

Title	オープンソースソフトウェアと企業の関わり
Sub Title	Firms' involvement in open source software
Author	神戸, 和雄(Kanbe, Kazuo)
Publisher	慶應義塾大学出版会
Publication year	2015
Jtitle	三田商学研究 (Mita business review). Vol.58, No.2 (2015. 6) ,p.121- 128
JaLC DOI	
Abstract	ここ数年、オープンソースソフトウェアの利用に関心が集まっている。実際に導入あるいは導入を検討する企業が増えつつあり、各種調査でもその利用の度合いの高まりを見て取ることができる。ただし、オープンソースソフトウェアを利用している企業といってもハードウェアベンダー、システムベンダー、ソフトウェアベンダーといった各種ベンダーや金融、一般事業会社としてソフトウェア自体を商品とはしていない企業が混在しており、オープンソースの利用の巧拙、得られるメリット、デメリットには多くの類型が存在すると考えられる。本稿ではハードウェアベンダーを除いた、システムベンダーとソフトウェアベンダーを総称してベンダーとし、またソフトウェアの提供を目的とせず利用を主とする企業をユーザ企業とし、この両者の間にオープンソフトとの関わりにどのような違いがあるかを概観する。
Notes	渡部直樹教授退任記念号#論文
Genre	Journal Article
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=AN00234698-20150600-0121

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the Keio Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

オープンソースソフトウェアと企業の関わり

神戸和雄

<要約>

ここ数年、オープンソースソフトウェアの利用に関心が集まっている。実際に導入あるいは導入を検討する企業が増えつつあり、各種調査でもその利用の度合いの高まりを見て取ることができる。

ただし、オープンソースソフトウェアを利用している企業といってもハードウェアベンダー、システムベンダー、ソフトウェアベンダーといった各種ベンダーや金融、一般事業会社としてソフトウェア自体を商品とはしていない企業が混在しており、オープンソースの利用の巧拙、得られるメリット、デメリットには多くの類型が存在すると考えられる。

本稿ではハードウェアベンダーを除いた、システムベンダーとソフトウェアベンダーを総称してベンダーとし、またソフトウェアの提供を目的とせず利用を主とする企業をユーザ企業とし、この両者の間にオープンソフトとの関わりにどのような違いがあるかを概観する。

<キーワード>

オープンソースソフトウェア、ベンダー、ベンダーロックイン、オープンソース化、開発力

I ベンダーサイドの活用と利点

1. オープンソースソフトウェアの利用による収益

オープンソースソフトウェア自体から収益をあげることは難しい。オープンソースソフトウェアに関わるサービス、関連製品で収益をあげることになる。

オープンソースソフトウェアをビジネスに適用した場合の類型はチェスプロウによれば以下の4ステップに集約することができる。

「では、どうすれば企業はオープンソースソフトウェアから利益を得るビジネスモデルを構築できるのだろうか。実際にはオープンソース・ビジネスモデルは数多く存在する。価値

が低いものから高いものに並べてみると以下のような¹⁾。」

- ・ソフトウェアの導入支援、サービス、サポートを販売する
- ・ソフトウェアの複数バージョンを提供し、無償バージョンをエントリーレベル製品として、より高度なバージョンを付加価値製品として提供する
- ・ソフトウェアを顧客の IT 基盤の他の部分と統合する
- ・オープンソースソフトウェアに対してベンダー独自の補完機能を提供する

「ソフトウェアの導入支援、サービス、サポートを販売する」では、リナックスのディストリビューションが好例であろう。リナックスに関しては多くのディストリビューションが存在し、オープンソースソフトウェアを提供するとともに自らはそれらの導入を支援するツールを有償で提供したり、サポートによる収益をあげている。またオープンソースソフトウェアに関する書籍の販売や有料セミナーの開催により収益をあげているベンダーも存在する。

「ソフトウェアの複数バージョン提供」に関してはその数が増大しつつある。近年ではコマースオープンソースと呼ばれ、注目を浴びる製品が増えてきている。例えば、SugarCRM という米国の SugarCRM 社が提供する CRM（顧客関係管理）ソフトウェアが該当する。同社からは無償でソースをダウンロードでき、使用・改変が可能な「SugarCRM Community Edition」というバージョンと、有償でより高機能な「Sugar Product Suite」が提供されている。「Sugar Product Suite」では「商品」、「見積」、「レポート」による分析、「チーム管理」、データの作成・修正に応じて自動処理をする「ワークフロー」、さらに Microsoft Office との連携といったさまざまな機能が付加されている。有償版では Community Edition から得られた知見もとりに、購入したユーザはソースを改変することで、より業務に適切な形での運用が可能となっている。

