

Articles

論文

ソフトウェアに起きる究極の価格破壊

— 業務系オープンソース・ソフトウェア普及の可能性 —

主任研究員

前川 徹



目次

- | | |
|--------------------------------|------------------------------|
| 1. はじめに | 4.2. ニュートーキョーの狙いは実現されたか |
| 1.1. オープンソフトウェアとは何か | 4.3. 利用企業とベンダーのメリット・デメリット |
| 1.2. OSS とフリー・ソフトウェア | 5. 業務系 OSS が増加・普及する可能性 |
| 1.3. OSS とバザール型開発 | 5.1. 業務系 OSS とインフラ系 OSS との違い |
| 1.4. インフラ系 OSS と業務系 OSS | 5.2. ソースコードのバージョン管理は必要か |
| 2. セルベッサの概要 | 5.3. 業務系 OSS の普及経路 |
| 2.1. セルベッサとは | 5.4. 業務系 OSS が普及する条件 |
| 2.2. セルベッサが OSS 化されるまでの経緯 | 5.5. 業務系ソフトを非公開にする意味はない |
| 3. セルベッサの現状 | 6. 結論 |
| 3.1. セルベッサの普及 | 6.1. 業務系 OSS の普及と IT ベンダーの対応 |
| 3.2. セルベッサの様々なバージョン | 6.2. ソフトウェアに起きる究極の価格破壊 |
| 3.3. ライセンス方式とソースコード管理の現状 | |
| 3.4. バグ情報などの共有 | |
| 4. 業務系ソフトの OSS 化がもたらす便益 | |
| 4.1. セルベッサを OSS 化したニュートーキョーの狙い | |

要旨

Linux の普及に伴ってオープンソース・ソフトウェア (OSS) に対する関心が高まっている。OSS の利用は、Linux のような基本ソフトだけでなく、DBMS などのミドルウェアやデスクトップ・ソフトから業務系ソフトにまで広がりつつある。例えば、ニュートーキョーが自社向けに開発し、1999年11月に OSS 化された食材等の受発注システム「セルベッサ」の利用企業は徐々に増加している。

セルベッサを事例に、業務系ソフトの OSS 化のメリット、デメリットを分析すると、業務系ソフトの利用企業は OSS 化によって、当該ソフトの質の向上、機能強化版の無償入手、サポートベンダーの増加などのメリットを享受できる可能性が大きいことがわかった。今後、利用企業の経営者や CIO が、業務系ソフトの OSS 化によるメリットを理解するようになると業務系 OSS の種類は急速に増えていく可能性がある。

多様な業務用 OSS が普及すれば、同種のソフトウェアを複数の企業がそれぞれ開発するという重複投資がなくなり、日本経済全体として無駄な情報化投資が排除できる。しかし、その一方で、ソフトウェアの新規開発需要は減少する可能性があるため、ソフトベンダーはソフトウェアの新規開発ではなく、情報システムの運用管理、サポート、カスタマイズなどを中心としたビジネスモデルに転換していく必要がある。また、既にハードウェアやネットワークの世界では価格破壊が進行しているが、OSS の普及はソフトウェアの価格がゼロになることを意味し、ソフトウェアの世界に究極的な価格破壊をもたらす。

1. はじめに

1.1. オープンソフトウェアとは何か

オープンソース・ソフトウェア (OSS) とは、一般的に、インターネットなどを通じてソースコードが公開されており、誰でも自由に使用・改変・再配布ができるソフトウェアのことである。短く「オープンソース」と表現されることも多い。

厳密に言えば、OSS の普及を推進している Open Source Initiative (OSI) が定めた「The Open Source Definition (OSD)」と呼ばれるオープンソース・ソフトウェアの定義を満たすソフトウェアがオープンソース・ソフトウェアである。この OSD によれば、オープンソース・ソフトウェアのライセンスは次の条件を満たさなければならない¹⁾。

(1) 再頒布の自由

(販売や無償譲渡を制限しないこと、またロイヤルティなどを要求しないこと)

(2) ソースコードの公開

(ソースコードを頒布すること。もし、ソースコードが頒布されない場合には妥当な料金でソースコードが入手できるようにするか、インターネットで無料ダウンロードできるようにすること)

(3) ソフトウェアの改変

(プログラムを自由に改変できること。その改変したソフトウェアや派生したソフトウェアは、もとのソフトウェアと同じライセンスで頒布することを許可すること)

(4) 開発者のソースコードの同一性保持

(パッチファイルの配布を許可する場合は、改変したソースコードの頒布を禁止することができる。改変したソフトウェアには異なる名

称またはバージョンをつけるように要求することができる)

(5) 個人やグループに対する差別の禁止

(ライセンスは特定の個人やグループに対して差別してはいけない)

(6) 使用分野に関する差別の禁止

(ライセンスは特定の分野でのソフトウェアの使用を制限してはいけない)

(7) ライセンスの継承

(プログラムに付与された権利は、このソフトウェアの頒布を受けた者すべてに対して、追加的なライセンスなしに与えられること)

(8) 製品依存の禁止

(プログラムに付与された権利は、その頒布形式に依存しない)

(9) 他のソフトウェアの制限禁止

(当該ソフトウェアと同時に頒布されるソフトウェアに制限を加えてはならない)

(10) 技術的中立性

(特定の技術やインタフェースの規格に依存する規定があってはならない)

一般的に「オープンソース」だと言われているソフトウェアすべてが、このこの条件を満たしているわけではない。単にソースコードが公開されているソフトウェアがオープンソースと呼ばれることもある。しかし、この条件を満たしているライセンスのリストが OSD のウェブサイトに掲載されており、それらのライセンスが適用されているソフトウェアが厳密な意味で OSS だということになる²⁾。

1.2. OSS とフリー・ソフトウェア

ソフトウェアのソースコードを公開したり自由に流通させるという行為は、コンピュータの黎明

1) 正確な定義については次の URL を参照のこと。 <http://www.opensource.org/docs/definition.php>

2) OSD を満たしているライセンスのリスト : <http://www.opensource.org/licenses/>

期においては、ごく普通のことであった。しかし、コンピュータの商業利用が拡大し、ソフトウェアがハードウェアとアンバンドリングされ、別の製品として扱われるようになっていく過程で、ソースコードの公開はハッカー³⁾間における限定された行為だとみなされるようになっていった。

ソースコードの公開と自由な利用が社会的に意味のあるものであることを主張し、社会運動を展開したのが、リチャード・ストールマン (Richard Stallman) である⁴⁾。OSS の思想的原点は、1985年3月にストールマンが発表した「GNU 宣言⁵⁾」と、1986年に同氏が設立した Free Software Foundation, Inc にあると考えてよい。

GNU 宣言は、UNIX と互換性のあるシステムをフリー・ソフトウェアだけで構築しようという GNU プロジェクトへの参加とサポートを求めるためにプロジェクト初期にストールマンが書いたものであり、Free Software Foundation は GNU プロジェクトを推進するために設立された非営利団体である。

Free には「無料」という意味もあるが、ストールマンの言う Free は「自由」を意味する。ストールマンの目標は、だれもが空気のように自由にソフトウェアを入手でき、自由に改変し、自由に配布できる世界をつくることにあった。一般的に市販のソフトウェアはソースコードを公開しておらず、その改変や再配布はライセンスによって厳しく制限されている。しかし、ストールマンは、プログラマにとってソフトウェアは自由にコピーでき、自由に改変できるものであるべきであり、それが社会的利益にもかなうと考えた。Free には「～を自由にする」「開放する」という意味があるが、

ストールマンはソフトウェアを企業の支配から開放することを目指しているのである。

しかし、ソフトウェアの著作権を放棄するだけでは、ストールマンが考えるフリーソフトウェアは実現しない。なぜなら、著作権が放棄されたソフトウェアを改変すれば、改変したソフトウェアには別の著作権が発生するため、改変したソフトウェアを有償でライセンスする企業や人が現れる可能性がある。そこで、ストールマンは著作権という権利を使ってソフトウェアの自由を保証する方法を考えた。それが GNU GPL (General Public License) である。GNU GPL は Free Software Foundation が提唱しているフリーソフトウェア用のライセンスの一つであり、主に GNU プロジェクトで開発されたソフトウェアや、その派生物などに適用されている。ソースコードの公開のほか、ソースコードを含めた再配布や改変の自由、再配布や改変の自由を妨げる行為の禁止などが条件となっている。

