入門オープンソース

~その哲学と意義~

Ver. 1.2

リナックスアカデミー矢越昭仁

2013/06/19

Linux や Apache HTTPD を始め、インターネット上でデファクトスタンダード（事実上の標準）となっているソフトウェアの特徴として、ソースコードを公開する「オープンソース」での提供があります。なぜ設計図ともいうべきソースコードを公開し、手の内を見せてしまうのか？オープンソースの特徴と時代背景や、意外と知られていない利用する上での注意点を解説します。

目次

はじめに

今では常識となったマウスとビットマップディスプレイを搭載したコンピュータ Alto が Xerox の PARC で誕生したのが 1973 年。今年でパーソナル・コンピュータが誕生して 40 年、コンピュータが誕生してから 60 年になります。

当時はハードウェアありきで非常に高価なコンピュータに、おまけとしてソフトウエアが添付されていた時代で、IT 予算のハードウェア比重が最も高かった時代です。それが 1980 年代頃になると、ソフトウェアの比重が高くなります。自社で全てを賄っていた IT 部門は、システムの複雑化と影響範囲の増大からアプリケーション・プログラムの内製を減らし外部調達へと移行し、2000 年問題の影響で加速されました。

1990 年代に入るとインターネットが普及し、コンピュータにアプリケーションを導入しなくても、ネットワーク経由で必要な仕事ができる(サービスを受ける事ができる)仕組みが充実してきます。それらは当初ホスティング、ASP(Application Service Provider)と呼ばれていましたが、規模の拡充や対応範囲の拡大に伴い、現在では SaaS(Software as a Service)と呼ばれています。さらに提供するシステムの階層によって、IaaS(Infrastructure～) / HaaS (Hardware～)、PaaS(Platform～)などと細分化されてゆきました。これらを合わせて XaaS ('X' ～)とも呼びます。

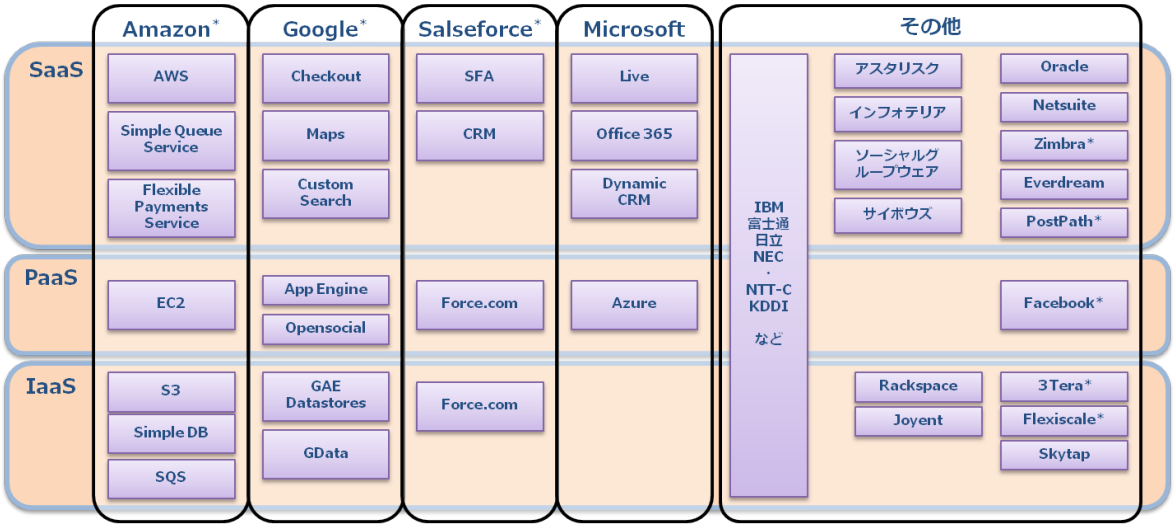


図 1:主なクラウドサービスとベンダー

ネットワークを介してサービスが提供されるようになると、大規模なデータセンターで集中管理し、種々のコストをボリュームディスカウントし単価を下げる事が可能となりました。利用したい時に欲しい分だけを使い、使用量を払うといった仕組み至ったものを古くはユーティリティ・コンピューティング(Utility Computing)、最近ではクラウド・コンピューティング(Cloud Computing)と呼んでいます。