「ソフトウェアを顧客の IT 基盤の他の部分と統合する」についてはいわゆるソリューションビジネスにその例を見ることができる。顧客企業によるリナックスや Java などのオープンソースソフトウェアと企業側が既に構築している IT 基盤の他の要素との統合を支援することで収益を得ている。また、オープンソースソフトウェアの改変が可能という特長をいかし、顧客のニーズに応じたシステム、ソフトウェアを開発することにより収益基盤を広げる例が多数、存在する。企業側にとっては高価なプロプライエタリなソフトを活用するより、少しの改修で済むのならオープンソースソフトウェアを活用すればコストを抑え、ニーズに適合したシステムを作ることが可能な場合も多いため需要を喚起できる可能性が高い。ソフトウェアベンダーにとっても後々のサポートなどにむすびつくことが期待できる。

Oracle は MySQL, GlassFish, Java, Linux, PHP, Apache, Eclipse, Berkeley DB, Net-

1) チェスブロウ著、栗原潔訳『オープンビジネスモデル 知財競争時代のイノベーション』翔泳社、2007年、57～59頁

Chesbrough, Henry William, Open Business Models: How to Thrive in the New Innovation Landscape, Harvard Business School Press, c2006.

Beans, VirtualBox, Xen といったオープンソースソフトウェアの開発に参加したり、多大な投資を行っている。特に Java は開発の担い手として多大な投資を行っている。無償で提供される Java そのものからは直接的な収益を得られずとも、オープンに開発される Java の普及及びその利用からのフィードバックによって派生するビジネスの伸張が期待できる。

チェスプロウの示した類型の発展形がいくつも存在する。ソフトウェアの開発においてはオープンソースソフトウェアの開発技法を通じて多くのビジネスが成功を収めている。例えば Google 社の Android がそうである。Android 自体はオープンソースソフトウェアであるが、Google 社は Android を通じて多くの収益をあげている。オープンソースソフトウェアの Android は誰もが無料でダウンロード可能となっており、Android の開発そのものからは直接的な利益はあげていないが、「Maps」「Gmail」「Google Play」など Google の公式アプリは含まれていない。特に多くのアプリを入手・購入できるオンラインストア「Google Play」が実装されていない Android は機能的に不完全であり、Google Play の利用があつてこそ、Android を活用することができる。Google 社は Android 自体からは直接収益を得るわけではないが Google Play を通じての収益は近年、急激に拡大しており、オープンソースソフトウェアの関連事業によって収益を高めていることがわかる。これは「オープンソースソフトウェアに対してベンダー独自の補完機能を提供する」の一種と考えることができる。

オープンソースソフトウェアから収益をあげる場合でも、なんらかのバージョンはフリーで提供されており、オープンソースソフトウェアに見られる多くの人々からのチェックを受け、さまざまなアイデアを吸収する仕組みを保持した上で、有償のサービスを提供している点に特徴がある。このことによってオープンソースソフトウェアにより収益をあげようとする企業は外部の多くの人々の知見を製品に反映し、より高品質の製品をより早く世に生み出すことに成功している²⁾のである。

オープンソースソフトウェアでクライアントソフトを無償提供し、製品自体の購入動機を高める方法もある。例えば商用サーバーソフトの販売が事業であり、クライアントソフトを自社製オープンソースソフトウェアで提供することで商品である商用サーバーソフトの売上伸張を図る。あるいはハードウェアを操作・管理するソフトをオープンソースソフトウェアとして提供することで、本来の商品であるハードウェアの販売促進に役立てるといった例がある。

2. ベンダーにとってのオープンソース化することの利点

ソフトウェアベンダーの場合、既に公開しているオープンソースソフトウェアを利用するだけでなく、元来プロプライエタリな自社のソフトをオープン化する動きもある。

本来、プロプライエタリなソフトであったものをオープンソース化すれば収益を失うだけだと考えられるが現実にはより大きな効用を得ることができる。既に2000年代初頭に外食大手のニュートーキョーが自社開発した座席予約システムを公開している。開発を担当したソフトウエ

2) 拙稿「創発的組織と協働要因：オープンな開発とは」『成城大学社会イノベーション研究』第9巻第2号、2014-10から引用・修正)

