

# Software Aberto: O Futuro da Engenharia Informática?

António Coutinho, Carlos Baquero, Francisco Moura, Victor Fonte

Grupo de Sistemas Distribuídos

<http://gsd.di.uminho.pt/>

*Departamento de Informática, Universidade do Minho*

## 1 Introdução

Uma das questões mais inquietantes que se podem colocar a qualquer classe profissional é a da sua própria relevância. Apesar de poder parecer um contra-senso, esta inquietação faz sentido no campo da Engenharia Informática, em particular no caso do software.

Neste contexto, o maior factor de irrelevância dos engenheiros é o progressivo desaparecimento das oportunidades para a sua intervenção. Isto resulta de uma transferência da sua capacidade decisória sobre o funcionamento dos sistemas e aplicações em favor das grandes companhias mundiais que produzem o software e em detrimento dos utilizadores, incluindo os que contratariam engenheiros.

Temos assistido nos últimos anos a um nivelamento por baixo da interação entre os utilizadores e o software, em que a estes é cada vez menos dada possibilidade de escolha, em troca de uma aparente simplicidade. Tudo funciona bem apenas se se seguir o caminho recomendado pelo fabricante.

Esta tendência seria ainda minimizada se o comportamento do software fosse planeado atendendo apenas a questões técnicas. Acontece porém que na maior parte das vezes esse planeamento obedece mais a imperativos de marketing ou domínio de mercado do que a considerações sobre as reais necessidades dos utilizadores. Há até um caso célebre em que uma empresa alterou as características de um produto seu, supostamente por motivos técnicos e para favorecer os utilizadores. Uma recente conclusão judicial<sup>1</sup> demonstrou que essas estratégias consubstanciaram práticas anti-competitivas que pouco têm a ver com o interesse dos consumidores.

Outro tipo de limitação técnica motivada por imperativos comerciais é o uso de protocolos e formatos de documentos fechados, ie. cujas especificações técnicas

não são integralmente divulgadas, o que põe entraves à inter-operabilidade dos programas que não pertençam à empresa dominante. Para piorar a situação, estes formatos e protocolos sofrem modificações e tornam-se incompatíveis em cada nova versão do software.

Este tipo de opções segue uma ética completamente oposta à da Engenharia, e não admira que os produtos resultantes tornem difícil a intervenção creativa dos engenheiros.

Uma ameaça ainda mais directa é a desvalorização dos nossos currícula, ricos em conhecimentos duradouros e com um certo grau de generalismo, em favor de certificados muito específicos criados e geridos pelas próprias empresas produtoras de software. No caso dos programas de certificação de várias empresas é preocupante a tendência para valorizar alguns desses títulos mais do que licenciaturas em engenharia. Tais certificados incidem normalmente sobre peculiaridades de determinadas tecnologias ou pacotes de software, tornando-se rapidamente obsoletos. Por analogia, se o mesmo se passasse no ramo da Engenharia Civil seria necessário ter uma certificação especial para poder usar uma determinada marca de ferramentas, mas a quem tivesse essa certificação seria até permitido projectar casas inteiras.

Esta situação não se deve atribuir ao carácter mais ou menos maligno das empresas. Estas apenas cumprem a sua função, que é gerar lucros. O que se passa é que o software é um negócio com propriedades muito especiais, em que o desfecho mais provável é uma só empresa ficar com o monopólio dos programas e aplicações de carácter genérico. Essa empresa pode então impôr as suas regras da forma que maximize o seu lucro e as perspectivas de manter a sua posição dominante, o que resulta no cenário pouco animador descrito nos parágrafos anteriores.

A justificação é simples: o software de uso genérico (sistemas operativos, *browsers*, *office suites*) tem um custo elevadíssimo associado ao seu desenvolvimento, mas um custo desprezável na sua duplicação em mas-

<sup>1</sup>Documento de um tribunal federal dos EUA. *Findings of Fact. USA, Plaintiff, vs Microsoft Corporation, Defendant.* (<http://usvms.gpo.gov/findfact.html>)

sa. Assim uma empresa que consiga dominar o mercado está em condições de tirar partido das inúmeras cópias que vende, enquanto um possível concorrente tem que se debater com o enorme custo inicial de desenvolver o software.