この様な IT 業界のイノベーション(innovation、技術革新)を支えた技術の多くは OSS によって提供されてきました。

この講義では IT 業界の変遷と、OSS 誕生の背景、現状と今後を解説します。

ＩＴ産業の概要

IT産業は、コンピュータシステム上にサービスを構築することだと言えます。ハードウェアと基本ソフトを調達し、その上にアプリケーション・プログラムを構築し、サービスを提供するという手順が必要です。そういったサービスを企画してから、コンピュータシステムを設計・構築、維持管理まで必要な作業を図示すると以下のようになります。

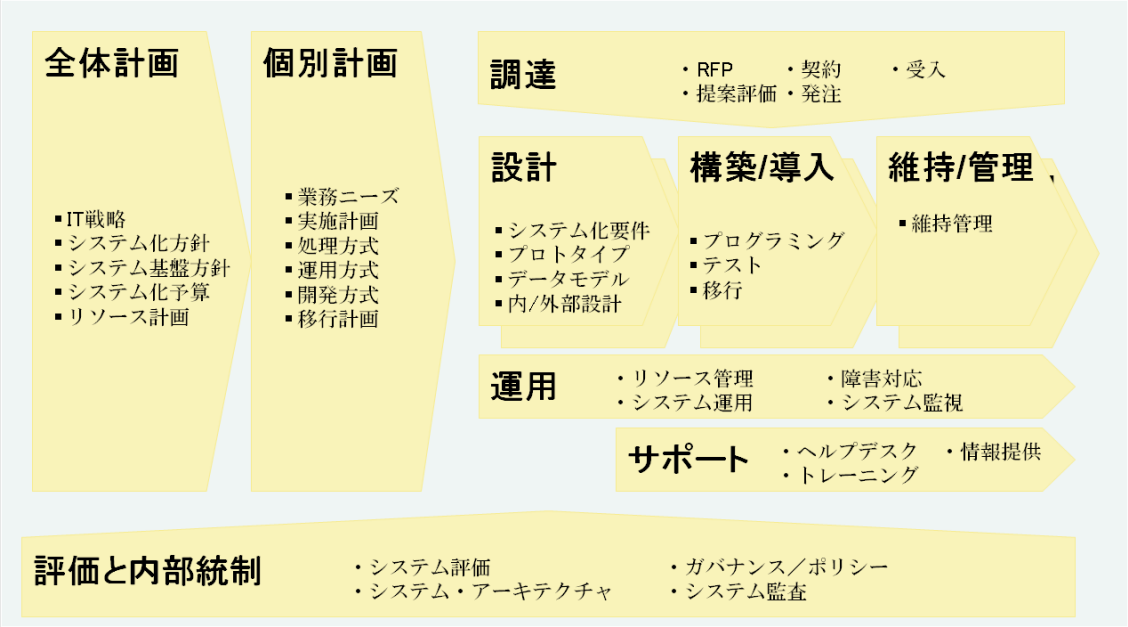


図 2:IT 業界マップ

この様にシステム構築には多くの分野があり、大規模なシステムでは1社で全てを賄う事は技術やノウハウが不足し現実的ではありません。実際にはビル建設のように、分野ごとに専門企業があり、複数の企業が一体となってサービスを構築します。

表 1:IT 業務分野と事業者

分野	部署・企業	例
企画	経営企画、コンサルティング会社	ビジネスの観点からITによって解決できる施策を検討し、中長期の計画立案 (会計事務所系コンサル、総研)
設計	システム部門、SI、メーカ	具体的なITソリューションを定義し、プロジェクト計画、予算化 (NTT-D、SCSK、日立、富士通など)
構築	システム部門、SI、ソフトハウス	プログラムの開発、テスト、文書化 (○○ソフト、Oracle、MSなど)
運用	システム部門、SI、データセンター、メーカ	日々の状況監視、定期的な処理の実行と報告 (CTC、IIJ、NTT-Com,KDDI,IBMなど)
監査	監査部、監査事務所、アプレイザー	運用ルール実施状況監査、報告、指摘 (監査法人、大和コンピュータなど)

ハードウェア産業

IT 産業のうち、ハードウェアを製造している会社は過去には沢山ありましたが、いまは淘汰され随分と少なくなりました。特に大型のサーバを提供している会社は数えるほどしかありません。国内では日立、富士通、NEC、グローバルでは IBM、HP、Sun Microsystems(現 Oracle、以下 Sun)程度になっています。

