

Chamando cada coisa pelo nome correto: as diferenças entre o Software Livre e o Código Aberto

Guilmour Rossi

Centro de Estudos e Sistemas Avançados do Recife (CESAR)
Recife, PE - Brasil
guilmour.rossi@cesar.org.br

Leonardo Bochnia

Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba, PR - Brasil
leonardomelo@alunos.utfpr.edu.br

Abstract—This article aims to clarify the differences between the definitions and meanings of the terms “free software” and “open source.” Through a literature review and historical events, we will analyze the construction and conception of these terms, highlighting how they emerged, were conceptualized, and how their use was appropriated by specialized literature, academia, and the market. The text also emphasizes the near-revisionist movement to eliminate the idea of free software in today’s software development landscape, underscoring the intention behind the creation of the term “open source” as an attempt to attract the software industry, which at the time resisted the adoption of the free software concept. The text concludes that, despite the conceptual similarities in encompassing software licenses, the terms present profound differences. These differences range from their creation objectives, the communities that support them, and the ideology of freedom that each represents.

Keywords—free software; open source; technopolitics.

Resumo—Este artigo busca esclarecer as diferenças entre as definições e significados dos termos “software livre” e “código aberto”. Por meio de uma revisão bibliográfica e dos eventos históricos, analisaremos a construção e concepção dos termos, destacando como eles surgiram, foram conceitualizados e como seu uso foi apropriado pela literatura especializada, academia e mercado. O texto também destaca o movimento quase revisionista para eliminar a ideia de software livre no cenário atual de desenvolvimento de software, destacando a intenção por trás da criação do termo “open source” como uma tentativa de atrair a indústria de software que resistia à adoção do conceito de software livre à época. O texto conclui que, apesar das similaridades conceituais ao englobar as licenças de software em si, os termos apresentam diferenças profundas. Essas diferenças vão desde o objetivo de criação, passando pela comunidade que os sustenta, até o ideário de liberdade que cada um representa.

Palavras-chave—software-livre; código-aberto; tecno-política.

I. INTRODUÇÃO

Vivemos em uma era onde o reuso de código-fonte se tornou uma prática fundamental e rotineira no ciclo de desenvolvimento de software, abrangendo desde grandes aplicações comerciais até projetos pessoais. Essa prática, que envolve o uso de bibliotecas, *frameworks*, *APIs* e trechos de código menores,

destaca a interconexão entre desenvolvedores e a riqueza da colaboração na construção do conhecimento tecnológico.

Para que esse reutilização de código seja possível, é preciso que os desenvolvedores iniciais desse código tenham autorizado a sua inspeção, a sua cópia e a sua modificação. Essa autorização comumente é formalizada através de uma licença, pela qual o desenvolvedor libera seu código-fonte. Embora uma infinidade de licenças possa ser criada (e um desenvolvedor pode usar qual preferir ou até criar a sua), atualmente existe uma gama bem definida de modelos de licenciamento de software. A grande maioria dessas licenças se alinha à duas definições criadas: ou pelas liberdades essenciais do software livre; e/ou pela definição de código aberto.

No entanto, a forma como essa licença é concebida e aplicada pode refletir diferentes filosofias sobre o propósito e o valor do software. Se pegarmos a concepção de software livre, conforme defendida por Stallman, o propósito vai além da mera técnica; ela representa um compromisso com a liberdade do usuário e a promoção da transparência no desenvolvimento de software. Por outro lado, o movimento de código aberto, que emergiu posteriormente, trouxe à tona uma abordagem mais pragmática, que ofuscou as raízes políticas que fundamentam a ideia de software livre. Esta distinção é crucial para compreendermos algumas dinâmicas atuais do setor.