OSS とフリー・ソフトウェアは極めて近い概念であり、混同して使われることが多いが、厳密に言えば、後者は「自由であること」、つまり「フリー・ソフトウェアの派生物もまたフリー・ソフトウェアである」ことを求める点で、OSS より制約が厳しい。このストールマンの思想は著作権 (copyright) をもじって「コピーレフト」と呼ばれる。

1.3. OSS とバザール型開発

「フリー・ソフトウェア」に代わるように「オープンソース・ソフトウェア」という言葉が世界的に広まる契機をつくったのは、エリック・レイ

3) マスメディアでコンピュータへの不正侵入者などを「ハッカー」と表現しているが、本来「ハッカー」はコンピュータに精通した優秀な技術者を指す用語である。

4) ストールマンは、UNIX 環境の標準的なエディタである Emacs の作者としても有名である。

5) 1984年に Stallman によって書かれたといわれている GNU 宣言 (GNU Manifesto) は、1985年3月に Dr. Dobb's Journal of Software Tools 10 (3) に発表された。

モンド(Eric S. Raymond)⁶⁾が書いた“The Cathedral and the Bazaar”である。ただし、この論文が公開された1997年秋の段階では「フリー・ソフトウェア」と記述されていたが、1998年2月に「オープンソース」と書き換えられたのだと言われている⁷⁾。

この論文は、Linuxの開発形態を分析したもので、決められたスケジュールと役割分担に従って限定された(クローズドな)メンバーからなる階層的なチームによって開発されるカテドラル型開発と、少数のコーディネーターと不特定のボランティアが途中経過を公開しながら開発を進めるバザール型開発を対比している。

この論文を契機としてオープンソース・ソフトウェアを推進する運動が始まったこともあり、オープンソース・ソフトウェアはバザール方式で開発されるものであるというイメージが広まった。確かに、Linuxを始め、数多くのオープンソース・ソフトウェアがバザール型で開発されているが、実際には企業が開発したOSSも存在しており、開発形態がバザール型であることはOSSの条件ではない。ただし、一度OSSとなったソフトウェアは、ネット上にそれを改良するコミュニティが生まれ、バザール型の開発がはじまるケースもある。

1.4. インフラ系OSSと業務系OSS

現在、もっとも有名なOSSはLinuxだろう。しかし、Webサーバー・ソフトウェアのApacheやDNSサーバー・ソフトウェアのBIND、メールサーバー・ソフトウェアのSendmailなどのインターネット分野のOSSの方が普及率が高い⁸⁾。

また、最近ではファイル共有サーバー・ソフトウ

ェアのSamba、データベース・マネジメント・システム(DBMS)のMySQLやPostgreSQL、スクリプト言語のPerlやPHPを利用する企業もかなり多い(図表1参照)。

現状を見ると、ほとんどのOSSはOS、DBMS、デスクトップ・ソフトウェアなど基盤的なソフトウェアであり、業務用ソフトウェアでOSSになっているものは少ない。しかし、業務系OSSがまったく存在しないわけでもない。例えば、外食チェーン向けの食材受発注システムの「セルベッサ」や座席予約管理システムの「ガラガラドア」、美容院向けWebPOSシステムの「フランシーヌ」、日医総研が開発した病院や診療所の医療報酬明細書(レセプト)処理ソフトである「日医標準レセプトソフト」などである。

現時点では、業務系OSSの種類はまだまだ少なく利用企業もさほど多くないが、今後、業務系OSSの利用が拡大していく可能性は十分あるのではないだろうか。

そこで、この研究レポートでは、外食チェーン向けの食材受発注システムのセルベッサをケースとして取り上げ、業務系ソフトの分野においてもOSS化が進むのかどうかを検討することにする。

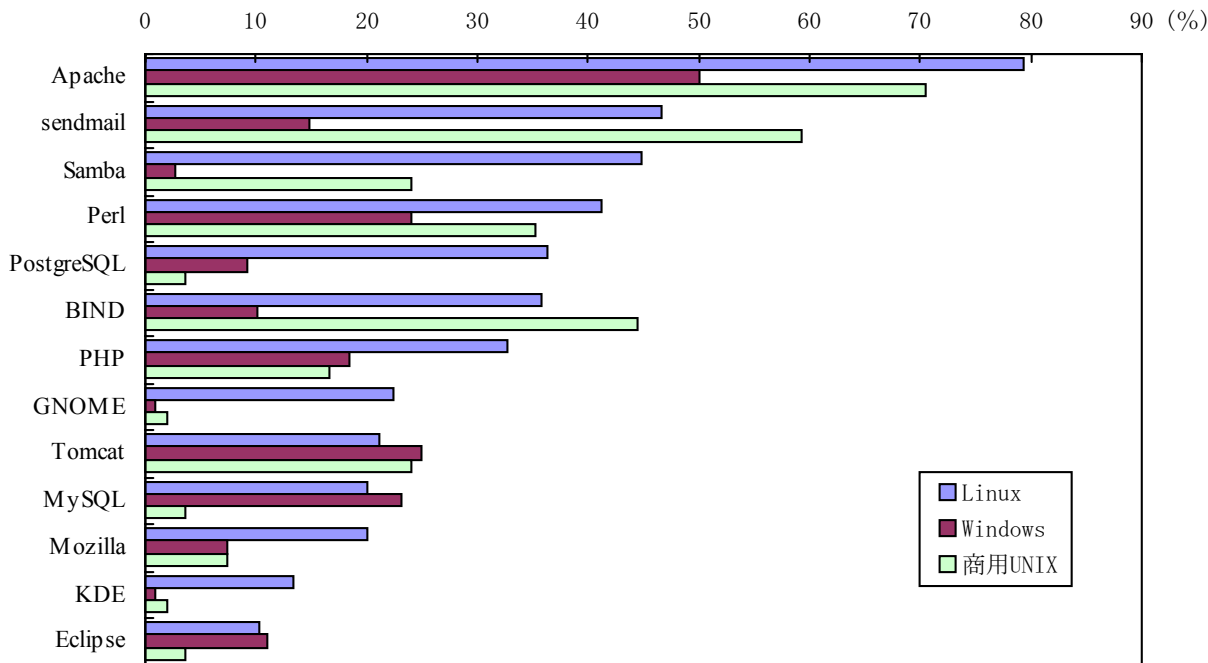
セルベッサを選んだ理由は、営利企業(ニュートキーヨー)が自社業務のために開発したソフトウェアをOSS化した希な例であるからである。もし、多くの企業がニュートキーヨーのように自社向けに開発したソフトウェアをOSS化するようになれば、業務系OSSの種類は一挙に増え、業務系OSSの利用もまた急拡大することになる。

6) レイモンド自身もオープンソース・ソフトウェアの開発に参加しているが、“The Cathedral and the Bazaar”の主題であるLinuxの開発には参加していない。レイモンドはOpen Source Initiativeの創設者の一人で、初代の会長を務め、現在は名誉会長となっている。

7) 佐々木、北山(2000) pp.64-66

8) 例えば、WebサーバーソフトにおけるApacheの普及率は7割以上だと言われている。

図表1 OS別利用オープンソースソフトウェアの種類



(出所) インプレス/矢野経済研究所 2005『Linux オープンソース白書2006』インプレス

2. セルベッサの概要

2.1. セルベッサとは

セルベッサは、株式会社ニュートーキョーが1998年末に株式会社テンアートニに委託して開発した外食チェーン向けの食材受発注システムである⁹⁾。食材や酒類を発注する店舗と食材を受注する企業(食材などの卸やメーカー)を結ぶEDI(電子データ交換)システムだと考えてよい。ちなみにセルベッサ(Cerveza)とは、スペイン語でビールを意味する。

セルベッサには、食材の受発注に必要な機能がほとんどすべて揃っている。例えば、重さや量が決まっている商品はもちろん、魚のように1匹の重量がバラバラの不定貫商品の発注や納品日を指定した予定発注、メニューによる発注(登録され

たレシピに応じて自動的に食材に展開して発注)、納品された食材の検品入力、仕入状況の照会、棚卸に対応しているほか、欠品時の代替品指定機能、店舗別の納品日や休日の指定機能、仕入先が休日の場合の次納品表示機能も備えている。また、仕入先では、受注確認、出荷指示書の作成、送り状の印刷などが可能である。

セルベッサはJavaで開発されたLinuxサーバー上で動くアプリケーション・ソフト(Javaサーブレット)であり、DBMSはOracle¹⁰⁾、WebサーバーはApache、アプリケーション・サーバーはTomcatを利用している。