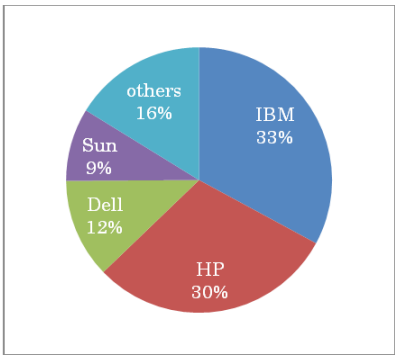


図 3:グローバルサーバメーカシェア(2010 年 IDC 調べ)

ネットワーク機器でも Cisco Systems、Juniper Networks、ハードディスク(RAID)は EMC、NetApp。PC は HP、Lenovo、Dell、Acer、Asus ですが、ほとんどのメーカが生産委託(EMS: Electronics Manufacturing Service)をしており、その多くは台湾に集中しています。

現在ハードウェア産業は成熟期にあり、新規参入は非常に難しい状況となっています。また多くの構成部品も極端な寡占状態にあります。ハードウェア産業は大手企業による M&A が活発で、これも寡占状態に拍車をかけています。たとえば Oracle による Sun 買収や、HP による 3COM、3PAR の買収などはニュースになりました。

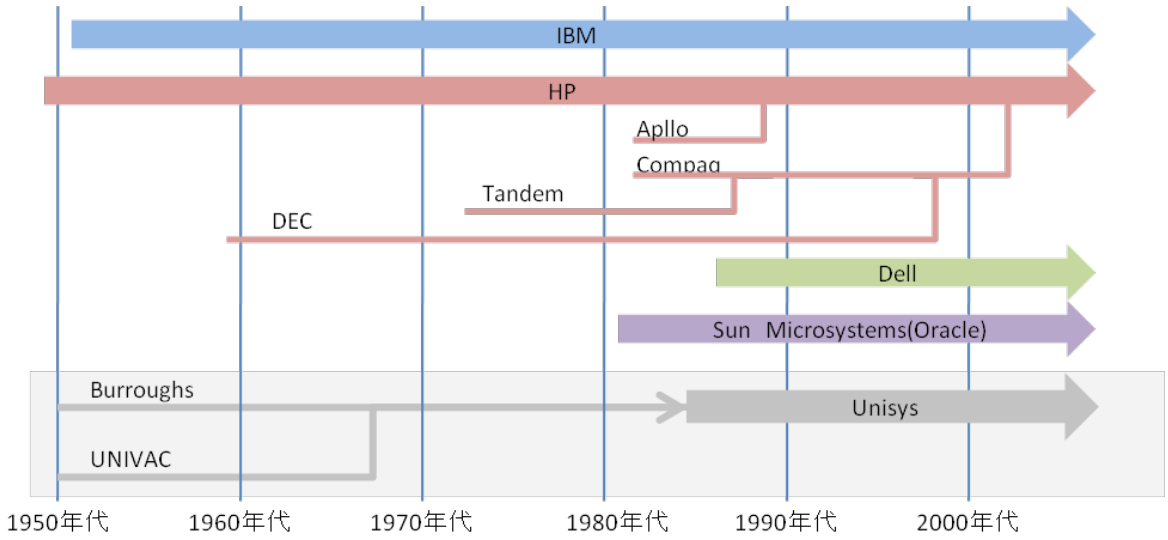


図 4:サーバーメーカの沿革

ハードウェア産業では、工場や部品在庫を維持するコストを削減するために工場を持たない企業が増えてきています。例えば携帯電話のノキア、クアルコム、それらに搭載されている CPU の ARM(英)などは設計・試作は行いますが量産は EMS[1]やファウンドリ[2]に委託しています。PC の構成部品も寡占状態にあり、HDD は Western Digital、東芝、Seagate の 3 社でシェアのほとんどを占め

[1] 電子機器の受託生産会社。Electronics Manufacturing Service。台湾の鴻海が有名。
[2] 半導体の設計・製造受託会社。Foundry 台湾の TSMC が最大

ますし、メモリは Samsung、SK Hynix、Elpida(旧日立、NEC、三菱電機)の 3 社、CPU に至ってはPCで Intel(米)、モバイルで ARM(英)がほぼ市場を独占しています。
またサーバ分野でも Sun は近年ソフトウェアに注力し、ハードウェアの多くを富士通に委託していました、他の国内コンピュータメーカも海外のコンピュータメーカと提携しハードウェアを担当している事が散見されます。
ハードウェアメーカも設計を行うソフトウェア企業へと変化し、製造委託会社が工場と人員を提供し生産する体制が一般化しつつあります。

