a de que um único hobbista não seria capaz de alocar o equivalente a anos de trabalho de três homens em uma tarefa desanimadora, porém milhares de hobbistas o conseguiriam facilmente.

Comprando uma boa briga

Em 1983, Richard Stallman, cientista do MIT, decidiu travar uma batalha de um único homem contra a indústria do software que Gates havia ajudado a criar. Ao fazer isso, deu um nome ao que anteriormente havia sido uma vaga predileção por parte da comunidade de hackers. Em 1970, Stallman chegou em Cambridge para realizar suas atividades na Harvard University, fruto de uma educação extremamente liberal, típica do bairro de Upper West Side, em Manhattan. Com uma juventude precoce e altamente idiossincrática, Stallman disse que não tinha amigos dentro de seu grupo até ir para o Artificial Intelligence Lab, do MIT. Ele passaria os próximos 13 anos de sua vida lá, dormindo em seu escritório e vivendo para escrever o código.

No início dos anos 80, Stallman observou como a comunidade de hackers, que havia prosperado no MIT, desmanchava-se. A maioria de seus amigos e colegas largou a universidade para abrir empresas dedicadas ao desenvolvimento de software patenteado que atenderia ao *boom* do mercado da computação. Nesse momento, até mesmo o Unix – a espora da cultura hacker – havia sido patenteado. Em protesto, Stallman fundou o GNU Project, um esforço para criar um sistema operacional baseado em um código-fonte “aberto” ou disponível gra-

tuitamente. (O GNU é um acrônimo recursivo de “GNU-Not-Unix” – um ótimo exemplo da inclinação do jogo bem-humorado de trabalho dos hackers.)

Foi a primeira tacada de uma revolução memorável, porém poucos notaram isso na época. Stallman cortou todas as afiliações com o MIT (embora a universidade ainda permitisse que ele usasse o laboratório e dormisse em seu escritório) e começou a escrever um sistema operacional baseado no Unix, porém de uma maneira que permitisse a outros usuários roubar, copiar, recortar, colar, modificar e, o mais importante, adicionar suas próprias contribuições ao código-fonte que Stallman estava escrevendo. Atraídos pelo acesso fácil do sistema de Stallman, outros programadores começaram a trabalhar com ele no GNU Project. “As pessoas começaram a pedir e a escrever sobre melhorias”, disse Stallman, “e isso acabou tornando-se muito melhor do que eu havia planejado originalmente”. Como o sistema operacional GNU estava baseado no Unix, com milhares de pequenos arquivos, era fácil para outros programadores escolher e pegar os bits individuais que eles queriam trabalhar, de acordo com o tempo que podiam dedicar ao projeto. Em 1985, Stallman fundou a organização sem fins lucrativos FSF (Free Software Foundation) “para estimular a liberdade do usuário de computador e defender os direitos de todos os usuários de software livre”. A FSF ajudou Stallman a apoiar seu trabalho sobre o GNU, mas também serviu para articular suas idéias: “O software livre é uma questão de liberdade, não de preço. Para entender o conceito, devemos pensar em liberdade como em um discurso livre, não como uma cerveja grátis.”

Ao tornar seu sistema operacional acessível a custo zero, Stallman estava praticamente sozinho, mantendo viva a cultura hacker. Até 1986, Stallman havia criado um compilador C, sem dúvida a parte mais importante de um sistema operacional, em um código completamente livre. Para assegurar-se de que nenhum programador de alguma empresa pegasse o código e o incorporasse em um software que então seria colocado à venda, Stallman criou uma contribuição ainda maior para o movimento do código aberto e para a cultura em geral: a GNU GPL

*Em 1998, uma reunião de líderes do software livre votou a favor do uso do termo “software de código aberto”. (Os termos, para nosso propósito, são sinônimos.)

(GNU General Public License), a qual não só determinava que qualquer coisa lançada sob a licença se tornasse disponível gratuitamente, mas também que qualquer software que a *incorporasse* usasse a mesma licença. “Um software ‘convertido’ da GNU GPL foi usado com sua própria licença, uma abordagem muito inteligente de propagar a liberdade”, observa Glyn Moody em sua história sobre o movimento do código aberto, *Rebel Code*. Esse pequeno truque tornou-se conhecido como “*copyleft*”, em oposição ao *copyright* (direitos autorais).

Até 1991, Stallman e um pequeno grupo de programadores já haviam quase terminado o GNU Project. Mas o último componente que faltava, o kernel (basicamente o coração de um sistema operacional) era também o mais difícil de escrever. Ninguém esperava que fossem necessários pelo menos mais dois anos. Enquanto isso, poucas pessoas fora da comunidade dos hackers haviam ouvido falar do GNU, e menos pessoas ainda haviam usado o sistema. Tudo isso iria mudar.