サーバーと各店舗、仕入先を結ぶネットワークはインターネットあるいはイントラネットを使用し、各店舗と仕入れ先に必要なものは、インターネット接続可能な普通のパソコンとWebブラウ

9) 実際にセルベッサの開発を発注したのは、ニュートーキョーの物流専門子会社である株式会社エヌ・ティー・トレーディング・コーポレーションである。

10) 後述するように、2005年5月にはOSSのDBMSであるPostgreSQLに対応したバージョンが公開されている。

だけでなく、パソコンやOSの種類は問わない。特別なソフトウェアのインストールも必要としない。また、自社でサーバーを持たずに、ASP（アプリケーション・サービス・プロバイダ）を利用するという方法もある。ASPを利用すれば、より短時間、低コストでセルベッサを導入できる。

2.2. セルベッサがOSS化されるまでの経緯

ニュートーキョーがセルベッサの開発をテナアートニに委託することになったきっかけは、1998年2月にニュートーキョーの財務部情報システム室長の湯澤一比古氏が、OSSの利用を推奨するテナアートニの取締役会長（当時）の角田好志氏の講演を聞き、名刺を交換したことにあったと言われている¹¹⁾。湯澤氏は、この講演を契機にOSSとJavaに興味を持ち、当時はメインフレームで動いていたファックスを利用した食材の受発注システムをJavaサーバレットを使って開発してみようと決断したのである¹²⁾。

まず1998年10月にプロトタイプが開発された。これは、Javaサーバレットで開発したWebアプリケーションで食材の受発注業務が処理可能であることを確認するためである。このプロトタイプの動作確認が終わった1998年末に、セルベッサの開発が正式にスタートしている。開発費は約2,000万円であった。1999年に入って完成したセルベッサは、数ヶ月のテスト運用を経て同年7月から本稼動に入った。

セルベッサがOSSとして公開されることは、1999年11月の日経BP社主催のセミナーで発表された。湯澤氏は当初からセルベッサをOSS化するつもりだったと語っているが¹³⁾、当時、テナア

ートニで開発を担当した山崎靖之氏は、セルベッサの開発を始めたときにはOSSになるとは思っていなかったと述べている¹⁴⁾。

3. セルベッサの現状

3.1. セルベッサの普及

セルベッサは食材等の受発注システムであるので、食材等を発注する企業（ニュートーキョー）だけでなく、その食材を受注する企業（食材等の卸やメーカー）も利用することになる。ニュートーキョーは当時、食材の仕入れを食材卸の株式会社三友小網（現在の三井食品株式会社）に集約しようとしていたため、同社は必然的にセルベッサの利用企業となった。そこで湯澤氏は、三友小網と取引のある外食企業にセルベッサの利用を持ちかけた。ただし、ニュートーキョー程度の規模がないと自社でサーバーを持つのは負担が大きいため、三友小網の親会社である三井物産株式会社がセルベッサのASPサービスを提供することになった。三友小網もこのASPサービスを利用して食材等の発注を行うという形でセルベッサの活用を始めた。

外食企業として最初の二次利用企業は、東京、大阪などに10店舗のステーキハウスを展開する株式会社アウトバックステーキハウスジャパンである。2000年5月から三井物産のASPサービスの利用企業となっている。同じように三友小網経由で、2000年9月にはカブリチョーザ、トニーローマ、ハードロックカフェなど約200店舗を運営している外食チェーンの株式会社WDIが、2000年10月には定食屋チェーンの株式会社大戸屋が三井物産

11) 竹田、米山（2002）p.5

12) 湯澤（2004）pp.18-22

13) 2005年1月31日の湯澤一比古氏へのインタビューによる。

14) 2005年3月17日の山崎靖之氏（テナアートニ Web ソリューション部長）へのインタビューによる。

の ASP サービスのユーザーとなった¹⁵⁾。

2001年4月には、ダイニングバーや居酒屋など約260店舗を展開する株式会社ダイナックが自社サーバーでセルベッサの利用を開始した。当時、ダイナックは食材等の受発注システムの構築にあたり、パッケージ利用ではなく自社開発をする方向で検討を進めていた。そんな時に、湯澤氏の講演でセルベッサの存在を知り、利用可能性を検討した結果、ソースプログラムを修正すれば利用可能だという結論に達し、テンアートニにセルベッサのカスタマイズを委託したのである。カスタマイズ作業は2000年10月に開始されているので、本稼動までに半年程度要していることになる。ダイナックの情報システム部長の大橋健治氏は、プログラムのほとんどを書き換えることになったが、それでもゼロから開発した場合に比べて費用は半分程度で済んだと語っている¹⁶⁾。

2002年12月には、名古屋を中心に中部地方で中国料理店を展開している株式会社浜木綿がセルベッサの利用を開始している。浜木綿は、インターネットで公開されているセルベッサ (RPM 版¹⁷⁾) をダウンロードしてカスタマイズを行っている。カスタマイズを担当したベンダーは株式会社ソフトリンクである。

更に2003年12月には、株式会社 R&D 外食ネットがセルベッサの ASP サービスを開始している。R&D 外食ネットは、交通広告大手の株式会社エヌ

ケービー (NKB) やニュートーキョーなどが出資して設立した企業である。当時 NKB は、飲食店検索サイト「ぐるなび」を運営している株式会社ぐるなびの親会社であり¹⁸⁾、外食企業向けの B2B 事業の可能性を探る目的があったと考えられる。この R&D 外食ネットの ASP サービスを利用している企業は、企業名が公開されているものだけを挙げて、しょうすけグループの出井商事株式会社、総合食品卸の株式会社クサマ、中国家庭料理チェーンの株式会社希須林、食材卸の永和物産株式会社、業務用酒類卸の株式会社河内屋があり、この他に焼肉チェーンを展開する外食企業、中華食材を扱っている食材卸など数社が ASP サービスを利用している¹⁹⁾。

なお、セルベッサは食材等の受発注システムであるので、受注側の企業も必然的にセルベッサを利用することになる。例えば、ニュートーキョーのセルベッサに関する Web サイト²⁰⁾ には「セルベッサの受注実績がある食品卸の一覧」が掲載されており、ここには約90社の企業名がリストアップされている。また、ダイナックの取引先でセルベッサで発注可能な仕入先は約450社ある²¹⁾。こうした仕入先企業まで含めて考えると、セルベッサの利用企業は数百社に達していることになる。

3.2. セルベッサの様々なバージョン

現在利用されているセルベッサは、すべて同じ

15) 現在、アウトバックステーキハウスジャパンと大戸屋はセルベッサを利用していない。

16) 2005年5月19日の大橋健治氏 (株式会社ダイナック 情報システム部長) へのインタビューによる。

17) RPM とは RedHat Package Manager の略で、ソフトウェアのインストールや削除を容易にするためのパッケージ管理ソフトである。名前のおおりに、RedHat 社が開発したものであり、OSS のディストリビューションによく用いられている。

18) 株式会社ぐるなびは2005年4月に大阪証券取引所「ヘラクレス」に上場された。上場時の NKB の持ち株比率は35%程度である。

19) 2005年5月18日の戸田和人氏 (株式会社 R&D 外食ネット 執行役員兼 COO) へのインタビューによる。

20) <http://www.newtokyo.co.jp/serveza/>

21) 2005年5月19日の大橋健治氏 (株式会社ダイナック 情報システム部長) へのインタビューによる。

ダイナック版を基にして、DBMS を Oracle 9i から OSS の DBMS である PostgreSQL に変更した「セルベッサ (PostgreSQL 版)」を開発している。現在、ダイナックが利用しているのは、この PostgreSQL 版である。なお、PostgreSQL 版のソースコードは2005年5月からインターネット上で公開され、ダウンロード可能になっている。

3.3. ライセンス方式とソースコード管理の現状

セルベッサのライセンスは、GNU GPL²²⁾ である。したがって、ソースコードを自由に改変したり、他のソフトウェアとリンクすることが許されているが、改変したソフトウェアを再配布する場合には、改変したソースコードやリンクしたソフトウェアのソースコードを GPL に従って公開しなければならない。しかし現実には、ソースコードが公開されているバージョンは、RPM 版と PostgreSQL 版のみである。

オリジナルなセルベッサの著作権を保有しているニュートキーヨーは、GPL を遵守しない利用企業に利用中止を求める権利がある。しかし、ニュートキーヨーは、GPL を遵守していない企業があるかもしれないという現状を容認している。理由はおそらく2つある。