ソフトウェア産業

ソフトウェア産業には大きく分けて、ソフトウェア製品ベンダーとシステムインテグレータ(SI: System Integrator)の 2 つがあります。製品ベンダーは基本ソフトやミドルウェア、アプリケーション・プログラムを製造し販売しています。パッケージベンダーなどと呼ばれる事もあります。最近では従来のソフトウェア(プログラム)に加え、その上で流通する情報をコンテンツとして販売する業種も登場してきました。

<table>
<tr><td colspan="2">コンテンツ</td><td>楽曲、文献、ニュース記事など</td></tr>
<tr><td rowspan="3">ソフトウエア</td><td>アプリケーション</td><td>MPGプレーヤ、ビューワ</td></tr>
<tr><td>ミドルウエア</td><td>Oracle DB, BPEL, Tomcat</td></tr>
<tr><td>OS</td><td>Windows, Linux, iOS, Android</td></tr>
<tr><td colspan="2">ハードウエア</td><td>サーバ、PC、スマフォ</td></tr>
</table>

図 5:システムの構成要素

なお日本では、総務省の統計局が「日本標準産業分類(平成9年11月改定)」において、情報通信業を以下のような分類としています。

大分類	中分類	解説
G:情報通信業	37:通信業	いわゆるキャリアで、固定通信と移動通信に分類される。さらに固定(電話)は地域と長距離、有線放送に分類され、電話だけでなくネットワークも含まれる。
	38:放送業	公共放送、民間放送、有線放送に分類され、民間放送は更にTV、ラジオ、衛星に。有線はTVとラジオに分類される。
	39:情報サービス業	ソフトウエア業、情報処理・提供サービスに分類。ソフトウエア業はさらに受託開発、組込み、パッケージ、ゲームに分けられる。
	40:インターネット付随サービス業	主にポータルサイト・サーバー運営、アプリケーション・サービス・コンテンツプロバイダ、インターネット利用サポートの3種類がある。
	41:映像・音声・文字情報制作業	制作と配給を行い、映画・ビデオ、TV番組、アニメ、レコード、新聞、出版、広告と細分化されている。

システムインテグレータ（SI）

システムインテグレータは日本固有の業種で、ソフトウエアの委託開発・保守、システム運用、関連するハードウエア等の調達などを一手に引き受ける業態を指します。IT 産業黎明期は、OS やミドルウェア(データベース、アプリケーションサーバなど)、開発言語はハードウェアベンダーが提供し、それらの上に IT 部門や依頼された受託開発ソフトウエア業者がオーダーメードのシステムを構築していました。受託開発ソフトウェア会社は規模が大きくなると、システムの一部分ではなく全体を一括請負するようになり、さらに必要なハードウェアやソフトウェア製品も調達して提供するようになります。これが一括請負でシステム開発を行うシステムインテグレータまたはソリューション・プロバイダーと呼ばれる業態が誕生した経緯です。平成 23 年 3 月 31 日付で廃止になりましたが経済産業省登録システムインテグレータ制度があり、219 社[3]の登録がありました。

SI 企業はその生い立ちから、コンピュータハードウェアを製造していたメーカのソフトウェア開発部門またはそれが独立した「メーカ系」、金融や大手製造業といったコンピュータを駆使していた企業のIT部門が独立した「ユーザ系」、受託システム開発から規模が拡大した「独立系」の 3 種類に分類されます。

日本の SI は収益率が低く、プロジェクト運営や著作権等の管理も甘いという指摘があり課題が多い業種であると言われています。また実際に多くのプログラマを擁する開発フェーズや手間のかかる保守フェーズでは、人件費を抑えるためにインドや中国といった新興国に委託する[4]事が多くなってきています。

表 2:主な国内 SI 企業

分類	例
メーカ系	日立ソリューションズ、東芝ソリューション、富士通エフサスなど
ユーザ系	野村総合研究所、新日鉄ソリューションズ、SCSKなど
独立系	NTT データ、IT ホールディングス、トランスコスモスなど

海外では昔から会計システムなどは既製品を使う事が多かったのですが、国内ではより効率的で、自社の流儀に合わせたシステムが求められ長くカスタムメイドが主流で SI が活発でした。特に 1980 年代、日本は金融インフラである「第 3 次オンライン」実現しますが、初期投資が莫大だったこと、長らく見直しをしなかったこと、当時のリーダが次々と退職しノウハウが薄れるなどの原因から、自らの手でシステムを刷新できない状況に陥りました。結局、みずほ銀行のような事件に発展しました。

