

Quando o barato sai caro

Um novo ponto de vista na velha questão software livre x software proprietário

por Ricardo C. Zímmerl'

O conceito de *Free Software* (software livre) é um pouco confuso, por isso devo, primeiramente, esclarecê-lo para que isso se torne mais compreensível no decorrer deste artigo. Software livre não significa, necessariamente, software gratuito, nem tampouco software de domínio público. O software livre é uma forma de licenciamento que usa um conceito chamado **CopyLeft**, ou seja, é divulgado de forma tal que seu código não possa ser “fechado” ou “privatizado” por uma pessoa ou organização. Isso não significa que ele (o software) não possua direito autoral, pelo contrário, mas seus autores devem concordar em distribuir os códigos-fonte e permitir que terceiros possam modificá-los e redistribuí-los sem incluir qualquer restrição a qualquer parte do programa. Para compreendermos melhor o que é esse tal software livre, devemos recordar como e por que ele surgiu.

No início da era digital, os desenvolvedores e programadores eram poucos e faziam parte de uma elite de alto nível. Essa elite compartilhava entre si seu conhecimento e ajudavam-se uns aos outros nos desenvolvimentos de soluções e produtos. Isso vigorou dos anos 50 até o início dos anos 80. Porém, na década de 80, algo mudou. Os softwares tornaram-se um produto comercial de massa, já não tínhamos mais apenas “meia-dúzia” de consumidores corporativos com necessidades muito específicas (e que pagavam **muito bem** pelas soluções que lhes eram desenvolvidas), passamos a ter produtos para venda em larga escala, assim, seus custos de desenvolvimento podiam ser rateados entre os consumidores.

Isso gerou o paradigma do software proprietário, cujo código-fonte é mantido oculto e não documentado para o usuário final. É uma consequência natural e evidente, já que seu produto, agora, não se destinava mais apenas a um único usuário (que geralmente contava com equipes de desenvolvimento) e visava, agora, atender às necessidades variadas e de muitas pessoas que raramente teriam conhecimento técnico para aproveitar esse código-fonte.

Daí houve uma certa perda da forma “romântica” do desenvolvimento de softwares e levou-nos à extrema competitividade, normalmente predatória porém grande incentivadora da inovação e da qualidade no mundo atual.

Um dos que levantou sua voz de forma contrária a esse padrão fechado foi Richard Stallman, autor do projeto GNU, o que acarretou no sistema de licenciamento *open-source/free-software* atualmente mais divulgado e que é objeto deste artigo.

Para Stallman (de acordo com o site www.gnu.org) os programas não deveriam ter restrições ou proprietários porque a tecnologia permite fácil cópia dos mesmos (quem sabe não se pensa em aplicar esse princípio às músicas em formato MP3, já que se trata de direito autoral da mesma forma, não é mesmo?). É um argumento bastante fraco, mas é fundamentalmente sobre ele que é baseada a filosofia do software livre.

Diferenças básicas do Software Livre e Software Proprietário

Talvez você, como eu, já tenha ouvido a frase: “Nosso produto é gratuito pois baseia-se em plataforma livre”. Quem normalmente diz esse tipo de besteira prova desconhecer totalmente o que está falando. Eu sou desenvolvedor e tenho diversos softwares gratuitos em meu site, a despeito de usar uma ferramenta de autoria licenciada para compilá-los. O software ser gratuito ou não, na maioria das vezes, depende apenas do autor ou da empresa que o desenvolveu determinar e o licenciar assim, não tem nada a ver com a plataforma de uso ou de desenvolvimento. As ferramentas proprietárias de desenvolvimento tem custo apenas aos desenvolvedores. Além disso, um software livre pode muito bem ser licenciado de forma onerosa, como vimos no início.

Neste ponto, você, leitor, pode estar achando que sou contra o “software livre”, a verdade é que há uma maré de pessoas dizendo uma porção de bobagens e praticamente ninguém mostrando que as coisas não são bem assim. E, para evitar que você seja mais um dos que são enganados pela “multidão”, quero dar-lhe informações nas quais, ao ponderar, você possa vir a formar um ponto de vista mais acertado.

Primeiramente deixemos de lado esse maniqueísmo: “Software proprietário” é mau, “Software livre” é bom. Nesta vida as coisas nunca são 100% boas ou más, há muitas faces em todas as coisas e saber distingui-las é fundamental para a correção das idéias.