Em agosto daquele ano, um estudante finlandês de ciências da computação, Linus Torvalds, colocou uma mensagem em um quadro online: “Estou criando um sistema operacional (livre) (é apenas um hobby, nada tão grande e profissional como o GNU)... Queria saber que recursos a maioria das pessoas prefere.” Sem paciência para esperar o GNU kernel, Torvalds estava escrevendo seu próprio, que logo seria chamado Linux – e esse pedido aberto de ajuda teria conseqüências históricas. Durante os próximos dois anos, milhares de codificadores dariam suas contribuições para melhorar o Linux. “O que havia sido o ‘hobby’ de um hacker tornava-se uma comunidade”, diz Moody. “Quanto melhor o Linux se tornava, mais pessoas o usavam; e quanto mais as pessoas corrigiam os bugs, mais rapidamente ele melhorava: um círculo virtuoso que levava o desenvolvimento do Linux a um ritmo vertiginoso.”

Atualmente, o Linux opera qualquer coisa, de supercomputadores a celulares e gravadores digitais, como o TiVo, para não mencionar os milhões de computadores pessoais que usam o Linux. Como o Linux usa uma licença GNU General Public, nenhuma empresa pode usá-lo como base para um lançamento comercial. O resultado é a garantia de que “esse círculo virtuoso” continue a prosperar.

O Linux também serviu para popularizar os projetos de software de código aberto em geral. Existem hoje em dia mais de 175 mil projetos

de código aberto em andamento, hospedados no site de desenvolvimento de software Sourceforge.net, e o código aberto tem sido amplamente adotado por grandes corporações. Cerca de 70% dos softwares de servidores da Web operam com o servidor Apache Web, que foi desenvolvido usando-se métodos de código aberto, e mais da metade dos programas de e-mail em grande escala também usam software de código aberto. Trabalhando fora de qualquer agência organizadora, como uma empresa ou instituição acadêmica, a comunidade de código aberto provou que as redes mais inteligentes eram auto-organizadas. Quem foi o autor do Linux? A multidão.

O movimento do software de código aberto sempre foi uma questão de promover uma filosofia, bem como desenvolver um software. Os proponentes do modelo de código aberto valorizam a transparência para seu próprio bem, não apenas porque abrir o processo de desenvolvimento a terceiros parece produzir um código melhor. Mas é a eficácia do modelo de código aberto, e não os princípios de igualdade que se escondem sob esse modelo, que levam empresas como a IBM, e mais recentemente até a Microsoft, a começar a adotá-lo como forma de economizar dinheiro e desenvolver melhores produtos.

O que torna o código aberto tão eficiente? De maneira geral, é a capacidade de um grande número de pessoas poder contribuir. O evangelista Eric S. Raymond resumiu de maneira fabulosa essa verdade fundamental quando escreveu: "Tendo olhos suficientes, todos os bugs são superficiais" – o que quer dizer que nenhum problema é difícil demais se pessoas suficientes tentarem resolvê-lo. Visto de outra forma, um grupo de trabalho grande e diverso encontrará sistematicamente as melhores soluções do que a maioria da força de trabalho mais talentosa e especializada. Isso vale para áreas como ciência corporativa, design de produtos e criação de conteúdo, como é o caso de software, e é um dos princípios do *crowdsourcing*.

Uma máxima de Raymond apareceu originalmente em "The Cathedral and the Bazaar", um ensaio que Raymond, engenheiro de software de profissão, apresentou em uma conferência em 1997. Escrito em prosa agradável, não-técnica, teria enorme influência sobre a migração das estratégias de código aberto a outras áreas além do software. No ensaio, Raymond contrasta dois métodos de desenvolvimento de software. A

“catedral” caracteriza a abordagem hierárquica, altamente gerenciada que sempre foi o procedimento operacional padrão desde a Revolução Industrial. Raymond contrasta isso com o Linux, “um sistema operacional classe mundial [que] coalesce como por mágica... através de milhares de desenvolvedores diferentes espalhados por todo o planeta, conectados apenas pelo cordão tênue da internet”.

O estilo de desenvolvimento de Linus Torvalds – libere primeiro e com frequência, delegue tudo o que puder, seja aberto e heterogêneo – foi uma surpresa. Nenhuma catedral tranqüila e respeitosa havia aqui – sem dúvida, a comunidade do Linux parecia um grande bazar barulhento com agendas e abordagens diferentes... do qual um sistema coerente e estável aparentemente surgiria devido somente a uma sucessão de milagres.

Em uma catedral, tudo é coordenado a partir de cima. No bazar, tudo é coordenado – se é que essa palavra se aplica – de baixo para cima. Raymond cria um argumento persuasivo para o ponto sensível do código aberto, o cavalo-vapor tumultuado, o qual não só produziu um sistema operacional livre de bugs, mas também o fez a uma “velocidade inimaginável para os construtores da catedral”. Quando Raymond apresentou seu trabalho pela primeira vez no Linux Kongress em 1997, sua importância só foi reconhecida por seus pares no mundo da programação de computadores. Entretanto, logo seria notado por uma platéia muito maior.