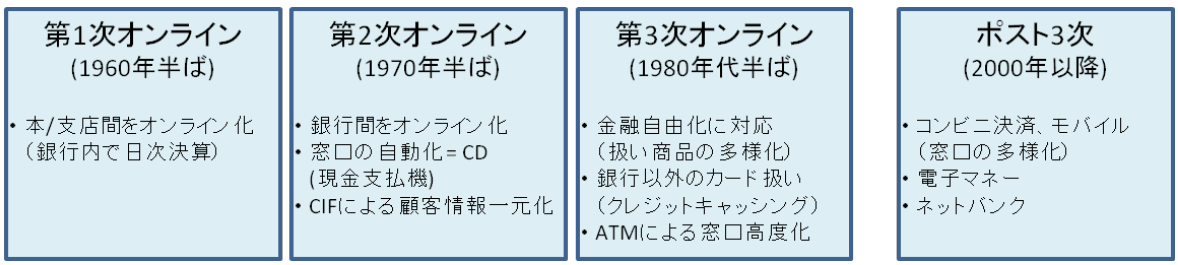


図 6:銀行システムの沿革

[3] 地域別登録 SI 内訳:北海道(4)、東北(3)、関東(146)、中部(16)、近畿(21)、中国(8)、四国(2)、九州(18)、沖縄(1)
[4] オフショア開発。最近ではさらにフィリピン、ベトナム、ロシア(およびウクライナ、ベラルーシ)も伸張しています。

大規模なシステムは構築時も大変な労力を使いますが、それを安定稼働し続けること(機能拡張、法規制変更への対応、調達品の老朽化・バージョンアップなど)も非常に手間がかかるという証明といえます。
また 2000 年問題への対応を転機として、古い手作りのシステムから新しいパッケージへの移行が盛んに行われました。経営に直結する会計システムの移行が顕著で、ERP(Enterprise Resource Planning)パッケージ[5]がビジネス用語として一般化する程です。

ソフトウェアパッケージベンダー

ソフトウェアパッケージ製品を提供するソフトウエアパッケージベンダーとしては、Microsoft(以下 MS)を筆頭に、Oracle、SAP、CA などが続きます。またグローバルの ERP パッケージシェアは SAP、Oracle、Sungard が上位を占めています。国内では SI 企業が過去の資産(経験)からパッケージを作る場合がありますが、海外への展開は積極的に行われていません。エンドユーザが海外に展開する場合に、国産パッケージを英語や現地語に修正する程度にとどまっています。
グローバルでは、どの大手ベンダーも M&A を活発に展開し、1 社で顧客が必要とする全ての機能(製品)を提供することで、囲い込み(Vendor lock-in)を促進する狙いがあると考えられています。

表 3:大手ベンダーによる主な買収

親会社	買収された会社(ソフトウェア)
MS	Skype, Fast Search & Transfer, Danger, Navision, Farecast, Visio, Hotmail,
Oracle	PeopleSoft, Retek, i-flex, Siebel Systems, Hyperion Solutions, BEA Systems
SAP	Sybase, Business Objects, Software AG(ADABAS)
IBM	Infomix, Rational, Cognos, ILOG, SPSS, Sterling Commerce, FileNet, ISS

こうした大手ソフトウェアベンダーは、技術を独占する傾向が強く標準化がなかなか進みません。A 社のソフトと B 社のソフトを連携させる X 社のソフトがあると、その X 社をめぐって、A 社と B 社が買収合戦を行うといった事が起こります。
またパッケージの販売形態(ライセンス形態)も各社ばらばらで、利用する側は十分に吟味する必要があります。

表 4:主なライセンス形体

ライセンス形態	意味
成功報酬モデル	売上やコスト削減額に比例して費用を徴収。売上の x%といった契約。
登録ユーザ数	登録されたユーザの数に応じて。1 ユーザ幾らといった契約。
同時アクセス数	同時にアプリケーションを使用できるユーザの数に応じて。
サイト数	事業所や企業(法人)単位で。但し企業規模に応じ調整される場合が多い。
CPU 数	サーバの CPU 数に応じて。最近ではコア数に応じて。サーバの筐体数の課金もある。
トランザクション数	扱うデータの規模や処理数に応じて。
サブスクリプション	主に登録ユーザの月額契約。クラウドでよく用いられる。

多くのソフトウェアベンダーは初期費用の他に、年間保守費用(サポート費用)を徴収します。それによりバグの修正パッチ提供、バージョンアップ、質問への回答など提供しています。一般的には初期費用の 1/4～1/5 程度となります。またバージョンが古くなると、保守契約を継続できなくなる