Particularmente, sou contra o conhecimento “fechado”, como se Isaac Newton, ao invés de divulgar sua leis da gravitação, cobrasse para fazer os cálculos a quem quisesse mantendo o conhecimento restrito apenas a si mesmo. O conhecimento deve sempre ser compartilhado, essa é uma forma da humanidade e do próprio conhecimento evoluírem. Aprendemos pelo conhecimento que outros geraram e devemos passar aos demais o conhecimento que geramos. Mas isso não implica no *open source*, e logo você entenderá o porquê.

Voltando às diferenças – a mais marcante é: um software proprietário tem um “dono”, ou seja, alguém responsável por ele (e alguém **que pode ser responsabilizado** por ele), já um software livre não possui nem “dono” nem alguém que possa ser responsabilizado. Aí as coisas podem começar a “feder”!

Riscos e garantias

Quem trabalha na – ou para a – “comunidade” *open-source/free-software* não faz isso com “obrigações” de qualidade, facilidade de uso, operacionalidade nem interoperabilidade. Normalmente os colaboradores procuram ajudar-se mutuamente como um hobby, enquanto outros procuram aproveitar-se do trabalho alheio para se promover, afinal, “todo circo tem um palhaço”.

Um sistema sem dono, pátria ou lei sempre será terra fértil para enganadores, aproveitadores e para toda a sorte de coisas ruins – e esse é o maior problema.

Quem está acostumado com softwares livres já se habituou à expressão “estável” (ex.: “essa é a versão estável 1.04”), mas o que isso quer dizer?

Quer dizer que, aparentemente, aquele sistema está funcionando, pelo menos essencialmente, bem. Não quer dizer que esteja totalmente apurado e livre de *bugs*, apenas que está funcionando.

E se, para você, ele não funcionar? E se ele “der pau” levando todos os arquivos do seu trabalho “para o saco”? A quem reclamar? A ninguém. Mesmo por que, quem fez o sistema não tinha obrigação de fazê-lo perfeito, já que o fez por *hobby*. Ainda mais: “cavalo dado não se olha os dentes”. O sistema foi, na maioria das vezes, fornecido de graça, assim, nem seria justo reclamar e, outro detalhe, o código-fonte está disponível, depure-o e corrija os erros, ué!

Esse é o pensamento dos que defendem o *open source*. Bem simples e prático, não é mesmo? Porém, partindo da realidade que 99,999% da raça humana não é composta por desenvolvedores, e que a maioria das pessoas não possui conhecimento, tempo ou disposição para fazer essa depuração e/ou correção, o uso de softwares livres às vezes é como brincar de roleta russa, e você não tem qualquer garantia de absolutamente nada, tampouco tem a quem reclamar. Alguns desenvolvedores até costumam ser bastante gentis e tentam solucionar os *bugs* que são encontrados, na medida que podem e de forma gratuita. Outros cobram pela assistência (se você é esperto já deve ter imaginado que aí há uma ótima possibilidade para desenvolvedores inescrupulosos auferirem bons lucros).

O próprio CEO da Red Hat, Matthew Szulik, recomendou o uso do Windows para usuários domésticos, já que o Linux só apresenta bons resultados aos usuários corporativos. Sabe por que?

Porque o sistema apresenta tantos problemas e complexidades que, para usá-lo de forma eficaz, você precisa ter administradores de alta especialização, capazes de configurar e depurar o sistema eles próprios. Assim, uma empresa economiza, digamos R\$ 300,00 na licença do Windows, mas tem que gastar muito mais todo o mês com um técnico capaz de operar o Linux (isso, é claro, sem qualquer garantia).

Muitas empresas se aproveitam desse fator para distribuir “pacotes” de softwares livres vendendo o suporte ao produto. Parece um bom negócio, não é mesmo? E realmente o é, mas para quem? Será para o usuário final ou para a empresa que fornece o suporte?

Enfim, o software livre acaba sendo uma ótima maneira para “espertinhos”, que não tem capacidade para desenvolver softwares, lucrarem com o mercado de desenvolvimento. É óbvio que,

se os softwares não derem problema, o usuário não precisará do suporte, por isso, eles sempre acabam dando alguma “zica”.