第1の理由は、著作権法上の「使用」と「利用」の違いにある。日本の著作権法上、そのプログラムをある特定のコンピュータで使えるようにするために、あるいはより効率的に使用するためにプログラムに非創作的な変更を加えることは、私的利用（私的使用）の範疇だとみなされ、ライセンスに違反しない。OSS のライセンス問題に詳しい

尾崎孝良弁護士は、著作物の「使用」と「利用」は異なり、書籍を読むことが書籍の著作権を侵害しないのと同様に、GPL 準拠の OSS を使用することは原則的に自由であり、自社のシステム構築にあわせてその OSS を改変して利用する場合も複製や公衆送信しなければ、そのソースコードを公開する義務は生じないと述べている²³⁾。セルベッサの場合、食材等の受発注ソフトウェアであるので、食材等の仕入先企業もセルベッサを使うことになる。しかし、これはソフトウェアの再配布でもソフトウェアの利用でもないと見ることができ。したがって、セルベッサのソースコードをいくらか改変しても、それを自社の業務で使っている限り、ソースコードの公開義務は発生しないという解釈ができる。

第2の理由は、後述するように改変されたソースコードが公開されなくても、利用企業やサポートベンダーが増加することがニュートキーヨーの便益をある程度増大させるからである。

亜種が発生しているのは、セルベッサのソースコードを管理しているコミュニティや組織が存在していないことに原因の一つがある。

そもそも、ダウンロード用のサイトは、ダウンロードした利用者进行を特定できる仕組みになっていない。ダウンロードサイトを運営するオープンソース・ジャパンによれば、RPM 版を公開した当初は、メールアドレスが実在するものかどうかをチェックしていなかったため、架空のメールアドレスを入力してソースコードをダウンロードする利用者が多かったという。そこで、ダウンロードに必要な情報を入力されたメールアドレスに送るという方法に変更した。ところが今度は、アドホック

22) 正式には GNU General Public License、FSF (Free Software Foundation) が提唱しているフリーソフトウェア用のライセンスの一つであり、主に GNU プロジェクトで開発されたソフトウェアや、その派生物などに適用されている。ソースコードの公開のほか、ソースコードを含めた再配布や改変の自由、再配布や改変の自由を妨げる行為の禁止などが条件となっている。

23) 垣内 (2003)

クなフリーメールのアドレスを利用してダウンロードする利用者が増えたのだという。

こうした状況を考えると、オリジナルなセルベッサの著作権を保有しているニュートキーヨーやダウンロード・サイトを運営しているオープンソース・ジャパンが把握していない企業がセルベッサを利用している可能性がある。

3.4. バグ情報などの共有

ソースコードを管理しているコミュニティや組織がないため、基本的にセルベッサのバグ（プログラム上の不具合）に対する修正は、それぞれの利用企業が個別に行っていることになる。初期においては、ニュートキーヨーの湯澤氏とテンアートの中心となってバグ情報などの共有が行われていた事実はあるし、現在でも関係者の間でバグ情報などを電子メールで交換することがボランティア的に行われている。ただし、これらの活動は組織的なものではない。

インターネット上で公開されているものは、ソースコードとシステム仕様書、利用マニュアルである。ただし、バージョン2と基本的に同じであるRPM版に対応する利用マニュアルは、利用企業であるWDIが作成したものである。

また、セルベッサの普及活動は、関係者がそれぞれ独自に実施しており、組織横断的な取組みはなされていない。

4. 業務系ソフトのOSS化がもたらす便益

4.1. セルベッサをOSS化したニュートキーヨーの狙い

ニュートキーヨーがセルベッサをOSS化した狙いはどこにあるのだろうか。誰もが、開発に

2,000万円を投じたソフトウェアを無償で公開した理由を不思議に思うに違いない。

ニュートキーヨーの狙いは、大きく次の2つに分けられる²⁴⁾。

第1は、食材等の受発注システムの開発・運用・メンテナンスに要する費用を可能な限り低く押さえつつ、できるだけ良質なシステムを構築することである。まず、OSS化すると宣言すれば、開発者のモラルアップを通じてソフトウェアの質が向上することが期待できる。他人がソースコードを見ると思えば、プログラマは整ったコードを書こうとするであろうし、多くの企業が利用する可能性があると思えばよいプログラムを書こうという意欲が増すだろう。また、ソフトウェアをOSS化して利用者が増えれば発見されるバグの数が増え、運用中に不具合が起きる可能性が小さくなる。更に、ライセンスにGPLを採用すれば、他社が機能強化を行った場合、そのソースコードを公開しなければならなくなり、ニュートキーヨーは無償でその機能強化版を利用することができる。つまり、最初の開発コストを負担すれば、その後の保守や機能強化に要する費用を節約でき、また同時にソフトウェアの質の向上も期待できるのである。

第2は、システムがメンテナンス不能になるリスクを低減することである。理論的にはソースコードと関連ドキュメントがあり、ある程度技術力があるベンダーであれば、どんなシステムでも運用や保守は対応可能で、システムがメンテナンス不能になることはないはずである。しかし、現実には、そのシステムを開発したベンダーが倒産したとか、担当したソフトウェア技術者がそのベンダーを辞めてしまったという理由でメンテナンス不能になるケースがある。OSS化することによってセルベッサをサポートできるベンダーが増えれば、セルベッサがメンテナンス不能になるリスク

24) ニュートキーヨーがセルベッサをOSS化した狙いは、湯澤（2004）pp.27-64に詳しく述べられている。

を大幅に低減できる。

4.2. ニュートキーの狙いは実現されたか

では、ニュートキーの狙いはどの程度達成されたのだろうか。

まず、OSS化を宣言することによる開発者のモラルアップを通じたソフトウェアの質の向上については、セルベッサの場合には実現できたとは言いがたい。なぜなら、湯澤氏は「セルベッサのOSS化は最初から決めていた」と述べているが、開発を担当した山崎氏は「セルベッサの開発を始めた時にはOSSになるとは思っていなかった」、OSSにするとと言われて「人に見られてもはずかしくないものにしなければいけないと思った。しかし、実際にはソースコードの書き直しをしたわけではない」と述べているからである²⁵⁾。

利用者の増加によってバグの発見が増え、ソフトウェアの品質が向上したかという点については、かなり実現されたと評価できる。例えば、当時WDIでセルベッサの導入を担当していた戸田和人氏は「WDIはかなりセルベッサを使いこなし、不具合（バグ）もたくさんみつけ、バグフィックスにはずいぶん貢献した。ニュートキーが踏んでいない地雷がたくさんあった。安定するまでに3ヵ月かかった記憶がある」と述べている²⁶⁾。同時期にセルベッサの利用を開始した三井食品や大戸屋でも、WDIと同じように不具合が発見されたに違いない。ニュートキーの狙いどおり、利用企業が増えるにしたがってプログラムに潜在するバグが発見され、セルベッサは質の高いソフトウェアになっていったと考えられる。

また、他社が機能強化したバージョンを無償で入手できるという点についても、現実のものとな

っている。現在、ニュートキーが利用しているセルベッサはバージョン2であるが、これは三井物産がASPサービスを行うために改善したものである。ニュートキーの湯澤氏は「この（バージョン2の）開発には三井物産がかなりの開発費をかけたはず、おそらくニュートキーがバージョン1を開発した時に投じた金額より多いのではないか。その新しいバージョンを無償で利用できているのだから、それだけでもメリットがある」と述べている²⁷⁾。

したがって、OSS化によってシステムの保守や機能強化に要する費用を節約でき、また同時にソフトウェアの質の向上を図るという狙いはかなり実現されている。

更に第2の狙いであるセルベッサをサポートするベンダーを増やすことにも成功している。「3.2. セルベッサの様々なバージョン」でみたように、現在セルベッサをサポートしているベンダーは、テンアートニ、オープンソース・ジャパン、ソフトリンクの3社に増加している。ベンダーが増えたことによって、セルベッサがメンテナンス不能になるリスクは極めて小さくなったと言えるだろう。

4.3. 利用企業とベンダーのメリット・デメリット

現時点では、セルベッサは企業が自社利用のために費用をかけて開発した業務系ソフトをOSS化した希な事例である。今後、セルベッサのような事例が増えるのかどうかを考えるために、業務系ソフトをOSS化した場合のメリットとデメリットを検討する。