[5]欧米では会計パッケージの浸透が M&A を加速した事例もあるほど、会計システムが経営に与える影響が大きくなっています。

事も一般的に行われています。最近ではOracle社のライセンス・保守料の引き上げや、MS社のWindows XP サポート終了などが話題になっています。

寡占とイノベーション

このようにIT 業界は標準化（寡占も含め）が進み、いくつかの基礎技術はベンダーに依存せず共通となりました。たとえばネットワークプロトコルや、プログラミング言語、ミドルウェアの標準化などがよい例です。これらを上手く使う事で、企業を超えた情報連携ができるようになりました（SOA: Service Oriented Architecture や Web Service と呼ばれる仕組み）。

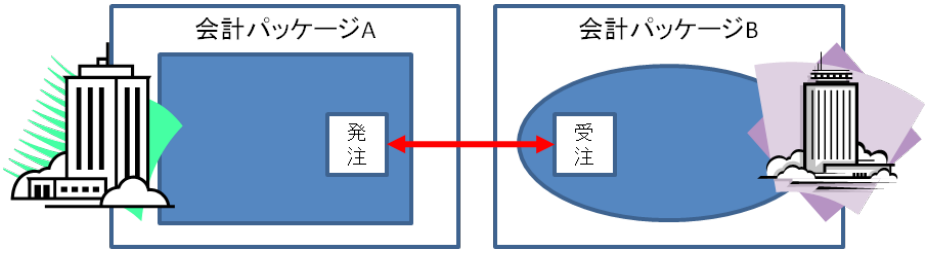


図 7：企業間連携概念図

しかし、その陰で新しい技術革新は「停滞しつつある」といえるかもしれません。例えばコンピュータの CPU は Intel が PC、サーバ分野とも 80%以上のシェアを持つに至りました。1990 年代には Intel 以外にも MIPS(R シリーズ), Sun(SPARC), DEC(VAX,Alpha), IBM(PowerPC, zSeries)などが競争していましたが、PC の普及と性能強化により Intel x86 および互換 CPU が市場を支配しています。

デスクトップOSでは、完全に MS の Windows シリーズの独壇場となっています。1990年代から最近まで市場はトップで現在でもデスクトップ市場では90％以上のシェアを持ちます。Windows 7からは 64bit への移行が始まり、Windows8 では新規投入される PC のほとんどが 64bit になっていますが、32bit 版ソフトウエアの互換機能によりシェアを落さずにいます。過去にはCPUの種類も多かったことから、CP/M、OS-9、DOS(MS-DOS,後の Windows)、OS/2 など多くの OS が存在しましたが、現在は Windows の独壇場です。

表 5：デスクトップOSシェア

OS	Ver.	シェア(%)	OS 別(%)
Windows	7 (NT6.1)	44.73	91.62
	XP (NT5.1)	38.73	
	Vista (NT6.0)	4.99	
	8 (NT6.2)	3.17	
Mac OS X	8 (Mountain Lion)	2.65	5.70
	6 (Snow Leopard)	1.87	
	7 (Lion)	1.18	
Other		2.05	2.05

NetApp. 2013/3

DBMS ではOSにより微妙に数値は変わりますが、Oracle, IBM DB2, MS SQL Server が御三家と称されています。

寡占が進むと、新しい技術は停滞する事があります。例えば Intel は従来の製品と全くかけ離れた

Itanium(64bit プロセッサ)を作りましたが、既存市場に浸透させる事はできず撤退が噂されています。最近の低消費電力分野(モバイル)では、ARM が事実上の標準となり、同社の Atom は苦戦しています。さらにサーバ市場では高機能よりも、低電力CPUを多く搭載するソリューションが登場し、同社の大きな課題となっています。
コストをかけて他の技術を開発するよりも、今の市場を保持しつつ互換を持たせることが先行者利益を守る事になります。しかし全く違う市場が立ち上がると、苦戦する事もあるという例です。

寡占により築いた市場がイノベーションの足かせとなるだけでなく、技術を公開しない事も業界全体の発展を阻害することがあります。いわゆるベンダーによる「囲い込み」です。
囲い込みの戦略の有名な例として「ハロウィーン文書」と呼ばれる MS 社の内部資料から明らかになった 3E 戦略(Embrace, extend, and exterminate:取り込み、拡張し、根絶する)があります。