Revolucionários acidentais

Nem sempre conhecemos o que criamos. Enquanto trabalhava para a Western Union Telegraph Company, em 1877, para melhorar o telefone de Alexander Graham Bell, Thomas Edison descobriu um método para gravar sons, em vez de simplesmente transmiti-los através de um fio. Em um mês, foi revelado o fonógrafo de Edison. Os papéis o consagravam como o “Mago de Menlo Park”, um apelido que dura até hoje. Edison e seus assistentes continuaram a melhorar sua “máquina falante”, mas ele jamais poderia prever que algum dia ela seria usada como entre-

tenimento. Seu objetivo era monopolizar os negócios, e nas três décadas seguintes o fonógrafo de Edison cumpriu essa tarefa admiravelmente. Foi preciso outro homem, Eldridge Reeves Johnson, para superar a barreira cognitiva que produziria o fonógrafo como o conhecemos atualmente. Esmerado, de fala mansa e ambicioso, Johnson uniu-se a um dos concorrentes de Edison e criou uma versão mais robusta da engenhoca. Adotou um logo inteligente, de um cachorro com as orelhas em pé ouvindo “a voz de seu dono”, batizou a empresa de Victor e chamou seu fonógrafo de Victrola. Sem ser inventor, Johnson sozinho fez procriar a indústria de música gravada. Não é preciso ser um inventor para inovar.

Assim como Edison não sabia que havia inventado o fonógrafo, Larry Sanger, um ex-professor de filosofia, não tinha a intenção de dar o tiro inicial do *crowdsourcing*. Raymond havia escrito “The Cathedral and the Bazaar” no contexto do desenvolvimento de software, mas tornou-se relevante para um mundo muito maior. Sanger teria papel fundamental na introdução dos princípios do ensaio de Raymond para um público maior, embora não soubesse disso em janeiro de 2001, quando saiu para jantar com um amigo de longa data chamado Ben Kovitz. Encontraram-se em San Diego, no Pacific Bar and Grill. Kovitz havia se mudado recentemente de Greenwich, Connecticut, para assumir um cargo de engenheiro de informações em uma empresa local. Estava de bom astral, ao contrário de seu companheiro. Sanger havia deixado a universidade há aproximadamente um ano, e agora se perguntava se havia ou não tomado a decisão certa. Seu novo projeto envolvia nada menos do que a reinvenção da enciclopédia, e não estava indo bem. Na verdade, Sanger temia que todos seus esforços não dessem em nada.

Esta não era a primeira incursão de Sanger no setor privado. Seu interesse por computadores vinha desde a infância, e em 1998 já havia começado o *Sanger's Review of Y2K News Reports*, um compêndio de notícias relacionadas ao Ano 2000 lido por gerentes de TI ávidos por impedir um desastre iminente. A missão do *Y2K News Reports'* expirou com o milênio, mas deixou Sanger com vontade de outro empreendimento relacionado à internet. Ficou sabendo, então, que Jimmy Wales, um rico empreendedor que conhecia superficialmente por meio de grupos on-line de discussão de filosofia, estava buscando projetos de internet para investir.

Sanger falou com Wales sobre a criação de uma compilação de notícias culturais, mas Wales tinha outras idéias. Wales havia mexido um pouco com código de computadores no passado, e havia se apaixonado recentemente pelo movimento Free Software, de Stallman. “A primeira coisa que Jimmy fez foi insistir para que eu lesse ‘The Cathedral and the Bazaar’”, Sanger lembrou-se mais tarde. Wales tinha a idéia de uma enciclopédia que seria chamada de Nupedia (uma brincadeira com GNU, o nome do sistema operacional de Stallman). Assim como a *Encyclopædia Britannica* e todas as outras repetições do conceito desde a rudimentar *Encyclopédie*, de Denis Diderot, em 1751, a Nupedia seria reunida por especialistas. Ao contrário dos trabalhos tradicionais de referência, tudo na Nupedia estaria disponível gratuitamente na internet e seria escrito por voluntários, exatamente como em qualquer outro projeto de código aberto – pelo menos parecia assim naquela época.

Sanger estava fascinado – “apaixonado”, ele se lembra – pelo projeto. Mas Wales e ele tinham algumas ressalvas. Ambos estavam um pouco céticos com relação à capacidade dos amadores em contribuir com conhecimento significativo ao que eles enxergavam como um trabalho de referência autorizada. Apesar de tudo, Sanger tornou-se editor-chefe da Nupedia em janeiro de 2000 e imediatamente criou um conselho consultivo formado principalmente por acadêmicos próximos a ele. Durante os meses seguintes, Sanger e seu conselho criaram um método sofisticado para detectar autores potenciais. Uma vez que um colaborador tivesse escrito uma entrada, isso poria em marcha um processo de revisão de sete etapas, oneroso e que consumiria muito tempo.