アベンダーは次のように述べている。「ソースを無償で流通させれば、興味を持った企業や人が自由に修正できる。同時に「こんなふうにはアップデートした」、「この機能はこう変えた方が使いやすい」といった知恵や技術を、どんどん追加してもらおうんです。」「元々の作成者の手を離れて、システムが進化していくと?」「そう。そうしてよくなったシステムは、再び皆の共有物になる。もちろん当社にもです。だから労せずしておいしい果実³⁾にありつける。」

古くはオープンソースソフトウェアを否定し、プロプライエタリな製品を販売することで利益をあげる代表格であった Microsoft もオープンソース化に関して大きく転換しつつある。

Microsoft が開発したアプリケーション開発、実行環境である「.NET」開発フレームワークの大部分を条件付きながらオープンソース化して、オープンソースソフトウェアのウェブプログラミング言語「TypeScript」を公開したほか、長きにわたって同社に利益をもたらしているプロプライエタリな「Windows」を動かすコードを、将来的にオープンソース化する可能性までのめかした⁴⁾。

プロプライエタリ製品を販売する企業の代表格であった Microsoft が収益の基本となる OS でさえオープンソース化さえ念頭に入れているのである。Microsoft は既に OS 以外では、オープン化によってユーザから多くの情報を得て、ソフトウェアを進化させることに成功している。最新 OS として登場が待たれる Windows 10 も Windows 7, Windows 8 ユーザには 1 年間に限ってだが無償でアップグレードを提供することを発表している。なぜここまでオープン化の方向に行くのか、それはオープンソースソフトウェアが持つ、多くの参加者による改善、改良の提案が大きな意味を持つことに加え、前項で述べたように周辺ビジネスでの収益機会を獲得することにつながるからである。Microsoft は Windows 10 を足がかりにしてクラウドサービスの利用拡大を図り、プラットフォームの無償提供により、自社のクラウドサービスへの誘導を図りビジネスチャンスにむすびつける意向のようである。

前項で述べたようにオープンソースソフトウェアはオープンソースソフトウェア自体ではなく、それに付随する事業、製品で収益をあげるものである。Microsoft のように収益の種であったものまでオープンソース化するの新しいものを自社だけでは開発しにくくなっているからである。プロプライエタリの世界から脱し、新しいアイデアを多くの人から吸収することで、より大きな果実を得ることができる道筋を見いだそうとしているのである。

II ユーザ企業の利点と問題点

従来、オープンソースソフトウェアの利用に慎重であった企業も活用を検討するようになっていく。その大きな理由はコスト削減にあるが、オープンソースソフトウェアを導入すれば確実にコスト削減にむすびつくとは言いがたい。場合によってはコスト削減の目的を達成する以前に、

3) 日経 IT プロフェッショナル, 2003/03号, 60~63頁

4) <http://japan.zdnet.com/article/35062952/> (2015/4/14参照)

5) Windows 10 では iOS や Android のソフトウェアを利用可能にすることを想定している。

システムの構築そのものに支障をきたす場合も存在する。オープンソースソフトウェア導入の巧拙、効果を大きく左右するのは企業のソフトウェア技術の高さである。またオープンソースソフトウェアの利用の仕方、局面によっても効果は異なってくるものと考えられる。

1. OSS の利用形態とメリット⁶⁾

企業が情報システムを構築する上で、オープンソースソフトウェアを利用する場合、利用形態にはいくつかの種類がある。

オープンソースソフトウェアの利用にあたってはオープンソースソフトウェアそのものを導入する場合と、オープンソースソフトウェアを独自に改良した製品あるいはサポートを伴った形で導入する場合がある。

オープンソースソフトウェアそのものを自社でダウンロードして活用する場合、十分に長い期間をかけて開発されたオープンソースソフトウェアを利用し、しかも枯れた技術を通常の範囲で使用するのであれば、問題のない水準になっているものも多く、またなにかあっても情報を十分に入手できることから問題が発生しても解決は比較的容易である。しかしながら、外部に委託するのを前提としているような企業あるいは部分的にオープンソースソフトウェアを活用する場合はどうであろうか。その効果は自社の情報システム部門がどれだけソフトウェア能力があるかによって変わってくる。

十分なサポートを期待して委託した場合、委託したソフトウェアベンダーの能力が十分であれば、それなりの効果を得ることができる。サポートや開発体制が整ったベンダーに依頼すればトラブルを回避でき、またよりよい改修も期待できる。ただしそのようなことを期待できる能力のあるベンダーに依頼した場合、プロプライエタリなソフトを利用するのとどれだけの違いがあるだろうか。十分なサポートを得られるような場合は、委託費用も高くついてしまうことが多い。またオープンソースソフトウェアの特長である、改変が自由である利点も失われてしまう可能性が高い。改変をするたびにソフトウェアベンダーに委託する形では時間もコストもかかり、結局、プロプライエタリなソフトを利用するのと変わりがなくなってしまうことにもつながる。