Pensando nisso, o objetivo desse artigo é ser um material acessível e sucinto para apoiar às discussões relacionadas ao tema. Nele exploraremos de forma breve a evolução e as interseções entre esses dois conceitos, começando pela definição do software livre e do *open source*. Analisaremos como a cooptação das bases do conceito de software livre - que chamaremos de uma tentativa de revisionismo hacker - tem impactado o discurso atual e as políticas que cercam ambos os movimentos. Além disso, discutiremos as implicações sociais e políticas dessas terminologias, com foco na necessidade de uma diferenciação mais clara entre elas, a fim de promover uma compreensão mais profunda de suas respectivas comunidades

e objetivos. Por fim, destacamos que a diferenciação entre software livre e código aberto pode ser sintetizada em três principais perspectivas: i) sua definição prática, ii) sua ideologia e iii) sua comunidade de apoio.

II. O SOFTWARE LIVRE

A. Breve histórico

Se voltarmos ao início da história da informática, encontramos um panorama marcado por cadeias de desenvolvimento de software fechado, onde as empresas protegiam rigorosamente seu código-fonte como ativo estratégico. Esse controle era tão rígido que o software, muitas vezes, era passível de patente, protegendo o produto e garantindo exclusividade de uso. Esse modelo incentivava uma visão de mercado onde cada empresa buscava desenvolver internamente suas soluções, isolando o conhecimento de concorrentes e dificultando o compartilhamento de tecnologias.

Esse panorama começou a mudar em meados da década de 1970, período que Castells destaca como o início de uma revolução da tecnologia da informação [1] – impulsionada por diversos fatores, mas principalmente pelos avanços na infraestrutura e na área das redes de computadores. Unindo-se a isso, um crescente “movimento hacker” fomentava o estudo de programação, a exploração técnica e o compartilhamento de conhecimento, criando um ambiente fértil para uma virada (contra)cultural na área da tecnologia.

Foi na década seguinte, em 1983, que a criação do desenvolvimento do movimento do software livre se dá paralelamente à concepção e ao desenvolvimento do Projeto GNU por Richard Stallman em 1983. [2] O Projeto GNU tinha como objetivo desenvolver um sistema operacional novo e totalmente livre, o GNU (acrônimo recursivo para “GNU’s Not Unix” em inglês, “GNU não é Unix”). [3] Stallman queria que a distribuição do sistema operacional GNU fosse diferente do que acontecia na época com os sistemas operacionais mais usados, como o Unix, sistema operacional da AT&T.

Ao criar as definições de como o GNU poderia ser distribuído, Stallman nos apresenta ao texto Manifesto GNU [3], que seria um documento chave para definir as bases filosóficas do software livre e do copyleft (mais sobre isso na sub-seção C). Esse movimento culminou na criação da Free Software Foundation (FSF) em 1985, fundação que passou a fomentar os objetivos do software livre.

Em 1989 é oficializada a 1ª versão pública da GNU Public License (GPL) [2], licença criada com embasamento legal para que pudesse ser usada na distribuição de quem quisesse distribuir o seu software. Vale destacar que não é ela quem define exclusivamente se o software é livre ou não. Qualquer licença que siga às quatro liberdades, que veremos a seguir, é uma licença de software livre.

TABELA I

AS 4 LIBERDADES DO SOFTWARE LIVRE. ADAPTADO DE [4].

Liberdade	Descrição
Liberdade 0	A liberdade de executar o programa como quiser, para qualquer propósito (liberdade 0).
Liberdade 1	A liberdade de estudar como o programa funciona, e alterá-lo de forma que ele faça sua computação como você deseja (liberdade 1). Acesso ao código-fonte é uma pré-condição para isso.
Liberdade 2	A liberdade de redistribuir cópias e assim você pode ajudar outros (liberdade 2).
Liberdade 3	A liberdade de distribuir cópias de suas versões modificadas para outros (liberdade 3). Ao fazer isso, você pode dar a toda uma comunidade a chance de se beneficiar de suas alterações. Acesso ao código-fonte é uma pré-condição para isso.