A princípio, a Nupedia parecia progredir. Wales e Sanger estavam felizes em ver que no início do segundo trimestre daquele ano alguns artigos já haviam sido escritos, revisados e colocados no site. Até o verão, alguns outros artigos passaram pelo processo, e nos meses seguintes chegaram outros mais. Wales e Sanger haviam imaginado um pingou transformando-se em uma enchente, mas o dilúvio nunca aconteceu, e quando Sanger saiu para jantar com Kovitz, a Nupedia podia gabar-se de apenas uma dúzia de artigos. Com o propósito de usar o bazar de Raymond para criar uma enciclopédia – o mais ambicioso dos compêndios de conhecimento – Wales e Sanger apenas acabaram com outra catedral, que não era lá muito impressionante.

“Ben ouvia o que eu lhe contava, e disse que sabia de um programa que poderia abrir o processo”, disse Sanger. Pouco antes do derradeiro jantar, o renomado programador Ward Cunningham havia escrito um software simples chamado WikiWikiWeb. A idéia era permitir aos codificadores trocar informações mais facilmente, mas Cunningham e a pequena comunidade de desenvolvedores de software, que foram os primeiros a usá-lo, logo perceberam que haveria implicações muito maiores. Um “wiki” – palavra de origem havaiana que significa “rápido” – permite que um número ilimitado de usuários crie e edite textos em uma única página da Web. Ainda melhor, um wiki mantém o registro de todas as edições, o que significa que todo mundo que acessar a página pode ver quais mudanças foram feitas e quem as fez. No início de 2001, os wikis ainda faziam parte do ramo daqueles que trabalhavam com (ou eram obcecados por) tecnologia. Mas Kovitz explicou a Sanger como a Nupedia poderia usar a tecnologia wiki para acelerar o trabalhoso processo de contribuição, edição e revisão de artigos enciclopédicos.

Sanger não precisou de muito para se convencer. “Isso imediatamente fez sentido para mim. Acho que liguei para Jimmy [Wales] mais tarde naquela noite e enviei-lhe uma proposta formal para começar a usar os wikis antes da tarde do dia seguinte”, lembra Sanger. Wales também viu o potencial dos wikis para ajudar Sanger a ganhar ritmo, embora a rapidez não fosse sua idéia principal. Embora Wales tivesse alcançado fortuna pessoal negociando no mercado futuro em Chicago nos anos 90, ele não foi o único a financiar a Nupedia. Isso foi feito através de um dos empreendimentos menos altruístas de Wales, um portal da Web chamado Bomis.com que continha, dentre outros itens, pornografia leve.

Se o interesse humano pelo sexo é constante, o investimento de capital é consideravelmente mais volúvel; Bomis estava sofrendo junto com o resto do setor de tecnologia. Após ficar claro que o volume esperado de contribuições não viria facilmente, Sanger propôs diversas formas por meio das quais a Nupedia poderia transformar sua arquitetura de software para acelerar o processo, mas todas elas envolviam o pagamento de um valor alto pelas horas de trabalho de um desenvolvedor profissional de software. Para piorar ainda mais as coisas, a *Encyclopa-*

dia Britannica havia mudado recentemente de curso e tornado disponível na Web todas as suas 100 mil entradas, fazendo o objetivo da Nupedia – de criar uma enciclopédia gratuita – parecer redundante. (Ao contrário da *Britannica*, a Nupedia usava a “licença pública” de Richard Stallman, o que significava que os usuários podiam reproduzir seu conteúdo de graça, e não apenas lê-lo.) Então, em 3 de janeiro de 2001, quando Sanger propôs transformar a Nupedia em wiki, encontrou Wales receptivo a qualquer idéia que permitisse salvar a Nupedia sem ter que gastar mais dinheiro com ela.*

Mesmo com a luz verde de Wales, Sanger não conseguia transformar a Nupedia em wiki por decreto. A Nupedia havia reunido uma comunidade de acadêmicos, filósofos e futuros enciclopédicos. Sanger estava inclinado a jogar fora um ano de trabalho e mandar os “Nupedians” começarem tudo de novo. Nove dias após o jantar com seu amigo Ben Kovitz, Sanger experimentou o primeiro wiki da Nupedia. “Minha idéia inicial era que o wiki fosse colocado como parte da Nupedia; seria uma maneira de o público desenvolver uma corrente de conteúdo que pudesse ser alimentada no processo da Nupedia”, Sanger escreveu em uma breve biografia sobre a criação da Wikipedia. A Nupedia exigia que todos os autores tivessem especialização credenciada na área em que estivessem escrevendo. “Entretanto, parece que a grande maioria do conselho consultivo da Nupedia não queria ter nada a ver com um wiki.” Sanger, como ele felizmente admite até hoje, era solidário a essa visão. “O compromisso deles era com o rigor e a confiança, uma preocupação que eu dividia com eles.”