Ainda mais insidioso é o fenómeno caracterizado no documento judicial já citado<sup>2</sup> como *positive network effects*. Este fenómeno é descrito como o que ocorre quando um produto se torna mais atractivo quanto maior é o número de pessoas que o usa. Um exemplo claro é o dos processadores de texto, em que a pressão social para obrigar ao uso do formato dominante atinge proporções virtualmente inimagináveis em qualquer outro tipo de produto.

Está talvez na altura de nos interrogarmos se as coisas poderiam ser de outro modo. O principal tema deste artigo é precisamente a afirmação da resposta positiva a esta pergunta.

A conclusão que defendemos é que o modelo do software estritamente comercial, em que se paga por cada cópia e em que o funcionamento dos programas é um segredo das empresas que os desenvolvem, não pode ser usado para sistemas operativos e aplicações de índole muito genérica, sem grandes prejuízos para os consumidores.

A alternativa é uma filosofia completamente diferente: o Software Aberto. Veremos nas secções seguintes uma caracterização deste tipo de programas, bem como os efeitos benéficos que já está a ter em relação à valorização das carreiras dos engenheiros informáticos, permitindo até reduzir o atraso e subserviência que têm caracterizado a relação da indústria de software europeia com a americana.

## 2 Software Aberto

A noção de “*software aberto*” é bastante simples e pode ser entendida, na sua essência, como traduzindo-se na garantia de um conjunto de direitos ao utilizador do software – a possibilidade de acesso e modificação do código fonte dos programas e a possibilidade de redistribuição das alterações por ele realizadas. Este conjunto de direitos está expresso sob a forma de uma licença de distribuição incluída sempre no próprio software.

Como facilmente se pode depreender, a questão da protecção do utilizador enquanto consumidor de software está subjacente ao conceito de software aberto. Ao contrário do que ocorre no software fechado, o utilizador passa a poder intervir activamente na evolução do software do qual depende ou simplesmente

aprecia. Na verdade, na génese deste conceito está o projecto GNU iniciado na década de 70 por Richard Stallman<sup>3</sup> e que surgiu como uma reacção ao fecho do código fonte do sistema operativo Unix. Nessa altura, não só as inúmeras contribuições voluntárias realizados pelos utilizadores deste sistema foram apropriadas pela AT&T, como também lhes foi negado o crédito, o livre uso e a possibilidade de influenciarem a evolução do software por eles próprios desenvolvido. O projecto GNU emergiu precisamente como uma tentativa de se constituir uma alternativa aberta ao sistema operativo Unix e como prometendo devolver o protagonismo aos seus utilizadores. Apesar das dificuldades tecnológicas de então e das vicissitudes e polémicas associadas a este projecto, as ideias de Stallman acabaram por se impor anos mais tarde, numa clara vitória em toda a linha.

Se por um lado é óbvia a preocupação em proteger o utilizador, garantindo-lhe a possibilidade de influenciar a evolução do software e dotando-o de maior independência face à produção do mesmo, a tese da superioridade qualitativa do software aberto tal como é defendida pela sua comunidade pode, no entanto, ser bem menos evidente. Contudo, são várias e facilmente explicáveis as razões que convergem na sustentação desta afirmação.

Um dos factores que mais contribui para a qualidade deste tipo de software prende-se com a transparência e a honestidade de todo o processo de desenvolvimento – a possibilidade do livre acesso ao código fonte permite o “*peer review*”, ou seja, a crítica independente dos pares. Deste modo, o trabalho realizado pelos programadores, quer na produção do software original quer na introdução de modificações, fica sempre sujeito à apreciação de todos os programadores e utilizadores interessados. Este procedimento, pouco frequente na indústria de software, tem desde sempre caracterizado o trabalho realizado no campo científico, com evidentes provas dadas. Independentemente do campo de trabalho, a exposição à crítica pública estimula, como se compreende, a procura da qualidade e a procura do reconhecimento e respeito dos pares.