B. A ideologia do software livre

O software livre, segundo Stallman [4], é aquele que respeita a liberdade dos usuários, permitindo que eles executem, copiem, distribuam, estudem, modifiquem e melhorem o software. Para que um programa seja considerado livre ele deve garantir quatro liberdades essenciais: a liberdade de executar o programa para qualquer propósito (liberdade 0)¹; a liberdade de estudar como o programa funciona e adaptá-lo, se necessário (liberdade 1), para a qual o acesso ao código-fonte é um pré-requisito; a liberdade de copiar e redistribuir o programa (liberdade 2); e a liberdade de melhorar o programa e liberar essas melhorias para a comunidade (liberdade 3) (mais detalhes na tabela I). [4]

Stallman [4] argumenta que o software deve ser tratado como um bem comum, acessível a todos e passível de ser melhorado por seus próprios usuários. Isso contrasta com a visão de um, por ele chamado, software proprietário - que seriam todos os outros softwares não-livres, que não seguem as quatro liberdades.

Segundo Torres [2], o projeto GNU, além de ser um empreendimento técnico, se desenvolve também como um projeto político que visa a criação de uma sociedade mais colaborativa através do uso de software livre. Como nomeia Torres, a tecnoutopia proposta por Stallman e seus seguidores propõe um mundo onde o software é compartilhado livremente, promovendo inovação para todos. [2]

Os benefícios do software livre são diversos e abrangem várias esferas. Primeiramente, a autonomia e controle oferecidos aos usuários permitem que eles modifiquem e adaptem o software conforme suas necessidades específicas, sem dependência de fornecedores exclusivos [4]. Além disso, a

¹A liberdade 0 foi inserida posteriormente, em 1986. Inicialmente, Stallman considerava essa liberdade implícita, mas decidiu torná-la explícita para reforçar. [4]

segurança e privacidade são aprimoradas. Isso se deve ao fato do código fonte estar livre ao público, possibilitando diversos desenvolvedores a acharem brechas de segurança e *bugs*, por exemplo. [2]

Outro benefício é a promoção da colaboração e inovação. O modelo de desenvolvimento colaborativo do software livre permite que desenvolvedores de todo o mundo contribuam com melhorias e novas funcionalidades, acelerando o progresso tecnológico [4]. Por fim, o acesso ao código-fonte também promove a educação e capacitação técnica, pois permite que estudantes e desenvolvedores aprendam diretamente com os programas que utilizam [2].

C. O Copyleft

Um dos mais importantes desdobramentos das definições de software livre é a criação das bases legais do *copyleft*. O Copyleft, trazido pela licença GPL, rege que ao redistribuir o software (modificado ou não), o mesmo tipo de licença copyleft deve ser aplicado. Assim, cria-se uma noção de reciprocidade e perpetuação. Reciprocidade pois se exige que qualquer modificação ou derivação do software original mantenha os mesmos termos de licenciamento. Perpetuação porque garante que as liberdades de uso, estudo, modificação e distribuição serão levadas aos softwares derivados indefinidamente. No Manifesto GNU, Stallman já destacava:

“GNU não está em domínio público. Todos terão permissão para modificar e redistribuir o GNU, mas nenhum distribuidor poderá restringir sua redistribuição futura. Ou seja, modificações proprietárias não serão permitidas. Quero garantir que todas as versões do GNU permaneçam livres.” [3]

As ideias do copyleft influenciam uma gama de outros movimentos para pensar a criação e a distribuição de bens comuns, como as licenças *Creative Commons*, a ciência aberta, as definições de dados abertos, patentes farmacêuticas, etc.

Em seu trabalho de argumentação de uma cultura livre [5] e das licenças de trabalhos criativos, *Creative Commons*, Lawrence Lessig deixa claro a grande inspiração no trabalho de Stallman e da FSF. Aracele Torres conclui sobre esse legado:

“A apropriação do copyleft por outros setores da sociedade e o desenvolvimento de uma espécie de cultura do copyleft, ajudaram a tornar o software livre um movimento mais amplo e que dá voz não somente aos interessados em tecnologia, mas também à artistas, educadores, juristas, etc. (...)”

