

翻訳

# オープンソース ソフトウェア 開発の教訓

Tim O'Reilly

翻訳：安藤 進

原文: "Lessons From Open-Source Software Development"  
Communications of the ACM, Vol.42, No.4, pp.33-37 (Apr. 1999). 原著者より許可を得て翻訳

最近、オープンソースという言葉が市民権を得てきた。これはソフトウェアの世界で、オープンな標準、ソースコードの共有、共同開発といった伝統の復活を意味する。オープンソースソフトウェアには、LinuxやFreeBSDといったOS、Apacheサーバ、PerlやTcl、Pythonなどのプログラミング言語、さらにDNS (Domain Name System) 用のBind (Berkeley Internet Name Daemon) サーバやSendmailサーバなどがあり、いずれもインターネットの基盤を支えるソフトウェアである。オープンソースが国際的に注目されたのは1998年だった。Netscape社はこの年、同社のWebブラウザ (Mozilla) の次期バージョンをオープンソース製品にすると発表した。IBM社も同年、Apacheサーバを同社WebSphereラインの目玉に取り上げた。

正式にいうと、オープンソースはオープンソースイニシアティブ (Open Source Initiative) という組織の商標である ([www.opensource.org](http://www.opensource.org) を参照)。ただし、この名前から、単にソースコードをオープンにするという意味だけに理解してはならない。ソースコードを無制限かつ無償で再配布し、ライセンスで派生作品 (derivative works) の作成や変更を許可しなければならない。さらにオリジナル作品と同じ条件で派生作品の再配布を許めなければならない。このようなOSD (Open Source Definition) に準拠したライセンスとしては、GPL (GNU Public License)、Berkeley Unixデリバティブ、X Windowシステム用のXコンソーシアムライセンス、Mozillaパブリックライセンスなどがある。

ただし、オープンソースはライセンスの件だけを重視しているわけではない。コンピュータの技術革新には著しいものがあるが、その中で米国の経済や将来の方向性に大きな影響を与えたのが「ハッカーの文化」であることは、あまり知られていない。ハッカー文化そのものだけでなく、この文化が斬新で高品質のソフトウェアを生み出した仕組みを理解しなければならない。だが、規模の大小を問わずどの企業も、ソースコードの無償配布が自分たちの依拠する経済モデルにどのような影響を与えるかが理解できずに苦慮している。

オープンソースをめぐる問題は、Windows NT対LinuxとかMicrosoft社対反Microsoft陣営の戦いといった観点で論じてはならない。まず、オープンソースでの経験がソフトウェア開発全体に適用できることを確認する必要がある。企業における製品の開発や拡張について考えると、むしろ企業がユーザコミュニティに関与を認めるほうがよい製品ができるのではないだろうか。ユーザ側について考えると、各個人が自分のソフトウェアを他者に提供し、その見返りとして他者からフィードバックを得るようにしたほうが、自分だけの世界に閉じこもって開発するより効果的ではないだろうか。新しいアプリケーションの対象がデスクトップからインターネットに移行するにつれ、オープンソースプロジェクトで採用された大規模な共同作業の指針がソフトウェア開発以外の事業にも効果的に適用されるようになるのではないだろうか。

オープンソースについてもっと深く知りたい方には、Eric Raymondの「The Cathedral and the Bazaar」の一読をおすすめしたい（[www.ccil.org/~esr/writings](http://www.ccil.org/~esr/writings)を参照）。Netscape社はこの記事を読んで同社のブラウザをオープンソースにする決断をした。そのほか、Mark Stoneの「Science of the New Renaissance」もおすすめである（[www.edventure.com/release1/1198.html](http://www.edventure.com/release1/1198.html)を参照）。この論文を読むと、西欧科学の伝統の延長線上にオープンソースが位置付けられることが分かる。

## ソフトウェアは製品かサービスか？

マイクロソフト社によるソフトウェア業界の制覇に伴い、ソフトウェアは製品だと考えがちだ。開発されパッケージに梱包されて販売されるものだから、ソフトウェアは製品だろうというわけだ。だが、シュリンクラップに包まれて販売されるPC用のアプリケーションは、実際に使用されているソフトウェアのほんの一部にすぎない。