ここでは「オリジナル利用企業」「オリジナル・

25) 2005年1月31日の湯澤一比古氏へのインタビュー、及び2005年3月17日の山崎靖之氏へのインタビューによる。

26) 2005年5月18日の戸田和人氏へのインタビューによる。

27) 2005年1月31日の湯澤一比古氏へのインタビューによる。

ベンダー」「二次的利用企業」「二次的ベンダー」の4つに分けて考える。オリジナル利用企業とは、その業務系ソフトの最初の利用企業で、初期開発コストを負担した企業であり、オリジナル・ベンダーはそのオリジナル利用企業からの委託を受け、そのソフトを開発したベンダーである。二次的利用企業とは、OSS化された業務系ソフトを利用する企業であり、二次的ベンダーはその利用をサポートする企業でオリジナル・ベンダー以外のベンダーである（図表3参照）。

4.3.1. オリジナル利用企業

オリジナル利用企業のメリットは、ニュートーカーのセルベッサ OSS化の狙いと同じように大きく2つに分けられる。

第1に、開発時に OSS化することを宣言すれば、プログラムのモラルアップを通じてソフトウェアの質が向上することが期待できる。また、OSS化したソフトウェアの利用企業が増えれば、発見されるバグの数も増えてソフトウェアの質がよくな

る。更にライセンスに GPLを採用すれば、他の企業がプログラムを改変した場合、その改変したプログラムのソースコードが公開されることになり、機能強化されたバージョンを無償で入手できる。

第2に、OSS化されたソフトウェアの利用企業の増加によって二次的ベンダーが生まれれば、特定の技術者やベンダーに依存しないですむ状況になり、オリジナル・ベンダーの倒産などによってシステムがメンテナンス不能になるリスクを大幅に小さくできる。

一方、オリジナル利用企業のデメリットは、まず、初期開発コストを1社で負担するために二次的利用企業に比べて IT投資コストが大きくなり、競争上不利になる可能性があることが挙げられる。ただし、二次的利用企業の IT投資コストがゼロになるわけではない。OSSを利用する場合でも、業務システムとして利用できるようにするためには、かなり手をかけなければならない。その OSSに合わせたデータの整備や DBMSなどの他のソフト

図表3 業務系ソフトの OSS化に関するメリット・デメリット

	オリジナル	二次的
利用企業	<ul style="list-style-type: none"> ○潜在するバグが発見され質が良くなる ○機能強化版を無償で入手できる ○OSS化宣言による開発者のモラルアップ ○特定のベンダーに依存しないで済む ●初期開発費用を負担（二次的利用企業の IT投資コスト削減を通じ、企業競争に不利に働く） ●二次的利用企業が増えないとメリットが生まれない→普及活動にコストがかかる 	<ul style="list-style-type: none"> ○無償で業務アプリケーションのソースコードを入手できるため IT投資コストを削減できる ●サポートしてくれるベンダーを捜す必要がある ●パッケージなどに比べバグが多い可能性あり ●機能追加、改修が必要なことがある
ベンダー	<ul style="list-style-type: none"> ○利用企業が増えれば、機能拡張やメンテナンス、運用サポートの受託機会が増える ●新規開発の機会が失われる ●ソースコードを（競合ベンダーに対しても）公開しなければならない 	<ul style="list-style-type: none"> ○ソースコードや関連ドキュメントを学ぶことによって当該分野に進出できる ○機能拡張やメンテナンス、運用サポートなどの受託機会が増える ●新規開発の機会が失われる ●GPLに従えば改修、機能追加部分などを含めてソースコードを公開しなければならない

（出所）EJB™コンポーネントに関するコンソーシアム（2005）

ウェアとの調整が必要なことが多い。また、ハードウェアやネットワークに対する投資も必要である。したがって、巨額の開発費を投じて開発したソフトウェアでない限り、競争優位に与える影響は限定的だと考えられる。

もう一つのデメリットは、二次的利用企業が増加しないと先に挙げたメリットの大半が実現しないことである。したがって、オリジナル利用企業は OSS 化のメリットを享受するために普及活動を行うことになるだろう。

4.3.2. オリジナル・ベンダー

オリジナルベンダーのメリットは、その業務系 OSS の利用企業の増加とともに、その運用、サポート、メンテナンス、機能強化といったビジネス機会が増えることである。しかし、もしその業務系ソフトが OSS 化されなければ、同様の業務システムを新規開発するというビジネスチャンスがあったはずであり、ベンダーにとっては新規開発の機会を損失していることになる。これは大きなデメリットである。また、OSS 化されれば、ソースコードを通して業務ソフトに内包されたノウハウなどを他社に公開してしまうことになる。

4.3.3. 二次的利用企業

OSS 化された業務系ソフトを利用するユーザー企業は、無償でそのソースコードを入手できるため、IT 投資コストを削減できるというメリットがある。

問題は、パッケージ・ソフトに比べてバグが多い可能性があること、自社の業務にあわせて機能強化やカスタマイズが必要になるケースが多いこと、サポートベンダーを捜す必要があることである。

セルベッサに見られるように、ダウンロードしたソフトウェアは、カスタマイズや機能強化が必要になる可能性が高い。しかし、それは業務パッケージを導入した場合とあまり変わらない可能性がある。一方、ゼロから同様の機能をもった業務

アプリケーションを開発することに比べれば、コストは大幅に安くなるだろう。

4.3.4. 二次的ベンダー

業務系 OSS を利用する二次利用企業をサポートするベンダーにとってのメリットは、まずそうした運用サポート、メンテナンス、場合によっては機能拡張のためのプログラムの改変といった業務の受託機会が増加することである。しかし、オリジナル・ベンダーと同様に、業務システムを新規開発するという機会を失っていることになる。更に、その業務系 OSS のライセンス方式が GPL である場合には、ソフトウェアを改変したり、他のソフトウェアとリンクした場合、リンクしたソフトウェアを含めてすべてのソースコードを公開しなければならないというというデメリットもある。

一方、当該分野にあまり詳しくないベンダーにとっては、その業務系 OSS のソースコードや関連ドキュメントを学ぶことによって、新規分野への参入が容易になるというメリットもあると考えられる。

5. 業務系 OSS が増加・普及する可能性

5.1. 業務系 OSS とインフラ系 OSS との違い

同じように GPL に従ってソースコードが公開されていても、Linux に代表されるインフラ系の OSS と業務系の OSS では環境が異なる。

第1に、インフラ系の場合には、バグの発見やその修正、機能強化などの作業は多くのボランティアによって行われている。これは、そのソフトウェアにどれだけ貢献したかがコミュニティにおける個人の評価につながり、ソフトウェア技術者としての名誉が一つのモチベーションになっているからである。一方業務系の OSS の場合には、インフラ系の OSS のようなコミュニティが存在し

ないか、あるいは規模が極めて小さく、業務系ソフトを開発しているソフトウェア技術者に OSS に関心を持つ人が少ないこともあり、ボランティアによる機能や質の向上はほとんど期待できない。

第2に、業務系 OSS の場合、あるユーザーによって行われた機能追加が、他の大多数のユーザーにとって無用の長物である可能性があることである。インフラ系のソフトウェアの場合、利用者が必要とする機能はかなり共通であるが、業務系の場合にはインフラ系に必要とする機能は利用者によってバラツキが大きい。したがって、ある企業が業務系 OSS に必要な機能を追加しても、その機能は他の企業では不要なものであるケースがあるだろう。つまり、改変した部分をオープンにしてフィードバックしても他の利用企業から感謝されないということが起きる。

第3に、業務系 OSS では亜種の存在は利用上問題にならない。インフラ系 OSS の場合、異なったバージョンがいくつも存在すると、その上で動くソフトウェアとの整合性やネットワークを通じたデータ交換において支障が発生する。このため、インフラ系 OSS では、コミュニティや団体、オリジナルベンダーによるバージョン管理が行われている。しかし、業務系 OSS は、それぞれの企業による利用範囲が閉じていることもあり、異なったバージョンが存在していてもトラブルになる可能性は少ない。したがって、業務系 OSS の場合には、数多くの亜種が存在していても、利用企業にとって問題はあまりないと考えられる。

5.2. ソースコードのバージョン管理は必要か

業務系 OSS の場合、バグの修正や機能強化は利用企業によって行われ、機能の中には利用企業が共通して利用しないものがあり、また亜種が数多く存在していてもトラブルになることは少ない。こうした性質を考えると、業務系 OSS に関しては、ソースコードのバージョン管理は必要ないと考え

てよいのだろうか。

おそらくそうではない。新規にその業務系 OSS を採用しようという企業にとっては、多くの亜種が存在している場合、どのバージョンが最適であるのか見極める作業が必要になる。普及のためには、バージョンがある程度管理されていることが望ましい。