“Foi através do software livre e da sua história, que todas essas questões foram descobertas e confrontadas pela primeira vez. (...)” [2]

D. Software livre como oposição à Opacidade informacional

Além de softwares e aplicações voltados para o usuário comum, como sistemas operacionais e aplicativos utilitários, é preciso lembrar que os softwares estão presentes em todos os setores que lidam com informações atualmente. Eles influenciam diretamente as operações e processos internos de empresas e governos, além de estarem integrados em dispositivos móveis e na Internet das Coisas, fornecendo dados em tempo real. Esse uso acelerado das tecnologias digitais tem gerado uma crescente preocupação com a opacidade informacional. Conforme Pasquale [6], “[...] o resultado final de ambos os tipos de segredo e ofuscação é a opacidade, meu termo abrangente para a incompreensibilidade remediável.”

Este termo se refere à falta de transparência sobre como as informações são coletadas, processadas e utilizadas por corporações e governos. Pasquale, em seu livro “*The Black Box Society*” [6], destaca como a opacidade informacional é implementada através de algoritmos de tomada de decisão, que controlam desde fluxos financeiros até decisões que impactam a privacidade e a vida das pessoas.

Pasquale [6] argumenta que a opacidade dos algoritmos cria uma dinâmica de poder desequilibrada, onde as corporações e entidades governamentais detêm o controle tecnológico sobre a população, que está alienada, reclusa a conhecer o software superficialmente, apenas. Esta opacidade não apenas obscurece o funcionamento interno dos sistemas, mas também impede que os usuários e a sociedade em geral questionem as decisões automatizadas que se encontram presentes em seu cotidiano. Isso se torna extremamente grave quando esses sistemas algorítmicos perpetuam desigualdades, violam a privacidade ou induzem a erros e discriminações [6].

Nesse sentido, podemos notar como o software livre consegue combater as práticas características da opacidade informacional na área da computação, por valorizar a transparência, a colaboração e a liberdade de uso, estudo, modificação e redistribuição. É uma ferramenta de combate ideológico aos modelos proprietários que restringem o acesso e gera dependência de tecnologias proprietárias, que resultam em controle excessivo sobre usuários e desenvolvedores.

III. O OPEN SOURCE

A. Breve histórico

A criação do movimento Open Source tem início em 1998, com a fundação, por Eric Raymond e Bruce Perens, da Open Source Initiative (OSI) [7]. Como apontando no próprio site da OSI, sua fundação já vem de um caráter estratégico para a substituição do termo Software Livre.

Para definir o que exatamente seria o código-aberto, a OSI apresenta a Open Source Definition (OSD), um documento

composto de 10 cláusulas² e que uma adaptação de uma licença de software já existente em 1998 - a licença da qual era distribuída o sistema operacional Debian (a Debian Free Software Guidelines). [8]

Argumenta-se que a iniciativa surge como uma adaptação mercadológica do movimento de software livre, adaptando-se às exigências da economia capitalista e destacando, principalmente, os benefícios comerciais das quatro liberdades essenciais, enfatizando os benefícios práticos e comerciais do modelo de desenvolvimento colaborativo, em oposição às motivações éticas do software livre. Esta transformação ampliou a aceitação do software colaborativo em empresas que detinham softwares privados [9], afinal, o *open source* não oferecia apenas um meio de desenvolvimento, e sim um mercado novo que o setor empresarial poderia lucrar, criando, mas também usando e distribuindo esses softwares.³

Juntamente com a fundação da OSI, Raymond apresentou seu texto à indústria e a comunidade de desenvolvedores da época em palestras, posts em seu blog e um artigo (que depois se tornaria um livro), chamado "A Catedral e o Bazar". Esse artigo foi um marco, tendo motivado a decisão da Netscape de liberar o código do *Netscape Communicator* [10] como um software de código aberto, influenciado pelo potencial comercial destacado por Raymond.