Já o software proprietário, embora geralmente também não dêem garantias (leia com atenção as licenças de uso e verá que os programas são sempre fornecidos a você com a expressão “como está” ou “As is”, o que significa, se der problema: “poxa, foi mau!” e enfatizam “sem qualquer garantia, expressa ou implícita”), porém, têm padrões de qualidade e de confecção estabelecidos. Uma empresa possui um nome a zelar, seus produtos não podem apresentar grandes falhas ou a empresa acaba ficando com uma imagem negativa no mercado, por isso todos os esforços são feitos visando reduzir falhas e dar aos produtos a máxima qualidade. Não por que elas sejam boas, mas por que elas não são bobas! E a concorrência existe e é impiedosa com quem não é competente – podemos considerar que, se o Windows não fosse tão vulnerável, praticamente ninguém adotaria outros sistemas.

Mas esse esquema de distribuir um software “de graça” e cobrar pelo suporte também vem sendo adotado por empresas de software proprietário, e o motivo é simples: a licença, normalmente, é cobrada apenas uma vez, já o suporte é mensal, dá lucro todo o mês. Mas, é claro, o software tem que dar problemas, do contrário o usuário irá dispensar o suporte. A empresa não faz isso por ser boa ou má, ela tem que funcionar, ter lucro e essa é uma forma de fazê-lo ao mesmo tempo em que neutraliza parte do problema da pirataria, ainda que o usuário esteja sendo prejudicado. Já as empresas que cobram o licenciamento e oferecem suporte gratuito devem produzir softwares com máxima qualidade para evitar que o usuário venha precisar recorrer a esse suporte, pois ele representa um custo adicional à empresa.

Outra coisa que atrapalha a qualidade do software livre é a falta de estrutura de desenvolvimento, de padrão e a descentralização. Quando muitos mexem em algo, já dizia minha avó, acaba saindo porcaria. Cada desenvolvedor, cá e lá, introduz modificações nos códigos. Cada qual tem seu jeito particular de pensar, agir e nível diverso de conhecimento, essas coisas serão refletidas na estrutura lógica do código-fonte gerado.

Como não há um modo de desenvolvimento definido (embora em alguns casos até haja, como nos mostra o **SourceForge** (<http://sourceforge.net>), já que esse é um problema sempre grave), o produto final não possui coesão e muitos erros daí decorrem (os famosos “paus”). Mas, mesmo nos casos em que há certa estrutura no desenvolvimento, como pessoas, mundo afora, podem fazer modificações e normalmente não há centralização, você não sabe se o produto que você está instalando possui *bugs* já detectados e corrigidos por outros e nem sabe se alguém não fez uma versão mais adequada ou mais “estável”, conseqüentemente, acaba assumindo riscos desnecessários.

E há, ainda, os limites técnicos. Um bom desenvolvedor (com larga experiência) dificilmente consegue fazer um programa com 10.000 linhas de código sem que ocorram pequenos bugs e lapsos. Isso se dá porque os humanos não são máquinas (embora alguns empresários pensem que seus funcionários sejam), e o nível de concentração, atenção e o lado emocional não são constantes todos os dias, portanto, a memória também costuma falhar. Um software grande, como um *Excel* ou mesmo um software para automação de empresas costuma ter dezenas ou centenas de milhares de linhas de código. Esperar que um desenvolvedor ou um pequeno grupo deles possa fazer, por *hobby*, algo dessa magnitude sem que surjam inúmeros problemas é, realmente, sandice.

Custo embutido: valor do software livre

Quando uma empresa desenvolve um programa, há diversos custos que ela tem que considerar: o salário dos desenvolvedores, da equipe de controle de qualidade, do programa de testes, da aquisição ou desenvolvimento de tecnologias, da produção de material de suporte e da própria equipe de suporte ao produto. Isso tudo é rateado na forma do valor da licença final do produto desenvolvido, que será tanto maior quanto menor for o número final estimado de licenciados. Por isso, um sistema operacional complexo como o Windows tem uma licença de poucas centenas de dólares e um produto muito mais simples, como um programa de CAD, pode ter sua licença orçada em vários milhares de dólares.

Como disse anteriormente, uma empresa, organização ou governo, ao adotar o software livre, geralmente precisa ter pessoal qualificado para usá-lo, depurá-lo e corrigi-lo. Isso significa que parte do custo de desenvolvimento, teste e suporte deverá ser arcado por cada usuário de forma **unitária**.