Um público ainda maior não parecia compartilhar as mesmas ressalvas. Segundo artigo do *Atlantic Monthly*, em três semanas, os colaboradores haviam criado 17 artigos. Esse número pulou para 150 um mês depois, quadruplicou até o final de abril e subiu para 3.700 até o final de agosto, uma taxa de crescimento muito mais rápida do que a Nupedia havia tido. O número de colaboradores crescia tão rapidamente quanto o número de entradas, enquanto o mundo começava a espalhar essa nova abordagem wiki de reunir conhecimento. Até o fim do ano,

*Para seu conhecimento, Wales alega que somente ele deveria ser considerado o fundador da Wikipedia. Infelizmente, ele não respondeu a nenhum dos pedidos de entrevista para este livro.

a Wikipedia continha 15 mil artigos, e mesmo isso mal poderia dar uma idéia do que estava por vir. Não só o número de artigos continuava crescendo, mas também a taxa de crescimento aumentava rapidamente. Durante os primeiros anos de sua existência, esse crescimento exponencial continuou, até recentemente alcançar um patamar. E que patamar. A Wikipedia tem atualmente 2,2 milhões de artigos – 23 vezes o número de entradas da *Britannica* – somente em inglês.

Sanger e Wales não foram os únicos a aplicar as idéias de Raymond e Stallman a novas áreas. Na mesma época em que Wales e Sanger estavam caçando idéias para salvar sua inexperiente enciclopédia de código aberto, Bob Kanefsky aplicava idéias de código aberto à geologia planetária. Kanefsky, um engenheiro de software que trabalha para a NASA, no Ames Research Center perto de Sunnyvale, na Califórnia, tentava descobrir como aplicar o modelo de computação distribuída usado pela SETI@home ao problema de analisar as imagens de Marte que chegavam ao Ames naquela época.

Em meados do ano 2000, ele procurou Virginia Gulick, geóloga planetária do Ames, com uma dúvida. Grande parte do trabalho de um geólogo envolve identificar e medir formatos de terra, como crateras, cumes e vales através de imagens de satélites. “Isso pode ser muito trabalhoso” Gulick afirmou dando risada. Todo esse tédio pode resultar em um enorme desfecho. Ela e seus colegas também geólogos planetários estão vasculhando o universo em busca de evidências de H₂O. “Esse é um dos motivos pelo qual fomos a Marte: para encontrar evidências de água; se houvesse água, poderia ter havido vida.”

Kanefsky queria colocar on-line todas as imagens de Marte geradas pelas missões Viking nos anos 70 e convidar amadores para fazer o trabalho mecânico de identificar e medir os formatos. Gulick estava cética. “Fiquei pensando: ‘Será que isso realmente vai dar certo?’, eu duvidava de que observadores não-treinados pudessem saber a diferença entre, digamos, uma cratera antiga, intacta e outra degradada.” (Uma cratera antiga tem bordas bem definidas, enquanto uma cratera degradada tem erosão devido à exposição ao vento e água que varreram a superfície de Marte durante bilhões de anos.) Kanefsky e Gulick selaram um compromisso. Antes de colocar um trabalho valioso nas mãos de uma multidão, eles aplicariam um teste. Gulick tinha acesso a

um enorme banco de dados com imagens de crateras de Marte que já haviam sido classificadas e catalogadas. Uma colega de Gulick já havia visto todas as 88 mil imagens, identificando, classificando e medindo todas as crateras impactantes que eram visíveis nas imagens Viking. “Ela era bastante disciplinada”, afirma Gulick. “Havia levado aproximadamente dois anos para isso.”

A NASA colocou silenciosamente todo o banco de dados on-line e pediu à comunidade de astrônomos amadores que acompanhavam todos os movimentos da NASA para ajudar os profissionais a analisar as imagens. Eles copiaram o programa “Clickworkers”. Era o estudo de caso perfeito, porque Gulick e Kanefsky tinham um conjunto de dados de controle – o banco de dados já terminado. Em um mês, milhares de colaboradores haviam analisado com êxito cada imagem do banco de dados. Gulick e Kanefsky ficaram embasbacados. Os voluntários não só aceleraram o mesmo trabalho que uma geóloga profissional levou dois anos para terminar, como também o fizeram com um grau de precisão comparável. O projeto do Clickworkers imita o modelo de produção do código aberto em muitos aspectos. Primeiro, uma tarefa enorme é distribuída através de uma rede massiva. Depois, não há limite do número de colaboradores potenciais. Finalmente, o trabalho em si é dividido em tarefas pequenas, discretas, para que o Clickworkers possa usar tanto uma pessoa com cinco minutos de tempo livre como alguém que não tenha nada melhor para fazer em um sábado além de medir crateras. Isto se tornou crucial para o sucesso do Clickworkers: um estudo da NASA mostrou que 37% do projeto foi realizado por colaboradores que contribuíram apenas uma vez.