コストダウンを達成するためにもある程度自社で保守・管理・改修ができることが重要となってくるのである。

2. オープンソースソフトウェア自体に潜む問題

オープンソースソフトウェアを使う場合、オープンソースソフトウェア自体に問題が存在する場合がある。オープンソースソフトウェアの利用対象者が一般的でなかったり、利用者も少ない場合、一般的なユーザにとってのオープンソースソフトウェアの大きな利点である、多くのユーザの声の反映ができていない場合もある。

また、オープンソースソフトウェアの場合、その開発体制が脆弱であるために問題が発生する

6) <http://www.tis.jp/special/report03/> (2015/4/14参照) を参考。

こともある。例えば OpenSSL の場合がそうである。

インターネット通信の暗号化のほとんどに使用されていると言っていい OpenSSL に脆弱性が見つかり、日本でもマスコミを中心に対策を早急に講じるよう盛んに広報されたのは記憶に新しい。OpenSSL の脆弱性は OpenSSL がごく少数の開発者（常勤は1人）によって開発されていたことと、容易に理解できることを目指して開発されていたことが相まって、大きな脆弱性が複数生まれてしまったと言われている。有名で利用者が多数いるオープンソースソフトウェアでも開発コミュニティが開発・保守の継続を打ち切ることもある。このようなことにはいくつかの理由があるが、1つはより高度なバージョンの開発に資源を傾注するために既存バージョンの開発を終了する場合や、開発資源（人員、機材、資金など）の不足により継続が不可能、あるいは困難になる場合がある。

このようなオープンソースソフトウェア自体に潜む問題を回避することは相当に困難である。オープンソースソフトウェアを活用する場合、的確なオープンソースソフトウェアを選定する能力が求められるのと同時に、大きな問題が発生した場合の緊急避難に対する知識も持ち合わせる必要がある。

オープンソースソフトウェアの利用にあたっては企業にオープンソースソフトウェアを選定する能力が求められることになるが、対処するためにも特に明確な要件定義ができる能力が必要となってくる。これらの能力がない、あるいは乏しい場合、ソフトウェアベンダーにいわゆる丸投げをすることにもつながってくる。

オープンソースソフトウェアを利用することに関心を持つ企業が関心を持つ大きな理由としてベンダーロックインを避けることが指摘できる。次にこのベンダーロックインとその回避について検討してみよう。

3. ベンダーロックインとその回避

オープンソースソフトウェアを利用する大きな理由の1つとして挙げられるのが、ベンダーロックインの回避である。ベンダーロックインとは、「ある特定のメーカーや販売会社がユーザーを自社製品で囲い込むこと。あるベンダー独自仕様のシステムを採用すると、結果として後継システムや周辺システムも同一ベンダー製を採用せざるを得なくなることもある。特定のベンダーに依存するため製品・サービスなどの調達に競争が働かず、高値でもしぶしぶ購入せざるを得なくなってしまう⁷⁾」現象を指す。

ベンダーロックインに陥った場合、ユーザーは提供される製品やサービスを使い続けることになり、より低い価格での調達や値下げ交渉が困難になる場合が多い。市場によりよい安価な製品、サービスが出現しても乗り換えが容易でないため、新しい技術の利点を享受できないこともある。サービス、システム等を調達する際の選択肢が狭められる。価格が高騰してもユーザーはそれを買わざるを得ないため、コストが増大するケースが多い。また、市場の競争による恩恵を十分に受

7) IT用語事典 e-Works <http://e-words.jp/>

けられない可能性もある。

情報システムにおいてはシステムを開発・構築したベンダーだけが仕様を理解している場合がある。これはひとえに発注側であるユーザ企業が仕様の細かな部分を厳密に定義せずに、完成にいたるまでの仕様変更・追加をベンダーに任せてしまうためにシステムを継続利用するために必ず発生する改修や保守などを開発したベンダーに依頼することを余儀なくされる。日本の企業ではシステム担当者あるいは部署がシステムの構築を丸投げすることもしばしばあり、技術的に深い部分に立ち入ろうとしないため、割高な構築費用をそのまま受け入れざるを得ない状況になっていることが多いと考えられる。