A possibilidade de modificação do código fonte do software permite, por seu lado, a correcção de eventuais erros, a sua adaptação a situações não previstas inicialmente, bem como a produção de novo software com base no original. Estas modificações podem ser

<sup>2</sup>Findings of Fact, item 32, pág 18.

<sup>3</sup>Richard Stallman é uma figura fundamental para o movimento do software aberto, tendo sido o precursor deste conceito, o fundador do projecto GNU (que pretende ser uma alternativa aberta ao sistema operativo Unix), autor de grande parte do compilador deste projecto, redactor da GNU General Public License e presidente da Free Software Foundation.

realizadas independentemente por qualquer utilizador mesmo sem a anuência explícita do autor original. O utilizador pode desta forma melhorar a qualidade do software do qual faz uso e adaptá-lo às suas necessidades específicas.

Duas outras razões relacionadas com a comunidade de software aberto justificam ainda a sua qualidade. Uma delas é que o forte sentimento de pertença e de respeito à comunidade estimula o *feedback* dos utilizadores e a evolução do software. No entanto, talvez a razão mais importante seja a que se prende com uma característica quase só presente na comunidade de software aberto. Ao contrário do que acontece no desenvolvimento de software fechado, na sua esmagadora maioria os programadores são também utilizadores finais do software que desenvolvem. Dificilmente poderia haver maior motivação para a qualidade do que o impacto directo da sua própria utilização.

## 2.1 A retribuição à comunidade

Se é verdade que as licenças de distribuição de software aberto garantem a possibilidade de acesso e modificação do código fonte, a redistribuição dessas alterações não é necessariamente obrigatória. A definição<sup>4</sup> de software aberto apenas garante a possibilidade dos programadores partilharem as suas alterações com a comunidade, se assim o entenderem. Isto significa que, de acordo com a definição básica deste software, pode ser legítimo o fecho do seu código fonte. No entanto, a eventual evolução fechada de um ramo do código fonte não impede nunca a contínua evolução aberta desse software. Além disso, existe a convicção no seio da comunidade de software aberto que é mais difícil manter o desenvolvimento fechado do software, assegurar a sua qualidade e inovar o produto, pelo que, a prazo e de uma forma geral, revelar-se-á como uma decisão errada.

Como se compreende, esta possibilidade, defendida por parte da comunidade de software aberto, legitima a apropriação das contribuições gratuitas e voluntárias dos seus membros. Tem no entanto vindo a impor-se o sentimento de que estas situações são moral e eticamente reprováveis e que podem prejudicar a dinâmica de evolução deste tipo de software. Uma possível solução passa pela adopção de licenças de distribuição que imponham igualmente a redistribuição de modificações sob a forma software aberto.

<sup>4</sup>Em <http://www.opensource.org/osd.html> pode consultar-se a definição de “*software aberto*” nos termos propostos pelo consórcio Open-Source Initiative (OSI), entidade reconhecida como tendo a legitimidade para certificar ou não o carácter de abertura das diferentes licenças de distribuição de software.

Hoje em dia, a licença de distribuição mais adoptada é certamente a GPL<sup>5</sup> do próprio Stallman. A GPL não só garante os direitos inerentes ao software aberto mas também impõe a transmissão desses mesmos direitos na redistribuição do software modificado ou derivado ao estabelecer direitos públicos de propriedade intelectual. Mais, a GPL nega o direito ao registo de patentes sobre esse software salvo se registadas em domínio público.

## 2.2 O desenvolvimento em bazar

Uma das razões para as dificuldades iniciais do projecto GNU prendia-se com a pouca interacção entre os membros do projecto, tanto programadores como utilizadores. Até à generalização do acesso à Internet a distribuição era essencialmente realizada por envio de *tapes* suportando o utilizador todos os custos daí inerentes. Com a Internet, a comunicação entre os programadores e os utilizadores tornou-se muito mais frequente, e a própria distribuição de software muito mais simples e barata.