Se olharmos então para a questão de definição, seguindo a OSD, - uma adaptação das quatro liberdades do software livre -, podemos dizer que um software livre também é de código-aberto, mas nem todos os softwares de código-aberto são software livre. Essa visualização é facilitada na Figura 1.

B. Um revisionismo hacker

Para os autores, ao criar a definição de código-aberto (fortemente baseada na Debian Free Software Guidelines (DFSG)) e englobar o software livre, mas descaracterizando suas bases filosóficas e focando em critérios técnicos e comerciais, Raymond e a Open Source Initiative (OSI) criam um dispositivo de apropriação. Apropriação essa dos softwares de software livre e do próprio movimento, já bastante forte e importante em 1998. A própria licença do Debian, levava "software livre" no nome, e foi base para a criação da definição de código aberto.

O ideal de Raymond de renomear as coisas não é exatamente um segredo. Na mesma época em que trabalhava nos ensaios

²1. Redistribuição Livre, 2. Código-Fonte, 3. Trabalhos Derivados, 4. Integridade do Código-Fonte do Autor, 5. Sem Discriminação Contra Pessoas ou Grupos, 6. Sem Discriminação Contra Campos de Atividade, 7. Distribuição da Licença, 8. Licença Não Deve Ser Específica a um Produto, 9. Licença Não Deve Restringir Outro Software, 10. Licença Deve Ser Neutra em Relação à Tecnologia. Completo em <https://opensource.org/osd>

³Na data de escrita deste artigo, na página de patrocinadores do website da OSI, costavam nomes como Google Open Source, Amazon, Bloomberg, Cisco, Intel, Github, entre outros... Disponível em <https://opensource.org/sponsors>

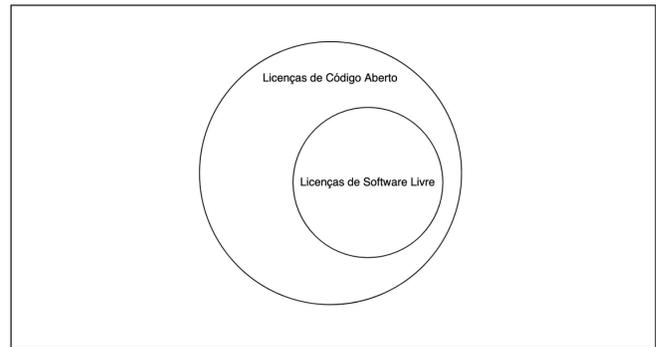


Fig. 1. Pelas definições das licenças, um software livre também é de código-aberto, mas nem todos os softwares de código-aberto são software livre.

que mais tarde formaria o livro "A Catedral e o Bazar" seu trabalho principal de difusão do termo "código aberto", Raymond escreve um texto em seu blog intitulado "Adeus, 'software livre'; [11] olá, 'código-aberto'" onde faz uma pequena defesa descrevendo por qual razão acredita que o termo "código-aberto" deve ser usado e não o de "software livre".

Raymond introduz o texto destacando que a comunidade de desenvolvedores da época tem "um problema com o termo 'software livre' em si, e não o seu conceito".

O primeiro ponto que Raymond destaca é o da ambiguidade do termo, no inglês, ao usar o "free". Em inglês, "free software" pode dar uma noção de software gratuito e não de software livre. Esse problema pode ser passado também nas traduções, já que alguns materiais e artigos no Brasil à época e em anos seguintes, cometeram a gafe técnica de traduzir "free software" para "software gratuito". Raymond destaca a necessidade de uma constante "propaganda" da Fundação do Software Livre (FSF) à época para combater essa confusão. "Pense o free como em liberdade de expressão, e não como em cerveja grátis" era um dos principais slogans da FSF ao tratar o assunto. Stallman defenderia o termo comentando como o quão natural se diz termos como "*free market*" sem em nenhum momento pensar se tratar de um mercado de coisas gratuitas [4].