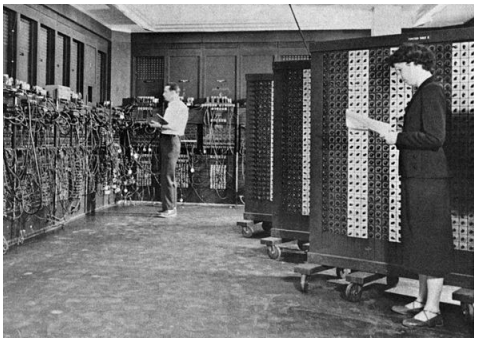
1. 取り込み段階:
   有望な市場に新規参入するため、既存のコミュニティや技術を分析し互換ソフトを提供し信頼を得る。
2. 拡張段階:
   既存技術の上位互換がある自社独自技術を追加する。旧来の製品に戻れないように工夫する。
3. 根絶段階:
   市場を押さえ自らが標準となる。標準外の製品や会社は買収するか、控訴その他の方法で市場から排除する。

Apple と Samsung 間で生じている特許紛争が話題になりますが、まさに(2)の段階で主導権を争っている状況なのかもしれません。

こうしたソフトウェア業界で、ライセンス形態の考え方や寡占・ベンダーロックインへの対抗勢力としてオープンソースが注目されています。オープンソースはソフトウェアの設計図ともいうべきソースコード(ソースプログラム)を公開したソフトウェア製品を指します。ソースを公開することで、カスタマイズや修正が可能となります。また独占するメーカに対向する勢力がコミュニティに参加し、知恵を出し合って新しいソリューションを作りだしています。

オープンソースの歴史

オープンソースはその名の通り、ソフトウェアのソースコードを公開した上で配布するソフトウェアを指します。無償である必要はないのですが、多くのソフトが無償で提供されています。
ソースコードは製造物の設計図に相当し、コンピュータ業界（とりわけソフトウェア産業）では工場や製造機器・治具といった資産がなくても、簡単にオリジナルそっくりのソフトウェア製品が作れてしまうため一般的には公開しないのが原則でした。しかし今日ではオープンソースと呼ばれるソフトウェア製品は数多く存在し、安価・高性能であることから注目を浴びています。なぜオープンソースが誕生したのかを歴史を紐解くところから解説してゆきます。



特許と著作権

コンピュータはその誕生から特許や著作権で保護されており、当時としては特異な手法で取得したと考えられます。最初に特許[6]を取得したコンピュータは 1944 年にエッカートとモークリーが制作した ENIAC だとされており、1973 年にミネソタ地裁で先行技術の公用であったとして無効判決が出た後も最初の IT 特許として記録されています。その後、コンピュータメーカが乱立しいろいろな特許や著作権を行使しています。
ここで一旦、コンピュータソフトウェアにおける特許と著作権を整理してみましょう。
ソフトウェアにおける特許とはアイデアそのものであり、実装方法が何であれ排他的に保護さます。分かりやすく言えば、アルゴリズムが特許という事になります。有名な例では Unisys 社(開発当時は合併される前のスペリー社・UNIVAC)が 1988 年に開発した圧縮アルゴリズム LZW 法があります。この圧縮アルゴリズムはイメージファイルの GIF(Graphics Interchange Format)や、古い UNIX に搭載されていた圧縮コマンド compress に搭載されていました。当初利用料を徴収しなかったため、データ容量が削減でき通信費用コストを抑えられる GIF は急速に広がっていきましたが、Web が一般化すると突如 1994 年に課金する事を表明しました。実際の特許は 2003 年(日本では 2004 年)に失効しましたが、その時に開発された PNG(Portable Network Graphics)や、ZIP が一般化する契機ともなりました。
この事件は、特許はアイデアであり、個々のプログラムではない点がポイントです。LZW 法を使っている限り、C や Java といったプログラミングが異なる実装であっても特許の使用許諾が必要になります。また生成ツールだけでなく、圧縮した結果の GIF ファイルまで課金する可能性もありました。ソフトウェア分野で特許を取得できれば、その権限は絶大だという事です。
つぎに著作権ですが、これは著作物の使用許諾です。つまりプログラムその物に対し効力が発揮されます。先の例と違い、C のプログラムと Java のプログラムでは原則として別の著作物であるという扱いになります。単純に言語を置き換える(Re-write)翻訳したようなプログラムの場合の扱いはグレーですが、基本的にはプログラムを保護する権利となります。著作権で保護されていると、無断での複製や改変・再配布が禁止されます。
IT 業界ではプログラムを複数のエンジニアが分担して開発するため、大抵の場合著作権はその開