Esta interactividade teve uma consequência não prevista nas ideias iniciais de Stallman. A possibilidade de modificação concorrente do código fonte do software por todos os interessados, cada qual com a sua agenda e com as suas necessidades específicas. O enorme sucesso do kernel do sistema operativo Linux talvez tenha sido o primeiro dos projectos de software aberto que tornou evidente as vantagens da descentralização do processo de desenvolvimento. Segundo Eric Raymond<sup>6</sup> este modelo de desenvolvimento, quase caótico, aproxima-se da noção de bazar em que cada qual tem os seus próprios objectivos. Por contraponto, o tradicional modelo centralizado de desenvolvimento de software pode ser comparado a uma catedral, no sentido em que existe uma hierarquia em que um reduzido número de pessoas tomam as decisões sobre a evolução do projecto. Note-se que o modelo de “catedral” não é exclusivo do desenvolvimento de software comercial. Na verdade, parte da explicação para as dificuldades iniciais do projecto GNU residiam da centralização do seu desenvolvimento, fruto das dificuldades de interacção entre os intervenientes.

A dinâmica inerente ao software aberto é muitas vezes encarada como uma característica indesejável, repetindo-se com frequência a crítica de que este software é por natureza instável e sem qualidade. Esta é normalmente uma crítica injusta. Verifica-se que, na

<sup>5</sup><http://www.opensource.org/licenses/gpl-license.html>

<sup>6</sup>Eric Raymond é um dos actuais teóricos do movimento de software aberto, tendo sido o primeiro a explicar as vantagens da descentralização do desenvolvimento de software.

prática, apenas as inovações comprovadamente úteis e o software melhor adaptado consegue mobilizar a massa crítica de utilizadores e programadores que lhe asseguram a sobrevivência. Ou seja, tal como na natureza, no mundo do software aberto apenas o mais adaptado sobrevive. A analogia com a natureza é ainda válida no sentido em que o ambiente é extremamente hostil e dinâmico. As vantagens competitivas são sempre muito temporárias uma vez que o processo de desenvolvimento é transparente. Aquilo que hoje é apresentado como inovação, é apresentado amanhã como solução adoptada por todos os potenciais concorrentes.

Uma outra ordem de razões justifica a qualidade e estabilidade do software aberto bem sucedido. Em quase todos os casos de sucesso é mantida uma distribuição estável e consolidada, e uma distribuição de desenvolvimento em que as propostas de alteração mais profundas são experimentadas<sup>7</sup>. Apenas os utilizadores mais afoitos, curiosos ou que participem activamente no processo de desenvolvimento necessitam de experimentar as versões de desenvolvimento. Todos os outros utilizadores usufruem das versões consolidadas.

### 2.3 Um novo modelo de negócio

A par da expansão do software aberto tem surgido um novo mercado onde a tónica se coloca na prestação de valor acrescentado sobre a plataforma de software aberto. Na área da distribuição, assistiu-se por exemplo à explosão da capitalização bolsista de empresas como a Red Hat, cujo negócio assenta na distribuição do sistema operativo Linux juntamente com ferramentas de suporte à sua instalação.

As possibilidades de configuração não se esgotam de modo algum nestas abordagens horizontais de apoio à instalação. A disponibilidade de uma vasta gama de software configurável que se encontra para além das distribuições usuais abre caminho à prestação de serviços qualificados e à composição de soluções personalizadas. Ao contrário dos modelos de software fechado, não existe aqui limite à capacidade de adaptação a necessidades particulares. Abre-se assim uma área de intervenção onde pode ser valorizado o trabalho dos engenheiros informáticos e onde se garante a durabilidade das soluções encontradas.

De facto, a durabilidade das soluções é em si um factor de imensa relevância para as organizações cuja dimensão já justifica a manutenção de equipas de

<sup>7</sup>A título de exemplo refere-se a variante Raw Hide da distribuição Red Hat Linux, composta por versões ainda instáveis do *kernel* e de outros pacotes de software.

informática. Nestes casos a adopção de um modelo de software aberto permite tirar partido do potencial técnico já existente e quebrar a dependência das soluções já em operação, em relação às políticas dos produtores de software.