Um segundo motivo para um "adeus" ao software livre, para Raymond, está em como o termo software livre deixaria os "tipos corporativos nervosos" [sic], e que, apesar de não se incomodar diretamente com o isso, sugere uma "rotulagem" para "converter as pessoas do mundo corporativo ao invés de fazê-las torcer o nariz". Em seu livro a Catedral e o Bazar, Raymond expande essa explicação:

"A maior parte do dano, porém, veio de algo pior — a forte associação do termo 'software livre' com hostilidade aos direitos de propriedade intelectual, comunismo e outras ideias que dificilmente agradariam

a um gerente (...).” (RAYMOND, 2001, tradução e grifo nosso)

Com isso, Raymond traz questões político-econômicas para o debate. Admitindo, em um segundo momento, que ele tratava com maior importância a questão do distanciamento do mundo de negócios do que a da ambiguidade do termo “free”. Essa distância pode ter sido percebida já no ano de 1998, onde Raymond comenta [11] que se reuniu com alguns “fãs de Linux”, incluindo Larry Augustin, e que juntos pensaram em várias alternativas, chegando então até o termo “código-aberto”. Larry Augustin, no entanto, era mais que um fã de Linux - era CEO e investidor na época, e no ano seguinte do encontro se tornaria bilionário pelas ações de uma das suas empresas.⁴

Para além de sugerir um “pragmatismo” de um novo “rótulo”, no texto de 1998 sugere a substituição do termo em todos os lugares possíveis:

Sugerimos que, em todos os lugares onde anteriormente falávamos sobre “software livre”, o termo seja alterado para “código aberto”. Software de código-aberto. O modelo de código-aberto. A cultura de código-aberto. As Diretrizes de Código-aberto do Debian (...) (tradução nossa).

Com esse chamado, Raymond explicita também o desejo de um fomento ao apagamento histórico do software livre. Logo em seguida, destacando que o novo rótulo “atraiu muito apoio (e alguma oposição) na cultura hacker”, Raymond tenta criar a proximidade dos seus novos esforços com uma cultura hacker já pré-estabelecida, e que esse seria o novo e melhor caminho. Para fins de terminologia engole o software livre e para fins do movimento (agora, trazido por ele mesmo, para um contexto político) sugere sua extinção. Colocando, já em seu cerne, o código-aberto e o software livre em campos opostos.

Tal movimento de cooptação se assemelha com ideias recorrentes de como o capitalismo como sistema econômico coopta e mercantiliza pautas como feminismo, diversidade e liberdade. Sempre que a mudança disruptiva é iminente e forte demais, há sua apropriação e reformulação. Na área de tecnologia propriamente, e antes das discussões de surgimento do código-aberto, esse ideário de adaptação já era analisado no próprio Vale do Silício, e foi chamado de “ideologia californiana” pelos autores Richard Barbrook e Andy Cameron [12].

IV. AS POLÍTICAS DO *Open Source* E SOFTWARE LIVRE PERANTE A SOCIEDADE

Langdon Winner argumenta que certos artefatos tecnológicos podem servir como ferramentas de segregação e ordenamento

⁴VA Linux - Easy Come, Easy Go / Stock’s dramatic rise and fall leave determined workers unfazed. Disponível em: <https://www.sfgate.com/business/article/VA-Linux-Easy-Come-Easy-Go-Stock-s-dramatic-2962977.php>

social, incorporando relações de poder específicas [13]. Aplicando esse conceito ao *open source*, podemos ver como essa forma de software, apesar de sua aparente neutralidade, pode ser cooptada por interesses capitalistas que não necessariamente servem ao bem comum, mas sim a objetivos que visam a acumulação de capital e poder em monopólios tecnológicos, uma aparente contradição na filosofia de colaboração comunitária.

A construção de pontes em Long Island, projetadas por Robert Moses, é um exemplo claro de como artefatos tecnológicos são instrumentos de controle político e social. As pontes, segundo Winner [13], foram projetadas intencionalmente para ter uma altura específica, de forma que impedisse que os ônibus urbanos passassem por elas. Essa característica específica das pontes não foi ao acaso, muito pelo contrário. Ela foi deliberadamente escolhida, servindo como uma restrição ao acesso das populações urbanas mais pobres, predominantemente negras e latinas, às praias públicas de Long Island, preservando assim o uso desses espaços para as classes médias e altas, majoritariamente brancas.