また、業務系ソフトウェアには、どの企業も利用するコアの部分と、企業によって必要性が異なる周辺部分があると考えられる。ある企業が業務系 OSS に改変を加えた場合、それがコアの部分に属し、他の企業にも役立つものであれば、その改変は他のバージョンにも反映されるべきものである。また、逆にある企業による改変部分が、別の企業の改変部分と不整合を起こす可能性もあり、バージョン管理には細心の注意が必要とされるだろう。

つまり、理想を言えば、業務系 OSS であっても、ソースコードのバージョン管理が行われることが望ましい。ただし、こうした管理を行うには、それなりの組織とコストを要する。その管理コストが、管理することによって得られる便益より大きいならば、管理しない方が合理的だと考えられる。また、管理することによって得られる便益が管理コストより大きくても、その管理コストを誰がどのように負担するのかという問題は残される。

5.3. 業務系 OSS の普及経路

今後、業務系 OSS が増加し、普及していく可能性を考えるため、まず業務系 OSS が普及する経路を、オリジナル利用企業がある場合とない場合(オリジナル・ベンダーが独自開発した業務系ソフトを OSS 化する場合)に分けて考えてみる。

5.3.1. オリジナル利用企業がある場合

セルベッサのように、利用企業が業務系ソフトウェアを OSS 化すると決めて開発した場合、オリジナルな利用企業(セルベッサで言えばニユート

一キョー)は、二次的利用企業や二次的ベンダーが増えないと OSS 化のメリットは享受できない。したがって、オリジナル利用企業が普及活動をする事は極めて合理的な行動である。

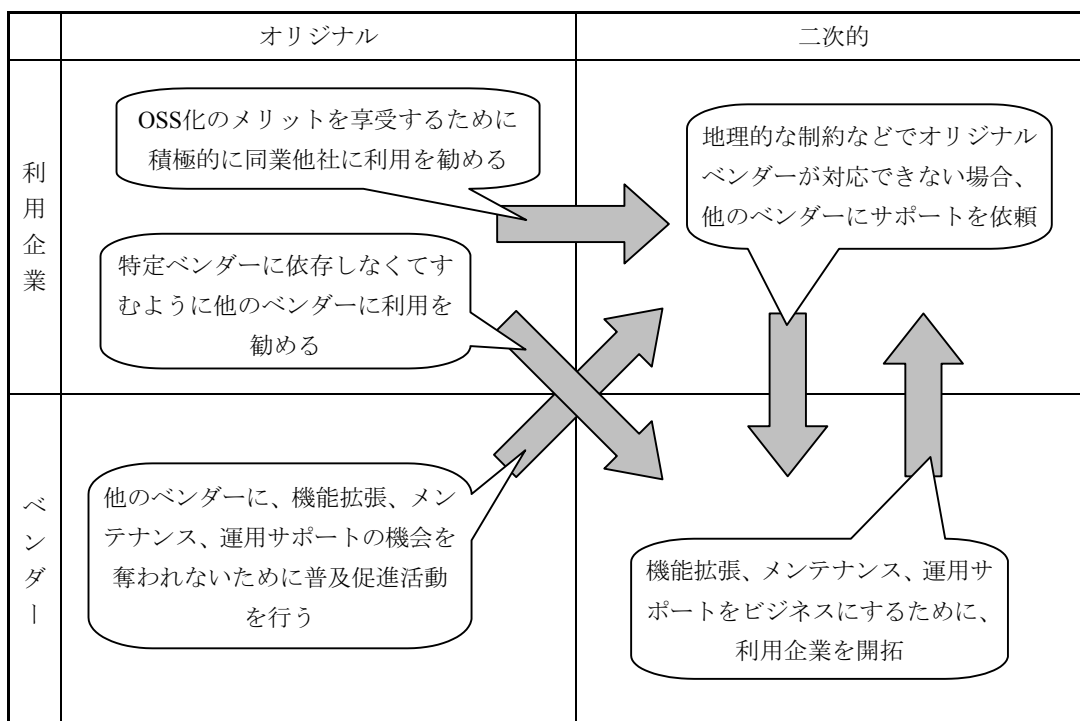
オリジナルベンダーにとっては、業務系 OSS の普及活動を行えば、同種の業務系ソフトの新規開発機会を失うことになるので、積極的に普及活動をしなことが合理的であるように見える。しかし、オリジナル利用企業は積極的に普及活動をするという前提に立てば、普及活動に消極的であると、当該業務系 OSS の機能拡張、メンテナンス、運用サポートの受託といったビジネス機会を他のベンダー (二次的ベンダー) に奪われることになる。したがって、オリジナル利用企業が普及活動をするなら、オリジナルベンダーも普及活動をするという戦略をとることが合理的だということになる。

一方、その業務系 OSS を利用する可能性のある同業他社は、ゼロからシステムを開発するより IT 投資コストを削減できるため、当該業務系 OSS を利用しようとする可能性が高い。

仮に二次的利用企業が増加し、オリジナルベンダーが受容能力面の限界あるいは地理的な要因から新規の利用企業をサポートできないケースが出てくれば、その企業の働きかけによって、その業務系 OSS をサポートするベンダー (二次的ベンダー) が出てくるという可能性がある。また、オリジナル利用企業にとっては、二次的ベンダーが生まれることは大きなメリットであるので (特定のベンダーに依存しない状況をつくり出せることになるので)、他ベンダーにサポートを働きかける可能性が高い。

図表 4 は以上をまとめて図示したものである。

図表 4 業務系 OSS の考えられる普及経路
(利用企業が OSS 化する場合)



(出所) EJB™ コンポーネントに関するコンソーシアム (2005)

5.3.2. オリジナル利用企業がない場合

業務系 OSS としては、ベンダーが自ら開発した業務ソフトを OSS として公開するケースも考えられる。

この場合、オリジナルベンダーは積極的に普及活動を行うことになると考えられる。なぜなら、第1に OSS 化した理由は、そのソフトウェアを普及させようという意思があると考えられるからである。第2に OSS 化して普及活動をしなければ、他のベンダーが普及活動をして機能拡張、メンテナンス、運用サポートなどのビジネス機会を奪ってしまう可能性があるからである。

オリジナルベンダーが二次的ベンダーに普及を働きかけることは考えにくい。ライセンス料を徴収できるパッケージソフトであれば、サポートベンダーが増えることによって、ライセンス収入が増えるというメリットがあるが、OSS の場合は無

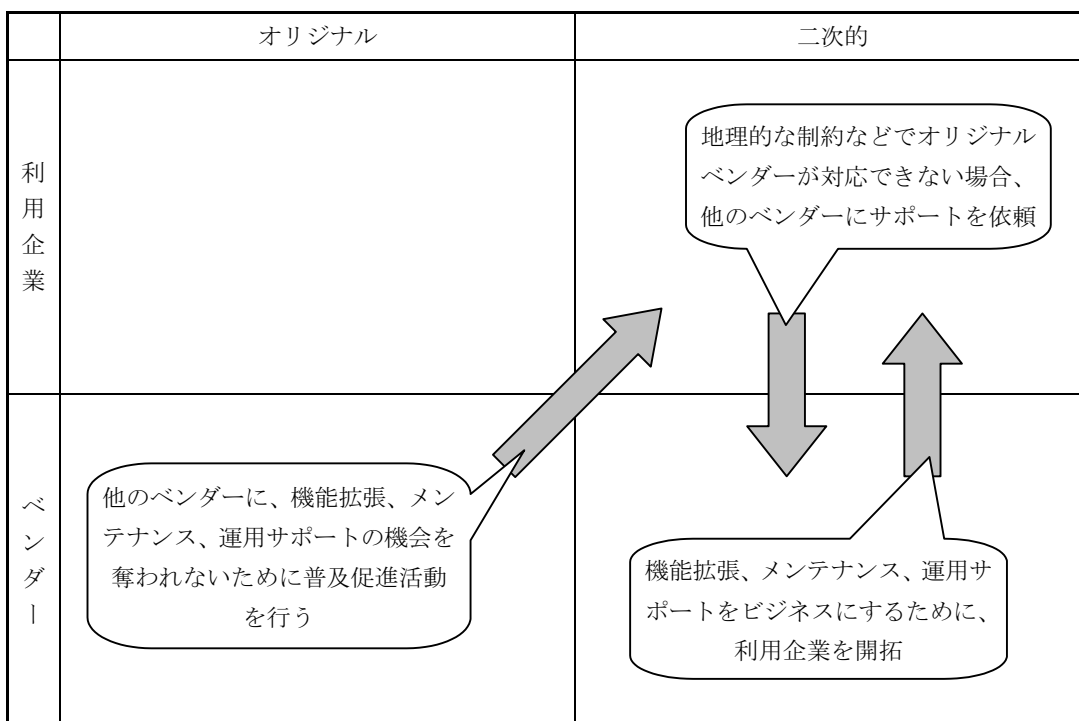
償で利用できるため、オリジナルベンダーにサポートベンダーを増やそうというインセンティブは働かない。