É do conhecimento geral que não são raras as situações em que muitas organizações têm de actualizar os seus sistemas operativos e mesmo o parque de hardware, para solucionar problemas de compatibilidade de versões que por vezes apenas obedecem à lógica de mercado dos produtores de software. Por natureza, o software aberto é imune a este tipo de distorções. Com um suporte de software aberto nada impede que soluções próprias que mantenham a sua validade técnica sejam mantidas em operação por um período indeterminado.

## 3 O papel da Europa

Partindo para uma análise geográfica do fenómeno, toma-se por exemplo o caso do Linux. Dos elementos presentes nos mapas de software de Linux é possível fazer uma estatística rápida sobre a distribuição geográfica dos seus autores. Observando-se os endereços de correio electrónico dos autores dos 3483 programas registados no arquivo Metalab<sup>8</sup> pode construir-se a distribuição expressa na Figura 1.

países europeus	37%
.com	23%
.edu	12%
.net	10%
.org	7%
outros	11%

Figura 1: Distribuição dos endereços de correio electrónico no arquivo Metalab.

Sendo pacífico atribuir ao EUA todos os *.edu*, já não é possível atribuir-lhes a totalidade dos *.com*, *.net* e *.org*. Assim, pode-se concluir com uma razoável segurança que os europeus são maioritários no desenvolvimento de programas software aberto, ou pelo menos no universo Linux.

Pode rapidamente constatar-se a relevância desta situação para o esforço europeu em tomar medidas que permitam um reforço da sua influência em diversos sectores da economia mundial, e em particular no sector da informática onde a batalha estava aparentemente perdida. A solução aparenta ser óbvia e passa

<sup>8</sup>O arquivo Metalab sucede ao antigo Sunsite.

pelo apoio concertado ao desenvolvimento e uso de software aberto na Europa.

Um caso paradigmático é o da França, um país tradicionalmente empenhado na manutenção do protagonismo da Europa, e que cedo percebeu as vantagens na adopção de soluções de software aberto. Não sendo de excluir alguma influência dos escritos<sup>9</sup> do investigador Roberto Di Cosmo e da jornalista Dominique Nora, ambos franceses, onde foram expostos os perigos da adopção de formatos e protocolos fechados e as vantagens em a evitar, surge muito recentemente uma proposta de lei nesse sentido<sup>10</sup>.

A proposta de lei principia por mostrar o interesse na adopção de meios de comunicação e registo digital para a simplificação dos processos administrativos, com as inerentes vantagens para a economia, bem como fomentar o aprofundamento da democracia com uma maior transparência e facilidade de interação entre os cidadãos e os centros de poder. Pode-se observar que estes pontos estão na mesma linha dos presentes nas directivas portuguesas para a Sociedade de Informação<sup>11</sup>.

Ao clarificar as técnicas de registo e processamento digital, a proposta francesa argumenta que não é desejável que os meios usados dependam da boa vontade ou dos interesses dos fabricantes de software, pelo que indica como regra geral o uso de software aberto e de protocolos abertos. O recurso a software fechado é assim relegado para as situações onde não existam alternativas abertas.

A discussão que esta proposta de lei motivou veio a ser enriquecida por uma intervenção do coordenador da informática do Ministério da Cultura e Comunicação, que justifica o início de uma vasta migração da sua infra-estrutura<sup>12</sup> para a plataforma Linux. Os motivos apontados foram os seguintes:

- Económicos - Na preparação dos postos, a nível do material, software e suporte.
- Técnicos - O Linux é um sistema de exploração extremamente fiável, robusto e eficiente.
- Segurança - O acesso ao código fonte permite averiguar o seu bom comportamento e a in-

<sup>9</sup>*Le Hold-up Planétaire - la face cachée de Microsoft.* Edição portuguesa pela Terramar em Janeiro de 1999.

<sup>10</sup>*Proposition de loi numéro 495 tendent à généraliser dans l'administration l'usage d'Internet et de logiciels libres* (<http://www.senat.fr/grp/rdse/page/forum/index.htm>).

<sup>11</sup>Tal como no Livro Verde para a Sociedade da Informação (<http://www.missao-si.mct.pt/livro.verde/>) e no documento Portugal na Sociedade da Informação (<http://www.mct.pt/PtSocInfo/indice.htm>) do MCT.