そもそもベンダーロックインになってしまうのはすべての行程をソフトウェアベンダーに丸投げしてしまう企業の体質にある。すべての企業ではないが多くの企業がシステム全体あるいはシステムの一部の開発を丸投げしてしまうことに原因がある。要件定義が綿密になされ、ドキュメントも再現可能なように記載されていれば、利用しているソフトから他のソフトに移行することも、費用と時間はかかるが可能である。しかしながらそのような綿密な要件定義やドキュメントの整備がされていない場合に、ベンダーロックインすると抜け出せなくなってしまう。そもそもベンダーと協力して開発するにせよ、自社のIT担当者が開発に深く関与して設計する必要があるにもかかわらずそのような慣行にはない。最終的には自社のソフトウェアに関する能力が問われることになるのである。

Ⅲ 開発に関与する能力と人材育成

自動車や家電など、高度なソフトウェア技術を製品にいかしている企業が存在する。これらの企業ではオープンソースソフトウェアのうち、OS、特に軽量なOSを組み込んで製品開発にいかすといった形が見られる。また、これらの企業ではソフトウェア技術に長けた人材が存在するため、生産工程の管理等にオープンソースソフトウェアを活用している。新しい製品を自社内の技術だけで開発することには限界が存在する。オープンソースソフトウェアを取り込むことで外部から多様な最新のソフトウェア技術を活用することにつなげている企業も存在するが、このような企業にあってもソフトウェア技術を製造以外の他の部門にいかし切れない場合もある。

ベンダーロックインを避けるためにも、オープンソースソフトウェアを利用してより有用なシステムを構築するためにも自社の情報システム担当者が開発に関与することが重要と考えられる。自社でオープンソースソフトウェアを使いこなすことができれば、なんらかのトラブルがあった場合や急ぎの改良が必要な場合、独自に対応することができる。プロプライエタリ・ソフトウェアは“ブラックボックス”と揶揄され、ソフトウェアベンダーが対応してくれる体制にあったとしても十分に人員を配置してもらうことができず、問い合わせから解決までに長時間かかるケースもある。

8) 「OSSを活用したソフトウェア開発の動向」

https://www.toshiba.co.jp/tech/review/2012/08/67_08pdf/a02.pdf

より有用なシステム、自社独自の仕様を織り込むためには開発に深く関わらないといけない。そのためには再三、強調したように自社のソフトウェアに関する能力、特にオープンソースソフトウェアを活用できる人材の存在が重要となってくるのである。そのような人材がいればオープンソースソフトウェアの大きなメリットである迅速性もいきてくる。

オープンソースソフトウェアに対処できる人材の育成には非常に困難が伴う。そもそも通常のプロプライエタリなソフトに対処する人材を育成することに多くの時間とコストがかかるためにシステム開発に関する工程を丸投げするような現象が生じるほどであるから、オープンソースソフトウェアを使いこなせる人材を育成するにはかなりの覚悟が必要であると考えられる。

プロプライエタリなソフトを利用する場合に比べ、オープンソースソフトウェアを利用するにはそのソフトウェアに関する多くの周辺知識を必要とする。プロプライエタリなソフトであれば、それ相応のマニュアルが整備され、少なくとも明確な使用方法が提供される。オープンソースソフトウェアの場合はオープンソースソフトウェアを利用する手順自体を学習する必要がある場合が多い。またなんらかのトラブル対処、改修をする場合、OSそのもの、あるいは他のオープンソースソフトウェアに関する知識も必要となる場合が多いため、習得するための体系的な方法があるとは言いがたい。体系的な学習方法を持たないオープンソースソフトウェアの習得のためには企業側は個人にある程度、自由に学習する時間を与え、自ら能力を育成する機会を設けることが必要となると考えられる。

従来、企業はプロプライエタリなソフトに関する人材開発の期間、コストを節約するために外部への発注という手段をとってきた。さらに期間とコストを要するオープンソースソフトウェアに対応できる人材の育成には腰がひけるであろう。しかし、プロプライエタリなソフトの利用にとどまるにせよ、オープンソースソフトウェアの活用を考えるにせよ、自社内部のソフトウェア対処能力を高めることは今後、重要度をより増すと考えられる。自社での対応ができないのであれば、結局、ソフトウェアベンダーの能力を超える活動はできなくなるからである。

本稿では限られた誌面の中でオープンソースソフトウェアの活用を概観した。今後、できるだけ多様なオープンソースソフトウェアの活用の実態を調査し、より明確な処方箋を描くことを考えている。