オリジナルベンダーの働きかけによって二次的利用企業が増加すれば、二次的ベンダーが出てくるという可能性があるのは、「オリジナル利用企業がある場合」と同じである。

また、二次的ベンダーが生まれれば、この二次的ベンダーが顧客開拓を行うことになると考えられるのも「オリジナル利用企業がある場合」と同じである。

なお、オリジナルベンダーが自社開発の業務系ソフトを OSS 化する最も一般的ケースは、ベンダーが、コストをあまりかけずに当該ソフトウェアを普及させ、そのサポートサービスで収益を上げる戦略を取る場合だと考えられる。とすれば、自社開発の業務系ソフトを OSS 化するオリジナル

図表5 業務系 OSS の考えられる普及経路
(ベンダーが OSS 化する場合)



(出所) EJB™ コンポーネントに関するコンソーシアム (2005)

ベンダーは、比較的規模の小さなベンダーやベンチャー系のベンダーが多くなるだろう。

5.4. 業務系 OSS が普及する条件

これまでの考察から、業務系 OSS が普及するためにはどのような条件が必要なのかを考えてみたい。

5.4.1. 当該分野で十分な品質・機能をもつ安価なパッケージソフトがないこと

もし、その分野で十分な品質・機能をもつ安価なパッケージソフトがあれば、それと同等以下のソフトウェアが OSS 化されても、利用企業にとってその業務系 OSS を利用するメリットは小さく、普及の可能性は小さいと考えられる。ただし、十分な品質・機能をもつパッケージソフトがあっても、その利用コストが高ければ、業務系 OSS の普及の可能性はある。

5.4.2. 当該業務系ソフトを開発し OSS 化する利用企業あるいはベンダーにとってメリットが大きいこと

当然のことながら、誰かがその業務系ソフトを開発し OSS 化しようとしなければ、普及はありえない。可能性があるのは、利用企業が自社用の業務系ソフトを OSS 化する場合（「セルベッサ」のケース）とベンダーが自社開発した業務系ソフトを OSS 化する場合である。

利用企業が、開発コストを負担した業務系ソフトを OSS 化することは、あまり合理的な選択ではないように思える。それは、同業他社がそのソフトを無償で入手することになり、情報投資を削減すると同時にソフトの利用を通じて業務ノウハウを得ることが可能になるからである。つまり、OSS 化によってライバル企業を利することになるからである。しかし、「4.3. 利用企業とベンダーのメリット・デメリット」で考察したように、OSS 化によってオリジナル利用企業が得られるメリットは少なくない。このメリットが、他社に無償で

ソフトを利用されるデメリットより大きければ OSS 化は合理的な選択だと言える。

では、ベンダーが、自主開発した業務系ソフトを OSS 化するメリットはどこにあるのだろうか。このタイプである美容院向け WebPOS システムの「フランシーヌ」や Linux サーバー上で動くグループウェアの「ペンギンオフィス」のケースをみると、これらの開発企業は、無償でダウンロードして試用できるためユーザーの裾野が広がること、GPL によるライセンスなので他社による機能追加などを取り入れて製品を強化できることがメリットであると説明している。手軽に試用できれば、その中にはサポートや運用管理、機能追加を依頼してくる企業が出てくるだろう。パッケージソフト化して宣伝広告するより、効果的かもしれない。また、パッケージソフトの場合、自社で機能強化のコストを負担しなければならないが、OSS にしておけば、機能強化を利用企業の負担で実施できる可能性や、他のベンダーが機能強化してくれる可能性も出てくる。ただし、パッケージソフトにした場合に比べて、ライセンス料による収入は得られなくなるし、他のベンダーにビジネスの機会を奪われる可能性もある。

5.4.3. OSS 化のメリットを理解できる CIO（あるいは経営者）が存在すること

「5.3. 業務系 OSS の普及経路」で考察したように、OSS 化を行った企業にはその業務系 OSS を普及させようというインセンティブが働く。セルベッサの場合、ニュートキーの湯澤氏がセルベッサの普及に大きな役割を果たしているが、これも経済合理性のある活動であったと解釈することができる。

問題は、業務系ソフトウェアを OSS 化することのメリットを理解できる CIO あるいは経営者がいるかどうかである。

また、業務系 OSS があまり普及していない現状では、その利用メリットを説く湯澤氏のように熱

心なエバンジェリストが必要とされるだろう。

5.5. 業務系ソフトを非公開にする意味はない

利用企業が委託開発した業務系ソフトは、これまでその企業内で利用され、ソースコードが公開されることはほとんどなかった。つまりカスタムメイドの業務系ソフトは非公開であることが常識であり、実際にそうであった。

しかし、業務系ソフトがこれまで非公開とされてきたのは、本当に合理的な理由があったからだろうか。開発費を負担したのだから、その業務系ソフトは自社のものだ。自分のものをタダで他人にあげる必要はない。同業他社を助けるようなこととしてはいけない。そう考えてきたのではないだろうか。

そもそも、ソフトウェアの複製コストはほとんどゼロであり、公開したものを他社が利用したからといって、自社で利用できなくなるものではない。ソフトウェアは（あるいは情報財は）本質的に非競争性を有しており、公共財的な性格を持っている²⁸⁾。

セルベッサの事例を分析すると、業務系ソフトを公開（OSS化）することは、二次的利用企業だけでなくオリジナルな利用企業の便益をも増大させることがわかった。つまり、OSS化こそが合理的であり、非公開にすることの方が非合理的なのである。

もちろん、その業務ソフトが、その企業の持続的競争優位の源泉になっている場合には、非公開であることに意味があるという反論があるだろう。そのようなソフトを公開すれば、その企業の競争優位が失われ、計り知れないダメージを受けるか

らである。しかし、ニコラス・カーの主張²⁹⁾のように、もはやITによって持続的な差別化を図ることは困難になっており、ITには戦略的な価値はなくなっているのだとすれば、OSS化すべきではない競争優位の源泉となっている業務系ソフトは存在しない。競争優位の源泉となるような業務系ソフトが存在するかどうかについては、意見が分かれるところだろうが、ここでは論じない。

仮に競争優位の源泉となるような業務系ソフトが存在したとしても、そのようなものを除けば、業務系ソフトをOSS化することは、極めて合理的な選択だと言える。現時点では、ほとんどの企業がカスタムメイドの業務系ソフトを非公開にしているが、それは業務系ソフトをOSS化するという発想や文化が無かったからである。業務系ソフトを非公開にしたからといって、それがその企業にとって何か新しい価値を生むわけではない。むしろ、OSS化することによって、ソフトウェアの質の向上や機能強化バージョンの無償入手の可能性が生まれる。また、社会全体にとっては重複投資の排除というメリットが生まれる。

業務系ソフトをOSS化することのメリットや意義が社会に浸透していけば、OSSとして公開される業務系ソフトが増えていくに違いない。

6. 結論

6.1. 業務系OSSの普及とITベンダーの対応

業務系ソフトのOSS化は、オリジナル利用企業にとってもメリットが大きいことがわかった。これまで業務系ソフトがOSS化されてこなかったのは、前例がほとんどなく、OSS自体の歴史が浅

28) 公共財にはもう一つ「消費の排除不可能性」という条件を満たすことが必要とされる。

29) Nicholas G. Carr は、Carr (2003) と Carr (2004) において「ITは電話や電力、鉄道などの基盤的技術と同じように、技術的な成熟に伴いコモディティになり、もはや企業にとって持続的な競争優位の源泉ではなくなっている」という主張を展開している。

いこともあって、OSS化のメリットが理解されていないことが原因だと考えられる。

今後、業務系ソフトをOSS化するメリットが企業経営者やCIOに理解されるようになれば、業務系OSSの種類は急速に増えていく可能性があるだろう。

業務系OSSの種類が増え、その利用が拡大していくことは、似て非なるソフトウェアの開発に投じられる費用を削減することができるという点で、経済全体にとってプラスである。しかし、ベンダーの立場で考えると、新規に業務系ソフトを開発する案件が減少するという大問題に直面することになる。もちろん、ソフトウェアの新規開発というビジネスがなくなるわけではないが、少なくともOSS化された業務ソフトに類似したソフトウェアを新規に開発するという案件は減少するだろう。