A NASA retomou o projeto Clickworkers em 2006, porém dessa vez sem realizar um teste. Os voluntários agora seriam responsáveis por analisar formatos de terra encontrados em milhares de imagens de alta resolução enviadas pela câmera HiRISE (High Resolution Imaging Science Experiment – Experimento Científico de Imagem de Alta Resolução) que agora circula em Marte. “Isso poderia ter um grande impacto na ciência”, diz Gulick. “As pessoas gastam dez minutos por dia fazendo isso, mas é uma ajuda enorme para nós. Elas se ocupam da parte trabalhosa, mecânica e liberam os cientistas para a carga intelectual mais pesada.” Assim como os observadores amadores de pássaros

deram contribuições inestimáveis para a ornitologia através dos esforços de coleta de dados, os Clickworkers estão contribuindo para a ciência planetária. A Wikipedia e os Clickworkers deram prova cabal de que o modelo de código aberto poderia ser aplicado em áreas além do software. Não demoraria muito tempo para que outros levassem isso muito mais adiante.

Crowdsourcing e o problema com as patentes

É engraçado como o destino normalmente muda nas decisões de última hora. No final de outubro de 2005, o cientista político de Berkeley, Steven Weber, reuniu as pessoas mais inteligentes que conhecia em uma sala de conferências em Manhattan para discutir o futuro dos negócios. Weber e um co-autor estavam escrevendo um livro sobre “métodos de código aberto de criação de valor” e queriam que alguns pesos pesados “contestassem seus argumentos”. Entre os convidados estavam um ex-conselheiro do vice-presidente Al Gore, um editor da Harvard University Press e vários executivos do alto escalão de empresas de consultoria de Nova York. Então, um dia antes da reunião, o anfitrião de Weber sugeriu que ele convidasse Beth Noveck, professora de Direito da New York Law School e uma espécie de agente provocador na área jurídica. Weber se lembra vagamente de ter tomado um lanche com uma mulher inteligente e auto-suficiente em uma cafeteria de Upper West Side alguns anos atrás e estendeu-lhe o convite. Noveck quase recusou a oferta de Weber. Estava muito ocupada, explicou, mas tentaria dar uma passada e ficar lá uma ou duas horas.

Em um dia ensolarado, fora do comum para a estação, as pessoas de confiança de Weber reuniram-se em uma sala sem janelas, no escritório da Monitor, uma empresa de consultoria que fica na Madison Avenue. Noveck apareceu pouco depois das onze horas. Sentou-se ao lado de David Kappos, um advogado que cuidava do portfólio de patentes da IBM. Os dois logo começaram uma conversa animada. Noveck havia criado o Democracy Design Workshop, uma comunidade on-line de advogados, eruditos e estudantes dedicados aos esforços colaborativos da reforma jurídica, e era uma das proponentes principais da área jurídica a favor de abrir os sistemas fechados para serem examinados

publicamente. O sistema de patentes estava agora bem no centro de sua atenção.

Parece estranho que a IBM quisesse reformar um sistema que ela mesma havia usado em benefício próprio. Foram concedidas 3.125 patentes em 2007 para a gigante da tecnologia – o décimo quinto ano seguido em que a empresa conseguiu mais patentes do que qualquer outra nos Estados Unidos. A IBM gasta aproximadamente \$6 bilhões por ano em pesquisa, desenvolvimento e engenharia, e seu portfólio ativo tem mais de 26 mil patentes somente nos Estados Unidos. Mas deter uma propriedade intelectual tão ampla gera também muita despesa. Desde 1990 o número de disputas por patentes nos Estados Unidos aumentou mais que o dobro, com um custo médio para cada litígio alcançando aproximadamente \$2 milhões. Uma boa notícia para o pesadelo diário de David Kappos. Ele não tinha 26 mil patentes. Tinha 26 mil alvos para processos complexos, gananciosos, frívolos e de quebrar a cabeça.

Quando o grupo parou para almoçar, Kappos e Noveck continuaram na sala de reuniões, entretidos em uma animada discussão. Noveck havia recentemente levado uma proposta radical para um seletivo grupo de colegas: entrar com uma solicitação de patente para um wiki e convidar o público em geral para ajudar a guiar o examinador de patentes. Os pedidos de patente seriam colocados on-line para que todos pudessem ler, analisar e fazer comentários sobre eles. Pessoas com vasta experiência seriam convidadas a analisar as patentes em suas especialidades, exatamente como a comunidade que cuida da enciclopédia on-line, a Wikipedia, organiza-se em torno de especialidades. Seria, como explicou Noveck, uma melhora significativa em relação ao atual sistema de patentes.

A idéia pareceu familiar a Kappos, cuja equipe na IBM estava de fato discutindo uma abordagem parecida. Nos últimos anos, surgiu um consenso na área de propriedade intelectual: o sistema de patentes estava falido. Toda discussão remanescente tratava de como consertá-lo. Mais da metade de todos os pedidos de patente é aprovada, dando origem a um ninho de ratos de patentes confusas e que coincidem com outras. “Ficávamos brigando sobre patentes que não sabíamos o que significavam, e os tribunais não sabiam o que elas significavam, e até mesmo os patenteados não sabiam o que significavam”, diz Kappos. Ele e sua

equipe tiveram a idéia de abrir as patentes para a revisão por pares, um sistema usado pelas revistas acadêmicas em que vários químicos orgânicos, por exemplo, são convidados a tecer comentários sobre um trabalho escrito por um de seus colegas. Mas isso não chega nem aos pés do que Noveck estava propondo.