Winner destaca [13] que a escolha de tecnologias específicas pode ter implicações duradouras para a sociedade, moldando suas práticas, suas relações de poder e até mesmo suas liberdades fundamentais. No contexto do software, isso significa que a escolha entre software livre e *open source* não se limita a apenas questões técnicas e de programação, mas também questões profundamente políticas e, principalmente, econômicas. O software livre oferece resistência à monopolização do conhecimento e promove que o conhecimento seja livre e acessível a todos. Já o *open source*, apesar de ter práticas de colaboração comunitárias na comunidade de desenvolvedores, cria monopólios tecnológicos. Estes monopólios são responsáveis pela criação de “bolhas de tecnologia”, onde se criam comunidades específicas para cada tecnologia *open source*.

Essas bolhas ditam qual será a tecnologia vigente em determinado período de tempo. Isso acaba culminando em diversas empresas adotando linguagens de programação específicas, *frameworks* para desenvolvimento, sistemas operacionais, entre outros. A partir deste fenômeno o mercado de trabalho é impactado, as empresas criam vagas para que desenvolvedores que tenham aptidão com a tecnologia adotada. Essa demanda cria a necessidade de novos profissionais, com experiência prévia ou não, que vão usar as ferramentas *open source* apenas como meio para o desenvolvimento de um artefato final, que gera lucro para a empresa que o contratou.

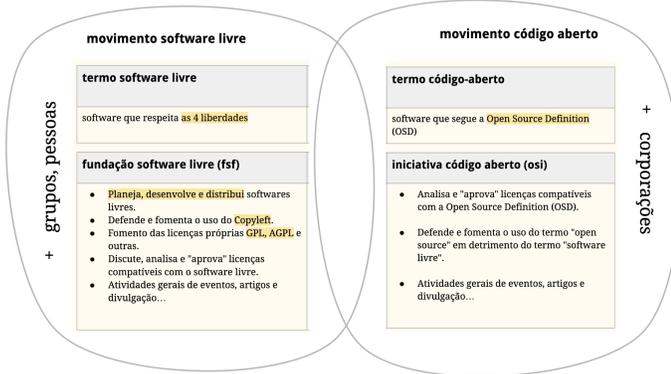


Fig. 2. Pensando a real organização do Software Livre e do Open Source, com suas definições, ideologias e comunidades. Fonte: Do autor.

V. CONCLUSÃO

“Só serei verdadeiramente livre quando todos os seres humanos que me cercam, homens e mulheres, forem igualmente livres (...)” dizia o filósofo anarquista Mikhail Bakunin. Apesar de multifacetada, a ideia liberdade é tão significativa que não pode ser deixada de lado. Quando o Código Aberto (*open source*) surge de uma adaptação do software livre, mas rechaça sua filosofia básica e foca apenas na questão técnica, podemos argumentar que ele não cria um movimento que, como parece, contempla o software livre, mas sim cria um novo movimento contrário ao software livre. (Uma visualização dessas diferenças foi sintetizada na Fig. 2.) O novo termo surge como uma cópia, mas onde as ideias originais de liberdade, colaboração e acesso irrestrito ao código foram ajustadas — e, em certo sentido, diluídas — para tornar o modelo mais “palatável” e integrado ao sistema capitalista.

Essa natureza de cooptação e conflito de sua origem, a proximidade de um “mundo corporativo” em detrimento de uma liberdade técnica (e política), e o trabalho do movimento código-aberto desde o seu princípio para o apagamento do software livre colocam os dois em polos opostos, com diferenças conceituais, filosóficas, e políticas.

Usar um termo ou outro carrega esse peso histórico e de significado, e torna-se ainda mais importante chamar cada coisa pelo seu nome correto, especialmente se você concorda com Bakunin.