また、ソフトウェアが無償で入手できるようになっても、それを実際に業務で使えるようにするためにはカスタマイズなどの作業が必要であるし、業務システムの運用をサポートする、あるいは一部の機能を強化する、新規機能を追加するというビジネスは逆に増加するだろう。

したがって、業務系OSSが普及していけば、ベンダーはソフトウェアの新規開発よりも、サポート・サービス、運用サービス、カスタマイズに重点をおくビジネスモデルへの転換が必要となるだろう。

6.2. ソフトウェアに起きる究極の価格破壊

OSSは無償で入手して使用することができる。ソフトウェアを新規開発すれば開発費が必要であるし、既存の商用パッケージ・ソフトウェアを使えばライセンス料を支払う必要だが、OSSを使えば、開発費もライセンス料も支払う必要がない。様々な分野でOSSが増えていけば、ソフトウェアは無料で入手できることになる。

情報技術の世界では様々な価格破壊が進行しているが、ソフトウェアの無料化は低価格化とは質の異なる影響をもたらすに違いない。

もちろん、現在、あらゆる分野にOSSが普及し、ソフトウェアの価格破壊が起きているというわけではない。ウェブサーバー用ソフト分野で7割程度のシェアを持つApacheのような事例は希で、多くの分野では、OSSの利用は限定的であり、市販のパッケージ・ソフトウェアやカスタムメイドのソフトウェアを利用するユーザーが多い。

ただ、「1.4. インフラ系OSSと業務系OSS」で述べたように、OSSの利用は徐々に拡大している。例えば、OS分野のLinux、DBMSのMySQLやPostgreSQL、デスクトップ・ソフトウェアのOpenOffice、プログラミング言語のPHP、アプリケーション開発環境ソフトのEclipseなどは、市販のソフトウェアと変わらない機能・品質を備えており、徐々に利用者が増えている。

そして、OSS化の波は業務ソフトの分野にまで押し寄せてきている。セルベッサの事例で検証したように、利用企業が自社向けに開発したソフトウェアをOSS化することは十分経済合理性のある行為である。開発したソフトウェアを秘密にしておくより、OSSとして公開した方がメリットは大きいのである。現時点では、業務系OSSの種類はまだまだ少なく利用企業もさほど多くない。しかし、今後、業務系OSSの種類もその利用はゆっくりとではあるが、着実に拡大していくだろう。

OSSの普及によって、ソフトウェアの世界で価格破壊が起きるとすれば、その価格破壊は情報サービス産業界に極めて大きな影響を与えるだろう。1990年代以降、ハードウェアの世界ではダウンサイジング、オープンシステム化によって価格破壊が進行した。また、1990年代後半からはインターネットとIPネットワークの普及によってネットワークの世界で価格破壊が起きた(図表6参照)。

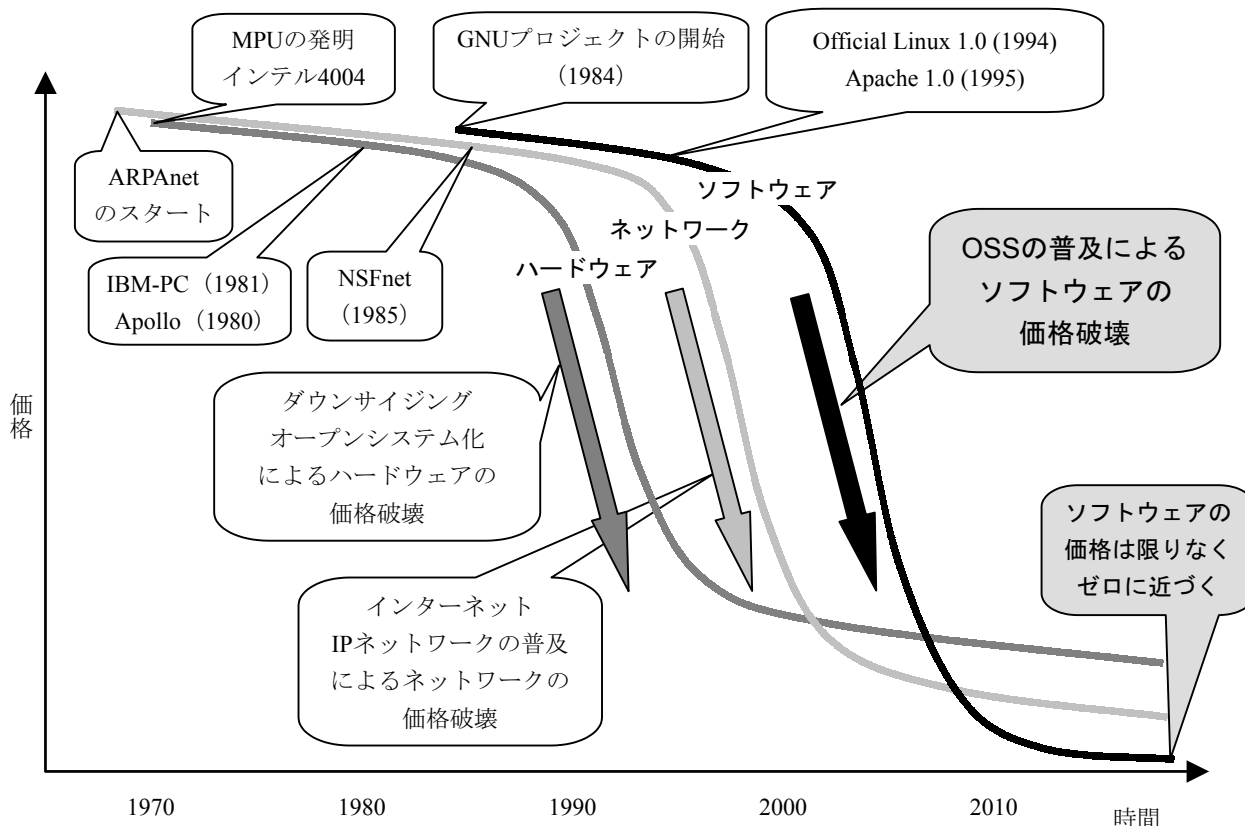
しかし、ハードウェアもネットワークも価格がゼロになったわけではない。しかし、ソフトウェアの場合には再生産（コピー）と配布に要するコストはほとんどゼロになる。また、ハードウェアは老朽化や陳腐化による買い替え需要が発生するが、ソフトウェアは朽ち果てることはなく、OSSの場合には、更に機能強化されたバージョンが無償で配布される可能性がある。これは究極の価格破壊だと言ってよいだろう³⁰⁾。

つまり、OSSの普及は、ソフトウェアに起きる究極の価格破壊なのである。

【参考文献】

Carr, Nicholas G., 2003 “IT Doesn’t Matter”, *Harvard Business Review*, May 2003, pp.41-49
 Carr, Nicholas G., 2004 *Does IT Matter?*, Harvard Business Press
 EJB™ コンポーネントに関するコンソーシアム 2005 『市場流通のための再利用事例研究報告書（テクニカルレポート）』 2005.9
 情報処理振興事業協会 2003 「オープンソース・ソフトウェアの現状と今後の課題について」
 垣内郁榮 2003 「オープンソースで損害賠償？ ビジネ

図表6 IT分野における価格破壊



(出所) 富士通総研作成

30) Tim O'Reilly は OSS がもたらす効果の一つは「ソフトウェアのコモディティ化」であると述べている (“Open Source Paradigm Shift” http://tim.oreilly.com/articles/paradigmshift_0504.html) が、これは正確ではない。コモディティは市場で取引される（つまり価格が存在する）が、OSSの普及はソフトウェアの価格をゼロにする。

- スにリスクはないか」@IT News Insight, 2003年8月2日、<http://www.atmarkit.co.jp/news/200308/02/oss.html>
- 村上敬亮 2004「オープンソースを巡る著作権論議と知的財産政策への示唆」特技懇 No.232 2004.3.30 pp.16-28
- 佐々木裕一、北山聡 2000『Linux はいかにしてビジネスになったか コミュニティアライアンス戦略』NTT出版
- 竹田陽子、米山茂美 2002「[ビジネスケース] セルベッサ ニュートキーの食材発注システムはなぜ公開されたのか」『一橋ビジネスレビュー』2002年 Winter pp.2-21
- 湯澤一比古 2004『オープンソースじゃなきゃ駄目』（アイデア教養文庫01）アイデア出版