大規模な業務用アプリケーションは、社内で開発するか、あるいは既存のものを大幅にカスタマイズしたものが大半である。科学技術用アプリケーションはその大半が「一時的なもの」であり、市販ツールで作られたアプリケーションでも、それぞれの顧客向け特注機能を組み込んでいる場合が多い。ネットワークや大規模なコンピュータシステム、Webサイトの管理では、全体がうまく動くようにするために、小さなツールやスクリプト、いわゆる「つなぎ用のアプリケーション」を絶えず開発しなければならない。さらにデスクトップ環境でも、定型作業を自動化するために、パワーユーザがマクロなど一種の「プログラム」を開発している。

エンドユーザ向けにパッケージ化された製品よりオープンソースのほうが優れているかどうかについては、今

まさに市場がテストしてくれている。だがその結果を待たずとも、この種のカスタムソフトウェアよりオープンソースのほうが優れていることはすでに明らかだ。ソースコードへのアクセスと派生品の作成をユーザに認めれば、ユーザ自身が手を加えることで、製品の自然な進化が促されるし、また製品設計の見直しにも役立つ。

オープンソース技術がパッケージ製品として成功するかどうか別にして、これにより新たな市場が開拓される点は経済的な貢献といってもいいだろう。

オープンソースコミュニティから出現し成長してきた市場を端的に示す例がインターネットである。ここは儲けのチャンスが目白押しだ。インターネット（インターネットに接続する多様なネットワークを含む）は、共同開発を大々的に普及させるのための仕組みであり、かつ共同開発が普及した結果でもある。BindやSendmail、Apacheなどのオープンソース本来のプログラム、そのほか元々オープンソースコミュニティで開発された機能（ブラウザやEメール）を商用化したプログラムが、インターネットの中核を支えている。企業に目を移すと、UUNETはISP市場で、YahooはWebサイト市場で活躍している。そのほかメーカや広告会社をはじめとするさまざまな企業がオープンソース革命から利益を得ている。市場効果は数10億ドルと算定されている。

おそらくもっと大切なことは、ネットワークを利用した共同作業に向かわざるを得ないというのが広範な分野での趨勢であり、その趨勢をオープンソースのやり方が反映しているということだろう。ちょうどルネサンス時代に印刷機が知識の普及を可能にしたように、今日インターネットが大規模な共同開発活動を可能にしている。それ以降、情報の普及スピードは格段に高速化した。そして、表現の自由や情報交換の原則がコンピュータとのかかわりの中で新たな重要性を浮かび上がらせた。プログラムはコンピュータによる表現の1つなのだから、これは当然の成り行きである。これまで人類の知識の普及に常に貢献してきたのは一般に公開された講話であり、まさにオープンソースは公開講話と同じ役割を果たしている。

Webには、HTMLファイルという人間の目に見えるテキスト形式と、CGIスクリプトやインラインJavaScript、サーバ側のJava、ASPなど、プログラム化された機能が混在している。そのためWebの普及に伴い、人間とコンピュータ間のコミュニケーションと、人間同士のコミュニケーションとの間の境界線がますます入り交じってきた。その結果、我々人間同士のコミュニケーションが活発になればなるほど、コンピュータ業界も忙しくなるのである。

## 革新と進化

商用製品の開発には、儲かる製品の開発に重点が置かれるという問題がある。真のニーズに対応しているのだが、特殊すぎるか時期的に早すぎるので、投資利益率(ROI)を考えると、大手企業でもベンチャー企業でも開発に踏みきれない製品がたくさんある。

これまでのオープンソースには、特定のユーザに固有の問題を解決する目的で開始されたプロジェクトがたくさんあった。確かに投資利益率は重要であるが、当初、オープンソースの開発者たちは、同じ目的を持つ開発者がネットワークでつながれているコミュニティに自分の作品を無償で提供すれば、ユーザから機能改良のフィードバックという形で配当金がもらえる、と考えていた。

広域ネットワークの普及ともあいまって、オープンソースは非常に活気付いた。従来のユーザコミュニティはメーカ各社とそれぞれのメーカの製品を中心に構成されていた。しかし、FTPやUsenet、Eメールを利用したソフトウェアの配布が可能になったことで、ソフトウェアをインターネットで配布するだけで意図しないプラス効果を、独立した多数の開発者たちにもたらした。インターネットは伝言ゲームのようなもので、次から次へと伝えられていくうちに、まったく意味不明なものになってしまうこともあるが、大半は機能が大幅に改良された。