Desta feita, para escapar de um custo com licenças acaba-se com custos muito maiores, redundantes, e, em alguns casos, com resultados de pior qualidade!

Por mais que se queira acreditar no contrário, programação é um artesanato! Nem todos os que fazem cursos de programação tornam-se programadores, nem todos os que se tornam

programadores são bons profissionais; da mesma maneira que nem todos que fazem curso de culinária se dão bem na cozinha e nem todos os que sabem cozinhar são *gourmets*.

Bons programadores são raros e são disputados em todo o mundo pelas grandes desenvolvedoras de softwares, ganhando bons salários em ótimas condições de trabalho ou montam suas próprias empresas para lidar com produtos por eles desenvolvidos. O que geralmente se encontra no resto do mercado são os programadores médios ou aqueles que não chegam nem a isso. Uma empresa, portanto, ao contratar uma pessoa qualificada para operar com software livre, evidentemente, dificilmente estará contratando o que há de melhor (nem teria disposição financeira para isso), mas, sim, o que há disponível.

Outro detalhe, importante para empresários, reside no valor agregado e valor patrimonial. Uma licença de software proprietário é parte do patrimônio da empresa, pode ser medido financeiramente. Um software livre, ainda que a empresa tenha gasto “horrores” com sua adaptação, depuração e testes, não vale nada nem agrega coisa alguma ao patrimônio da empresa. É, praticamente, dinheiro jogado fora.

Quem perde com o software livre?

O governo geralmente combate a pirataria porque ela gera evasão de impostos: produtos piratas não pagam impostos! E produtos gratuitos pagam? Ao optar por sistemas livres, os governos acabam dando “um tiro no pé”, primeiro por aumentar seus gastos com pessoal mais “qualificado”, segundo por fomentar produtos que não pagam impostos, mesmo que não sejam piratas. O que é um contra-senso ao combater a pirataria.

Uma coisa que as pessoas não percebem é que o software livre destrói a capacidade dos países em desenvolvimento de ter uma boa indústria de softwares.

Por quê? Veja, se você tem duas opções para um mesmo produto: uma gratuita, mesmo estrangeira, e uma paga, ainda que nacional, indiferente ao fator qualidade, a maioria das pessoas optará (se não for piratear, é claro), pelo gratuito. A indústria norte americana é sempre protegida por seu governo, já que gera empregos, tecnologia e impostos lá, por isso, os sistemas livres não são ameaça substancial, por que, se forem, o governo de lá dá um jeito, mas, para os demais países, tornam quase impossível o surgimento de empresas sólidas. Mesmo no Brasil as empresas que conseguem alguma rentabilidade são aquelas que trabalham ou com soluções corporativas (para poucos clientes que pagam bem) ou com áreas nas quais o software livre e também o software gratuito ainda não possui destaque.

Uma evidência de como softwares gratuitos, sejam livres ou proprietários, podem impedir ou dificultar o desenvolvimento da indústria é patente no caso *Internet Explorer versus Netscape Navigator* (ao disponibilizar de forma gratuita o Internet Explorer a Microsoft praticamente liquidou a rival e anterior líder em navegadores para internet).

Outro problema é com relação ao desenvolvimento educacional e tecnológico do país. Praticamente todas as pessoas, se dados dois caminhos, um fácil, outro difícil, optarão pelo que é fácil. Não é necessário ser gênio para perceber isso.

Entre você ter que desenvolver produtos e tecnologias ou só “dar uma garibada” em um que já está feito, quase todos optarão pelo segundo.

Quando, em vez de ter que exercitar o cérebro (e os dedos) para criar algo a partir do zero, você pode contar com uma base já pronta e funcional para fundamentar o desenvolvimento de um produto, o que você escolherá?

Alguns dizem que isso, desenvolver a partir do zero, é contraproducente, o que, absolutamente, não é verdade e eu exemplifico com experiência própria. Tenho um produto chamado *CdC* (Cadastro de Clientes) que é gratuito e está disponível em meu site para download (www.itabra.com). Quando surgiu a primeira versão, ela foi feita sobre um módulo correspondente desenvolvido para um programa meu de automação de empresa chamado *ZW-Gestor*. Essa adaptação demorou um tempo considerável porque tinha que se desvencilhar das demais partes do programa e a versão final apresentava vários *bugs* por estar operando dessa forma isolada, o que levou um outro tanto de tempo para a depuração. Depois de alguns meses criei outro produto, o *C3* (Cadastro Comercial de Clientes), porém – aprendida a lição – dessa vez a partir do zero. Embora ele seja similar na operação e aspecto ao *CdC*, esse novo produto possui mais recursos e tecnologia central muito superior tendo, contudo, levado menos de um quinto do tempo necessário para desenvolver o *CdC*. Várias empresas também tiveram problemas semelhantes. O inchaço de versões pré-existentes incorporando novas funções é uma prática que a maioria das grandes empresas de

desenvolvimento evita o quanto possível, em benefício de produtos mais leves, confiáveis e melhores. É o preço da qualidade.