Langdon Winner, em suas reflexões sobre como “artefatos têm política”, oferece uma perspectiva para compreender a conclusão desta pesquisa. Winner argumenta que as escolhas tecnológicas incorporam relações de poder específicas, moldando o uso social e fortalecendo estruturas existentes de autoridade e controle. Aplicando esta perspectiva ao contexto da pesquisa, conclui-se que as decisões de design e as políticas

de licenciamento em software não são meramente técnicas, mas têm raízes profundas em políticas capitalistas. Assim, o surgimento do código aberto acaba por definir quem controla os meios de desenvolvimento e, conseqüentemente, quem tem poder na sociedade informacional. Este entendimento ressalta a importância de uma vigilância contínua sobre as práticas de desenvolvimento de software, assegurando que elas promovam uma liberdade que beneficie todos, não apenas uma elite tecnológica ou corporativa.

AGRADECIMENTOS

Agradecemos ao Centro de Estudos e Sistemas Avançados do Recife (CESAR) pelo fomento ao desenvolvimento deste trabalho e sua apresentação. Obrigado ao Grupo de Estudos em Cultura, Software e Streaming Livre (GEC2SL) e ao Coletivo Sharepunks, pela possibilidade de debate e construção desse tema. Agradecemos nominalmente também a Gustavo Alberto Gimenez Lugo, Leander Cordeiro de Oliveira, Gabriel Bérti e Luis Felipe Moro Coelho pelos valiosos apontamentos nos rascunhos iniciais deste trabalho.

REFERÊNCIAS

- [1] M. Castells, *A sociedade em rede*. Paz e terra São Paulo, 2005, vol. 1, no. 6.
- [2] A. L. Torres, “A tecnoutopia do software livre: uma história do projeto técnico e político do GNU,” Mestrado em História Social, Universidade de São Paulo, São Paulo, Jan. 2014. [Online]. Available: <http://www.teses.usp.br/teses/disponiveis/8/8138/tde-31032014-111738/>
- [3] R. Stallman *et al.*, “The gnu manifesto,” 1985.
- [4] R. Stallman and M. Free Software Foundation (Cambridge), *Free software free society: selected essays of Richard M. Stallman*, 2015, oCLC: 927962245.
- [5] L. Lessig, *Cultura livre: como a grande mídia usa a tecnologia e a lei para bloquear a cultura e controlar a criatividade*. São Paulo: Trama, 2005, oCLC: 69938584.
- [6] F. Pasquale, *The black box society: The secret algorithms that control money and information*. Harvard University Press, 2015.
- [7] “History of the OSI.” [Online]. Available: <https://opensource.org/history>
- [8] B. Coleman and M. Hill, “How Free Became Open and Everything Else under the Sun: Introduction,” *M/C Journal*, vol. 7, no. 3, Jul. 2004. [Online]. Available: <https://journal.media-culture.org.au/index.php/mcjournal/article/view/2352>
- [9] M. N. S. Sudibjo, “The Proliferation of Open-Source Licenses: The Cooptation of an Industrial Transition Movement by Multiple Institutions,” thesis, Aug. 2017, accepted: 2020-08-22T20:42:04Z. [Online]. Available: <https://ir.vanderbilt.edu/handle/1803/13784>
- [10] R. d. A. Evangelista, “Traidores do movimento: política, cultura, ideologia e trabalho no software livre,” Doutor em Antropologia Social, Universidade Estadual de Campinas, Campinas, Feb. 2010. [Online]. Available: http://acervus.unicamp.br/index.asp?codigo_sophia=477515
- [11] “Goodbye, “free software”; hello, “open source.”” [Online]. Available: <http://www.catb.org/esr/open-source.html>
- [12] R. Barbrook and A. Cameron, “The californian ideology,” *Science as culture*, vol. 6, no. 1, pp. 44–72, 1996.
- [13] L. Winner, “Artefatos têm política,” *Analytica*, vol. 21, no. 2, pp. 195–218, 2017.