ここでオープンソースについて、初期のころの経緯を一瞥しておきたい。当時のプロジェクトは小さな問題の解決に取り組んでいたが、後にそのやり方が非常に大切であることが分かった。

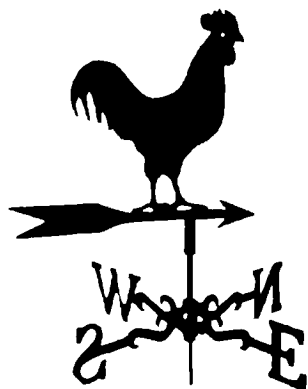
Eric AllmanがSendmailの前身となるものを作成したのは、UCバークレイ校の研究者に対して自分のマシンにログインさせるよりEメールで転送するほうが簡単だ

と考えたからだ。その20年後に、異機種ネットワークやシステム間でのEメール転送がインターネットで必須のアプリケーションになるとは、その当時だれも夢想すらしなかった。Larry WallがPerlを作成したのは、システム管理で発生する頭痛の種を解決したかったからだ。Tim Berners LeeがWorld Wide Webを作成したのは、高エネルギー物理学という特殊な専門領域の研究者たちとの共同研究に役立つものがほしかったからだ。これがインターネットの伝統である。学問の世界で情報の共有を可能にしようとして、これらのインターネットソフトウェアが開発されたのである。Richard Stallmanが、FSF(Free Software Foundation)、GNUプロジェクト、そしてLinuxの誕生にかかわる仕事をしたきっかけは、誤動作するプリンタドライバのソースコードをメーカが見せてくれなかったからだ。ここで紹介した非常に影響力のあるプロジェクトはどれも、最初から市場原理に基づいていたわけではなかった。また、世直しの効果を発揮するまでに何年もかかっている。オープンソースが次々に革新的なものを生み出しているのは、まさにオープンソースが、個人で小さな問題に取り組めるようにしているからである。

Peter Schwartzは、同氏の著書「The Art of the Long View」(Doubleday, 1991年発行)で次のように述べている。「人間や組織は知識を中央に集め、願望や期待の大半を中心部に閉じ込めてしまう。中心部から最も離れた周辺部には、大多数の人々には受け入れられない考え方だけが配置される。その中間に、中央が許可したわけではないが、さりとて禁止したわけでもない灰色の領域が広がる。革新的な考えに中央は敏感だ。中央集権の体制、権力構造、官僚の体質などがすべて革新的な考えを抑制し、中心部から周辺部へ追い出してしまう」。

確かに、既知の問題や市場チャンスに全員で集中して取り組むことで革新的な発明が生まれることもある。このような開発プロジェクトを遂行する仕組みは根付いている。しかし、オープンソースのほうが低コストでしかも成功するチャンスも広がる。

オープンソースは、当たり外れの激しい競馬のような世界ではなく、着実に安定した世界である。LinuxやApache、Perlなどのオープンソースプログラムは、副次的な製品から本格的な製品に成長している。オープンソースの次の目玉は、革新的なものを生み出すための環境を提供することだ。まず、オープンな標準規格とプロトコル。これが整備されれば、新しいソフトウェアを既存のソフトウェアに簡単につないで使えるようになる。次に、よいドキュメンテーション。そして明確に定義された拡張機構。中央にお伺いを立てなくても、だれでも自分で自分なりにやっていかれるようにする。もちろん、正規にライセンスを得ていれば、わざわざ許可を求めないで、他人の作品を利用することができる。



## 共同開発

「十分な人手があればバグは見つかったようなものだ」という最近有名になった文句をはじめ「The Cathedral and the Bazaar」で披露された名言の多くは、大手企業よりインターネットにつながれた開発者コミュニティのほうが大きいという考え方に基づいている。

製品のユーザを開発協力者として巻き込めば、デバッグも速くでき、品質もよくなり、さらに、特化した新機能の追加もしやすくなる。これは、ユーザ層の拡大にもつながる。オープンソースがユーザに自分のやりたいことをできるようにしてくれるので、あまり手間をかけずに新機能がどんどん市場に投入できる。ただし、その新機能が広く受け入れられるかどうかは、市場が決める。この市場は、シリコンバレーのベンチャーキャピタリストの熱気溢れる開発スピードよりずっと速いし実りも多い（大手企業の中央集中型の製品計画などは、比べるまでもない）。