Desenvolver a partir de coisas já existentes acaba levando a uma falta de real conhecimento, já que você não mais se preocupa com o todo, só com o que quer fazer. Gera desenvolvedores menos capazes, acostumados a facilidades (ou “preguiçosos”), e dificulta ao país ter condições (recursos humanos qualificados) para desenvolver tecnologia.

Alguns, possivelmente desprovidos de conhecimentos acerca do assunto, dizem que a vantagem do software livre é eliminar as “caixas-pretas” tecnológicas embutidas nos softwares. Parece piada, no meu ponto de vista. Que caixas-pretas?

As mesmas tecnologias que a Microsoft, Sun, Oracle ou qualquer outro usam estão disponíveis para qualquer um que tenha vontade de conhecê-las. Todos os protocolos, formatos de arquivo, formas de programação, técnicas e comandos podem ser encontrados em livros, sites, fóruns na Internet, enfim, estão disponíveis a todos de forma igual em qualquer canto do mundo.

Não existem tais caixas-pretas! Isso eu posso lhes assegurar com toda a convicção! Quem fala que existem não sabe o que diz.

Outros argumentam, ainda, em relação à segurança pessoal. Como você pode saber que um software proprietário, cujo código é fechado, não está espionando você ou se não pode sabotar, via controle remoto, sua empresa ou trabalho? Se você tem o código-fonte do software você pode saber tudo o que ele faz. Ótimo argumento, mas baseia-se em pressupostos errados.

Primeiro, ter o código-fonte não significa que o programa que você roda é exatamente aquele cujo código-fonte está anexo! Quem quer fazer coisa errada costuma disfarçar isso e a maioria das pessoas não vai recompilar o programa, embora, para ter realmente segurança, devesse fazê-lo.

Segundo, a maioria das pessoas não entende de programação, assim, mesmo que o código-fonte inclua rotinas para abertura de portas e servidor para comandos remotos, elas não serão capazes de entender isso, ainda que se dêem ao trabalho de lê-lo.

Terceiro, empresas têm nomes a zelar e concorrência predatória. Se uma empresa produz um software que causa danos, ela pode ser processada e ter que indenizar o usuário lesado; se um software livre fizer isso, quem pagará? Os autores podem alegar que a parte que lhe causou dano não é de autoria deles ou dizer que você deveria ter visto isso no código e corrigido, afinal, o código-fonte vem anexo para isso, não é mesmo?

Em relação ao software-proprietário, não há código secreto. Se você obteve um programa de fonte legal, autorizada pelo desenvolvedor, está licenciado e teve problemas, pode solicitar à justiça a análise do código-fonte, se acha que o programa foi responsável, e esse programa poderá ser “descompilado” para verificar se o autor agiu de má fé incluindo no software rotinas ou elementos que pudessem causar o dano.

A sociedade também tem certa perda com o software livre, não apenas pela perda de impostos, de empregos na indústria de softwares e desenvolvimento de tecnologia, mas, pelo fato de que os “espertinhos”, “malandros”, aqueles que se locupletam do trabalho alheio ou com os problemas que usuário pode vir a ter é que são os maiores beneficiados com esse tipo de sistema de licenciamento.

Favorecem-se pessoas e empresas que ganham com o caos tecnológico, aproveitando-se que a maioria dos usuários não tem tempo ou disposição para aprender coisas demasiado técnicas para impingir-lhes produtos sem garantia, às vezes de qualidade duvidosa, sob um rótulo de liberdade e combate aos “tubarões” internacionais, garantindo uma segurança impossível de se ter e que, quando dá errado não se pode culpar ninguém.