Noveck inspirou-se não só nas universidades, mas também no modelo de produção da comunidade usado pelo software de código aberto, bem como nos fenômenos contemporâneos da internet, como os comentários gerados pelos usuários da Amazon sobre os produtos, o banco de dados de filmes on-line IMDb.com e o serviço de respostas do Yahoo!, por meio do qual as pessoas tentavam – com índices de acerto surpreendentemente altos – responder qualquer pergunta. É difícil imaginar que isso pudesse existir se não fosse por alguns iconoclastas irredutíveis, como Richard Stallman, que propuseram algumas idéias não-ortodoxas sobre como a informação deveria ser produzida e distribuída. Esses esforços de *crowdsourcing* basicamente adotaram uma abordagem de código aberto para criar produtos além de software. A proposta de Noveck foi praticamente a primeira a tentar aproveitar a sabedoria presente de forma inerente nas grandes comunidades on-line. Mas foi uma das mais radicais, já que seu objetivo era derrubar uma das funções do governo mais veneradas: a concessão de proteção legal para invenções e idéias originais.

Kappos saiu da reunião intrigado e um pouco assustado. Alguns dias depois, ligou para Noveck para continuar a conversa e sugeriu que comesçassem a trabalhar juntos na proposta. “Disse a ela que achávamos seu plano muito eficiente, mas que não nos atreveríamos a pedir ao governo norte-americano para ceder sua soberania sobre a concessão de patentes”, lembra-se Kappos. O fato de a maior detentora de patentes no mundo, a IBM, e uma professora de Direito de Yale estarem discutindo a terceirização da análise de patentes para a população era uma indicação do alcance que havia conseguido o modelo de produção defendido pelo movimento de software de código aberto.

Nada poderia ser mais antiético em relação à maneira como as patentes atualmente são concedidas nos Estados Unidos. Veja como funciona: um inventor tem uma idéia brilhante; baixa o pedido de patente do site do USPTO (United States Patent and Trademark Office); preenche o

formulário; anexa todos os desenhos pertinentes, descrições técnicas e documentos comprobatórios que julgar necessário e envia o dossiê para o órgão de patentes. Depois é sentar e esperar o telefone tocar.

É uma longa espera. Se o requerente da patente tiver sorte, o julgamento de seu pedido sairá em dois anos e meio – tempo médio entre preencher o pedido e a decisão final. O USPTO atualmente tem um acúmulo de mais de 1 milhão de pedidos que sequer chegaram às mãos de um examinador. Sabe-se que os examinadores estão atolados de trabalho e são mal remunerados. Atualmente existem pouco menos de 5.500 examinadores trabalhando no escritório de patentes, e o número de pedidos de patentes vem aumentando rapidamente, chegando a 467 mil em 2007. Por causa disso, os examinadores só conseguem dedicar vinte horas, em média, para analisar até mesmo os pedidos mais “bizantinos”.

Quando um examinador pega um pedido, deve, primeiramente, localizar o estado da técnica (*prior art*) que inclui patentes anteriores e qualquer outro material publicado que se refira ao pedido em questão. Se nosso inventor tiver uma idéia para, digamos, uma escova de dentes musical (não dê risada, essa é a patente de número 5044037), o examinador terá que vasculhar todo o banco de dados da agência, que contém 7 milhões de patentes, em busca de qualquer coisa que possa tornar a escova de dentes musical redundante.

Isso pode ser viável quando se fala de escovas musicais, mas o trabalho torna-se extremamente difícil quando a patente envolve uma melhoria obscura para um código de programação já existente. Até 2005, o escritório não considerava o treinamento em ciências da computação pré-requisito para trabalhar no USPTO. (Também não havia engenheiros de software batendo na porta do USPTO.) Do mesmo modo, o escritório de patentes tinha examinadores formados em química orgânica tentando avaliar um pedido que poderia muito bem confundir até Bill Gates.

Assim como o público não tem acesso ao processo de análise de patentes, os examinadores não têm acesso ao público. De acordo com as normas do USPTO, “um examinador de patentes não deve consultar alguém de fora durante o processo de análise de uma patente”. Teme-se que esse contato possa comprometer a imparcialidade da agência. Muitos departamentos dentro da agência proíbem até mesmo o uso da internet como fonte em uma análise de patentes.