オープンソースでは、製品そのものではなくユーザコミュニティこそがプロジェクトの成否のカギを握るといえる。起業家であつエンジェル投資家であるわたしには、さまざまな開発者からいろいろな話が持ち込まれる。ほとんど市場には受け入れられない限界製品を見せて、これをオープンソースとして人気製品に仕立て上げられないか、といった話もあった。オープンソースを魔法の杖だとでも思っているのだろうか。とんでもないことだ。製品がユーザのニーズを本当に満足させるものであり、その製品からオリジナル以上の機能を引き出してみようという意欲のわくような魅力がなければ、ダメだ。メーカーが定評のある自社製品を既存のオープンソースコミュニティに寄贈することで、自社製品のサポートを手軽に獲得できるともくろむのも、的外れだ。そんなことより、製品のソースをユーザコミュニティに公開するのが、先決だ。

これには巧妙な抜け道がある。ソフトウェアの完全無償配布を確約せずに、オープンソースの一部の方針だけを利用しようというのだ。OSD (Open Source Definition) に準拠しないので、オープンソースの主導者には目の仇にされるが、大手企業にとってはかなりうまみがある。特に潜在的な競合他社にとっては垂涎の的だ。そんなところから、最近大手企業向けのオープンソースの説明会で参加者のひとりが、冗談めかして「門があるのにオープンなコミュニティ」と言われてしまった。

大きなユーザベースを持つ企業は、ソースコードへのアクセスを厳しく制限するが、一方では全ソースコードや完全なドキュメンテーション、ユーザ独自の機能を付加するための仕組みを提供しようとする。ユーザ独自の機能は、メーカー提供のソースコードに取り込まれる。こ



のような場合、企業は、Apacheのようなオープンソースプロジェクトが共同開発を管理するやり方を調べて、密かにその原理を自社の顧客とのやりとりに利用しようとはする。実は、メインフレームが隆盛の時代に、この手法がとられていた。その当時、ソフトウェアはハードウェアに無料で付属されるものだった。たとえば、CICSなどのIBM製品は、元々顧客によって開発されたものをIBMに提供してもらい、それに保守や改良を加えてきたのである。

この点については、Apacheプロジェクトを参照してもらいたい。新しい機能の是非について投票を行い、ソースコードのそれぞれにアクセスできる人物を選定し、オープンソースの目玉である自由な開発を束縛しない方法でやりとりするプロセスを作る必要がある。これがオープンソースコミュニティの大きな課題である。

もちろん、このような試みで、オープンソースに「ネットワーク効果」を発揮させるには、どのぐらいの規模の開発者コミュニティが必要かという点を問う必要がある。個々のメーカー企業に雇われている開発者の数ではあまりにも少なすぎる。メーカー各社は、母屋を無料で開放せずに、インターネットにつながる開発者のコミュニティ全体を活用するために、新しいタイプのライセンスの実験を開始している。たとえば、ソースコードを配布し、非商用目的の変更や再配布は許可するが、商用目的には別のライセンスを要求するというやり方だ。

この新ライセンスとしては、Sun社が最近発表したJava Community Source Licenseや、最近人気のGhostScriptパッケージを開発したPeter DeutschのAladdin Free Public Licenseがある。「コミュニティ開発」に関するPeter Deutschのインタビュー記事を一部紹介しよう ([www.devlinux.org/ghost/interview.html](http://www.devlinux.org/ghost/interview.html))。

いわゆる1960年代のルールでやりたいというなら、AladdinライセンスがまさにGPLと同じ権利と利益を与えてくれる。ソフトウェアというものは本来、使用、コピー、変更が自由にできるものだ。

わたしなりに1960年代の共同使用ルールを要約すると、「すべての人々が貢献し、それぞれが利益を得る」といえるだろう。

GPLとの相違点は、商用目的の配布と非商用目的の配布とを非常に厳格に区別したことである。

少し大げさにいうと、共同ルールに従えばルールの適用範囲内でAladdin製品の利点を享受できるが、共同ルールに従わなければAladdin製品の使用权に相当する費用を払わなければならないということだ。