Os softwares gratuitos, diferentemente do software livre, normalmente possuem autor e, por terem seu código fechado, dificultam fraudes, garantindo maior segurança e, por serem centralizados, você pode verificar qual a última versão disponível, além de oferecer maior garantia ao pegar uma cópia direto da fonte, ainda que suas licenças nada custem. Aí, se o software será licenciado de forma gratuita ou não, parte da política do desenvolvedor. Muitas empresas adotam um sistema misto bem satisfatório: licença gratuita para usuários domésticos e paga, para usuários corporativos.

A difusão de parte de códigos-fonte de programas (principalmente de programas proprietários) em universidades, centros de ensino e desenvolvimento ou em sites para interessados também é algo saudável, permitindo contribuições para melhoria desses códigos através do surgimento de novas idéias, conceitos ou funções, ainda que sejam sujeitas ao crivo e rigor técnico da empresa para eventual adoção ao sistema original.

O suposto direito alegado pelos defensores do software livre de acesso ao código-fonte pelas pessoas seria equivalente ao direito das pessoas saberem como um medicamento é produzido. A maior parte da população não só não tem o conhecimento técnico para aproveitar esse “direito” como

não dispõe de tempo ou interesse para poder tirar proveito disso. Ademais, as pessoas não são obrigadas a serem especialistas em informática ainda que, em seus trabalhos, sejam obrigadas a serem usuárias dela. Se eu tenho que escolher entre um produto “feito sabe lá Deus como” mas que vem com seu código-fonte de modo a que eu possa corrigir quaisquer erros e deixá-lo do meu jeito ou um produto com qualidade, profissional, eu optarei pelo segundo, ainda que possa me custar mais “dinheiro” inicialmente, porque irei poupar tempo considerável e ônus ao longo da vida útil desse produto.

Os softwares proprietários, via de regra, possuem suporte gratuito por parte dos desenvolvedores aos usuários licenciados, e isso pode poupar-lhes montantes muitas vezes superiores ao valor da licença.

É importante, pois, que você, leitor, perceba a falácia e artifícios que os defensores do software livre usam, principalmente porque muitos deles vão faturar bastante com o suporte àqueles que lhes seguirem às cegas e que, irremediavelmente, acabarão como presas em suas teias de ilusões.

A tese de Richard Stallman em defesa do software sem dono é fraca e frágil. Pelo princípio que ele defende, de que a tecnologia favorece a cópia fazendo-a algo natural e que, portanto, os materiais copiáveis não devem ter donos, poderíamos imprimir nosso próprio dinheiro (as impressoras laser tem ótima definição para isso), copiar músicas, livros, etc. Também é fácil cultivar maconha, fazer bombas, matar pessoas, é mais fácil roubar que trabalhar. Se o fato de uma coisa ser fácil e simples torna-a natural e por isso deve ser lícita, então queimemos as leis, vivamos no completo caos, o que é um evidente absurdo!

Fora dessa discussão filosófica, o fato é que o software livre pode oferecer graves riscos, desvantagens e custos excessivos àqueles que os adotam, ao desenvolvimento tecnológico e nacional. Assim, ao pensar em qual opção adotar para sua empresa ou em seu computador doméstico, reflita nestes fatos, nas garantias que você pode ter, no suporte pago ou gratuito e no valor que agregará com sua escolha. Muitos defendem o software livre quase como se isso fosse uma religião. Porém repare na verdade por trás das coisas e nas múltiplas facetas da realidade. O sistema de licenciamento free-software/GNU nos abriu novas portas e perspectivas, permitiu melhor discussão em relação à propriedade intelectual, direitos autorais, forneceu novas opções de acesso aos benefícios tecnológicos e contribuiu com a melhoria da qualidade dos produtos de informática apresentando e popularizando versões alternativas aos padrões da indústria. Abriu portas para desenvolvedores desconhecidos, democratizou o acesso à “elite digital”, porém abriu brechas que podem e são usadas por pessoas mal-intencionadas, eliminou as garantias para o usuário final, adotado em larga escala dificulta o surgimento de empresas de produção de softwares em países em desenvolvimento e favorece os desenvolvedores preguiçosos e empresas mal-intencionadas a aparecerem sobre o trabalho alheio. Como ferramenta de inovação (e de difusão da mesma), como fomento de uma sociedade mais democrática ligada à tecnologia informática, o “open-source” não tem rival, mas, para aplicação concreta em empresas, governos e para o usuário comum, é fonte de riscos substanciais.

Pense nisso.