De modo geral, examinadores sobrecarregados e mal informados têm uma reação previsível, errando por excesso de cautela para não correr riscos na concessão de patentes. Isso cria o nó cego dos processos legais conflitantes que tornaram o trabalho de Kappos tão difícil. Capitalizando-se nessa confusão de patentes, as empresas aceleraram seus pedidos. Em 2005, o *New York Times* divulgou que a Microsoft havia aumentado sua meta de 2 mil para 3 mil pedidos de patentes por ano. Dois dos pedidos de patente da Microsoft naquele ano incluíam “Sistema e Método para Criar um Recado Relacionado a uma Ligação Telefônica” e “Adicionar e Remover Espaços em Branco de um Documento”.

Mas os efeitos dessa fartura de patentes são muito cômicos. “Há empresas lá fora para as quais os litígios de patentes tornaram-se a fonte principal de receita”, observa Kappos. “Elas detêm patentes que jamais tiveram a intenção de usar, que serviram apenas para extorsão.” Essas empresas são normalmente chamadas de “*patent trolls*” e histórias sobre elas viraram lendas: em março de 2006, a NTP, uma empresa detentora de patentes, ameaçou enviar executivos para o exterior para brigar pelos telefones públicos quando sua ação contra a RIM (Research in Motion), fabricante do ubíquo BlackBerry, quase encerrou o serviço. A RIM fechou um acordo de \$612,5 milhões com a NTP. Era evidente que algo precisava ser feito para modificar o processo de patentes.

Logo após sua primeira reunião, Kappos pediu que seus advogados trabalhassem com Noveck em sua nova proposta de abrir o processo de análise de patentes para comentário público. Foi um endosso inequívoco do modelo de software de código aberto. Embora o plano parecesse extremamente radical para uma empresa como a IBM, na verdade encaixava-se perfeitamente na direção que a empresa seguia desde que reformulou radicalmente seu modelo de negócios principal nos anos 90. A empresa que antes protegia com zelo seu software patenteado, agora contribuía com milhares de horas de programadores para trabalhar em projetos de software de código aberto – que não geravam nenhuma receita através de licenciamento; inclusive doava algumas de suas patentes para o Free Software Foundation, grupo defensor do código aberto. Isso não é filantropia. Segundo a empresa, a receita gerada pela prestação de “serviços profissionais” relacionada ao software de código aberto compensava em muito a receita perdida com o licenciamento

de software patenteado. Para a IBM, a jogada resultou em inovações, novos produtos e em um aumento de sua reputação dentro da fechada comunidade de programadores de software.

Em dezembro de 2006, Noveck e Kappos foram convidados para um encontro com alguns procuradores do órgão de patentes. O próprio órgão vinha pensando em uma maneira de recorrer à comunidade de código aberto. Em 2004, o presidente George W. Bush nomeou Jon Dudas para chefiar o USPTO com a tarefa de reformar as patentes. Os procuradores da república disseram a Kappos e a Noveck que se eles conseguissem agendar uma reunião, o escritório de patentes cederia o local. Em janeiro de 2007, a IBM anunciou publicamente a criação do "Peer-to-Patent Project" (uma jogada inteligente com o termo Peer-to-Peer, a popular tecnologia de compartilhamento de arquivos normalmente usada para fazer downloads ilegais de música). O escritório de patentes, mantendo sua palavra, algumas semanas depois sediou um simpósio com alguns dos principais especialistas em propriedade intelectual. Durante esse ano, empresas como Microsoft e General Electric concordaram em participar do projeto piloto, que foi lançado em 15 de junho de 2007. Embora o projeto ainda estivesse na fase piloto, Noveck tinha grandes expectativas. No fim de 2008, aproximadamente 33 mil pessoas haviam analisado cerca de 22 pedidos de patentes e enviado 192 decisões de estado da técnica. "O projeto demonstra que os cidadãos têm mais para oferecer ao governo do que simplesmente votar ou responder pesquisas de opinião. Eles realmente podem contribuir com conhecimento especializado e ficam felizes quando solicitados a fazê-lo", diz Noveck.

Isso marcou um momento histórico para o movimento de código aberto. A agência mais reservada e preconceituosa do governo havia aceitado a idéia de que, talvez, mais conhecimento especializado pudesse ser encontrado em uma ampla e enorme rede pública do que entre uns poucos "engomadinhos". Vinte anos antes, as únicas pessoas em favor dessa abordagem foram alguns programadores de computador desconhecidos, e nem eles poderiam ter previsto que isso revolucionaria a maneira como tudo, desde sistemas operacionais, mapas e até camisetas, é criado.

Contudo, criar um programa de computador é bem diferente de, digamos, desenvolver um mapa de código aberto ou criar um segmento

de notícias em vídeo. O êxito do software de código aberto dependia, em grande parte, da proliferação de tecnologia e software acessíveis. O código aberto forneceu o projeto de como as pessoas deveriam unir-se para trabalhar – de forma competente, entusiástica e sem remuneração – em áreas além de software. Mas, para preencher a lacuna entre a teoria e a prática, a multidão precisaria de ferramentas e de conhecimento para saber usá-las.