オープンソースにはさまざまなコミュニティがあるが、その中核になっているのは、あるソフトウェア開発者と複数のユーザとの密接な社会的連合体である。この連合体では、開発者とユーザが双方に利益のあるルールに従って協力することを約束する。このやり方は、メーカ製品に関与している開発者コミュニティにも当てはまる。つまり、オープンソースは、金銭的な関係ではなく共同作業そのものによって、コミュニティ内の連携を維持するものなのである。

## 拡張性の課題

共同作業に対する障害を取り除くライセンスがオープンソース成功のカギになる。しかし、Linus TorvaldsとLarry Wallが指摘しているように、ふたりの成功はほかの開発者たちが参加しやすいようにしたことが大きな要因になっている。たとえば、Linus Torvaldsは、Linux成功の要因として、少なくとも設計方針が適切であったことを指摘している。その結果、当初は思いつかないような形で展開したという。同様にLarry Wallも、ちょうど人間の言語と同じように、ユーザのニーズに対応して自然に機能セットを拡張できるような方法でPerlを開発したと述べている。

実際、成功したオープンソースプロジェクトにはモジュール方式を採用したものが多い。これは、ユーザが既存のコア機能を変更せずに、システムの機能を拡張できる方式である。ユーザコミュニティの拡大に合わせてオープンソースプロジェクトも大きくなるが、コア製品だけはオリジナルの卓越した開発者（あるいはチーム）が握っている。

個人または小人数グループが管理するコア製品部分とユーザコミュニティでの入力部分との間で、どこに境界線を引くかが今後の課題になるだろう。

## 商用製品開発

ユーザコミュニティに基づくオープンソース開発と商用開発との間でどこに境界線を引くべきかを検討しているのは大手企業だけではない。

オープンソースにとっても、一般ユーザが使用できる製品であることを証明するのは、大きな課題である。初期の少数ユーザから大きな市場へ「飛躍する」ための最良の方法を探す試みが多くオープンソース開発者の手で行われている。

ACMの1999年4月号でJohn Ousterhoutは、オープンソースと商用開発はお互いに補足しあうものだと述べている。たとえば言えば、原材料を提供するのがオープンソースで、それに手を加えて洗練させるのが商用開発だというのだ。

Win32プラットフォーム用のPerlを開発したActiveStateのDick Hardtも同じような意見を述べている。最近開催されたPerl会議でDick Hardtは製材される前の丸太と製材後の材木にこれをたとえて説明した。丸太をそのまま提供するのがオープンソースである。丸太小屋を作りたい人にはこれで十分だが、利用者は限られる。組み立て方式のバイナリプログラム部品や付加価値を備えたインタフェース、開発環境、ドキュメンテーション、そのほか商用製品の機能を「製材された材木」として提供すれば、もっと利用者は広がる。

もちろん、そのような商用開発を実現するための最良の方法については、オープンソースコミュニティ内でも議論がある。Linuxについていうと、付加価値機能に対しても、GNU Publicライセンスを厳格に遵守しているRed Hatのような会社もある。Red Hatによると、自社の市場を守るにはパッケージング、ブランド、流通チャネルがあれば十分であり、投資利益率も妥当であるという。

オープンソースの豊富な製品ベースとメーカ独自の付加価値機能を組み合わせる方式を採用している例として、OusterhoutのScriptics社、AllmanのSendmail社、HardtのActiveState Tool社などがある。GPLの熱烈な信奉者やOSDの原理主義者らはこのような動きを批判するだろう（OSDについては[www.opensource.org/osd.html](http://www.opensource.org/osd.html)を参照）。また、オープンソースプロジェクトの成功と発展のためには何が重要かで議論が分かれている。有料にしろ無料にしろ、ソースコードへのアクセスを認めるオープンソースライセンス（オープンアーキテクチャ）を重視すべきだという意見もある。あるいは、ユーザコミュニティを開発プロセスに統合するための各種ツールが重要だという意見もある。いずれにしても、机上での哲学論争より市場の出す実験結果に注目したい。

謝辞 この翻訳では、いくつかの疑問点について執筆者のTim O'Reillyから有益な情報をいただいた。また、日立教育部の田口昭二氏に下訳に目を通してもらい貴重なアドバイスをいただいた。この両氏に感謝を申し上げます。

(平成11年6月16日受付)