<sup>12</sup>Trata-se de uma rede geograficamente distribuída por 200 locais e que integra 5300 máquinas da classe PC.

xistência de risco de envio ilícito de informação para agências estrangeiras.

- Políticos - Não se cria dependências de monopólios de facto.
- Humanos - Valoriza-se o trabalho dos técnicos de informática.

No reverso da medalha refere-se em particular a dificuldade em contratar engenheiros informáticos com boas competências no software aberto face à grande procura no mercado e à baixa oferta remuneratória na função pública. O testemunho acaba com uma frase relevante no nosso contexto: “*Donc l'administration n'est pas nécessairement une vieille dame endormie*”.

Pode-se interpretar estas propostas como o reconhecimento da importância em se proceder à construção de infra-estruturas digitais nacionais com base em ferramentas que não estejam sujeitas a controlo de outros interesses, que de um modo geral não têm de ser coincidentes, e cuja índole é estritamente económica. A adopção e contribuição para uma base de software aberto é algo que faz particularmente sentido na aplicação estratégica dos esforços de organismos públicos e algo que lhes pode conferir uma desejável independência.

## 4 O papel de cada modelo

Para finalizar a exposição deste modelo importa analisar brevemente qual a sua interacção com o modelo comercial de software. É um erro comum assumir que o software aberto acarreta implicitamente um qualquer espírito de cruzada contra o modelo comercial. Antes pelo contrário, este pode actuar como catalizador para um desenvolvimento sã e competitivo de software comercial.

O modelo comercial apenas deixa de ser benéfico quando aplicado sobre algo que se designa como *commodities*, termo este que exprime aplicações de software de carácter horizontal com uso em larga escala tal como os sistemas operativos e as aplicações de automação de escritório. Nestes contextos torna-se fácil caminhar gradualmente para situações monopolistas, uma vez que após amortizado o investimento feito no desenvolvimento torna-se fácil viciar o mercado inundando-o com produtos que cada vez mais tendem para custos quase nulos para o produtor. É algo que apenas ocorre com produtos intangíveis tal como o software. É também nestes contextos que os monopólios formados são mais danosos uma vez que facilmente são de abrangência global, e que no caso

em que controlam as camadas de exploração dos sistemas condicionam o modelo de programação de todo o outro software que neles opere.

Contudo é também nas aplicações de uso em larga escala que se verifica um factor crucial para o desenvolvimento de software aberto<sup>13</sup>. Este factor reside na obtenção de uma massa crítica de utilizadores programadores. Foi a catalização desta massa crítica que conduziu ao aparecimento de sistemas operativos em modelo aberto.

Nos nichos de software vertical o modelo comercial já opera um papel insubstituível uma vez que em muitos casos é impossível reunir a massa crítica de programadores e que o esforço de programação precisa consequentemente de ser pago e amortizado pela venda dos produtos de software. Não sendo software de aplicação em larga escala não se colocam os perigos de crescimento monopolista apontados.

## 5 Que futuro?

Em face dos motivos expostos, é nossa convicção que se torna claro o interesse em ter uma plataforma de exploração baseada em software e protocolos abertos. Esta abertura constituiria o garante da possibilidade de intervenção do engenheiro informático na evolução e adaptação do software. Além disso, permitiria a sustentação de um desenvolvimento comercial equilibrado, onde as regras de concorrência não se encontrassem distorcidas.

No espaço europeu vislumbram-se já excelentes oportunidades para a iniciativa privada, quer no desenvolvimento de software especializado quer na prestação de serviços pela adaptação e configuração de soluções integradas.

Quanto ao panorama nacional, podemos-nos interrogar se desta vez vamos estar atentos e ter um papel activo no suporte ao software aberto, ou se vamos esperar uns anos até que uma directiva comunitária nos force a adoptá-lo tardiamente.

---

<sup>13</sup>Um outro factor é a capacidade de comunicação eficiente a nível mundial, que foi introduzida com a generalização do acesso à Internet.