[6] 米国特許 3,120,606、出願 1947 年 6 月 26 日、特許 1964 年 2 月 4 日。写真は U.S.Army (Public Domain)

発を行った会社に帰属します。日本のIT業界で問題なのは、この著作権をユーザに帰属させる事が慣習となっている点です。

例)

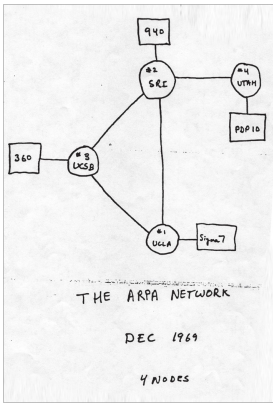
第〇条(著作権の帰属) 本契約において開発されたプログラムの著作権は、甲(ユーザ)に帰属する。

この様な条項があると、プログラム開発で身につけたノウハウを他のプロジェクトへ展開する事が難しくなります。また判例に照らし合わせると、実際に開発しているのはシステム会社であり、原始的著作権(実際に書いた人の権利)はシステム会社に帰属します。そのため本来であれば、開発費用以外に著作権の譲渡費用が発生するはずです。また下請けに委託する場合は、どこに原始的著作権が発生するのか非常に複雑になり専門家の中でも意見が分かれています。

パッケージソフトウェアではソフトウェアベンダーに著作権が帰属し、利用者は使用許諾契約によって利用が認められるようになっています。

では、せっかく保護(独占し利益を守る)されているプログラムのソースコードをわざわざ公開するのは何ためでしょうか。ソースコード公開の歴史を紐解くと、ソースコードが公開されたプロジェクトは意外と古く、インターネットが発端だという説が有力です。

ARPAによるインターネット



[7]インターネットは1969年に完成したARPANETが始まりとなります。ARPANETは本来米国防総省がスポンサーとなり最新技術を研究する機関ARPA(Advanced Research Program Agency: 高等研究局、後にDARPA)の1プロジェクトとして1966年に開始されました。

もともと米国防省のプロジェクトでしたが、利便性から他の研究グループが相互乗り入れを行うようになりました。1980年にはこれらネットワークが再編され、軍事目的ネットワークはMIL-NETとして分離独立する事になります。

ARPANETの管理は米国政府から委託を受け南カリフォルニア大学情報科学研究所(ISI)が運営していました。1990年代クリントン政権下の情報ハイウェイ構想を受け、運営母体が民間の非営利団体ICANN(The Internet Corporation for Assigned Names and Numbers)へ移管します。また技術標準の策定はIETF(The Internet Engineering Task Force)が行っています。

ICANN、IETFとも原則として誰でも参加でき、成果物に関する著作権を放棄(Public Domain公有)しています。このためインターネットに関する技術を参照する上で費用やライセンスといった制約は発生しません。これがインターネット普及の原動力となった理由の一つという事ができます。

アメリカ政府はインターネットを民営化するにあたって、多くの関係者と調整を行い検討した結果、以下の原則を定めました。

- “Private, Bottom-Up Coordination”(民間主導、下意上達的調整)
- “Rough consensus and running code”(大方の賛同と実際に動いている仕組みの尊重)

[7] 図は初めて相互接続に成功した時のARPANETの構成。ユタ大学(DEC PDP-10)、スタンフォード研究所(SDS 940)、カリフォルニア大サンディエゴ校(IBM System 360/75)、同ロサンゼルス校(Honywell DDP 516)。http://www.darpa.mil

- “Open and transparent”（公開と透明性）

参考）

上記原則を策定する際に公開されたインターネットに関する米政府の見解

グリーンペーパー http://www.nic.ad.jp/ja/translation/icann/bunsho-green.html

ホワイトペーパ http://www.nic.ad.jp/ja/translation/icann/bunsho-white.html

AT&T による UNIX

インターネットと時期を同じくして、ソースコードが公開されたのが UNIX です。UNIX は当初から無償配布と普及を目指したのではなく、実際にはビジネス上の制約から偶然生じた成果という事ができます。

そもそも UNIX は 1964 年に始まった多機能 OS 開発プロジェクト(MULTICS)が頓挫し 1969 年に AT&T が撤退した時がきっかけとなります。MULTICS プロジェクトは MIT が音頭をとり、AT&T の Bell 研究所、GE が参加する形で始まりましたが 1970 年は AT&T、GE とも撤退しています。この辺りの詳しい背景は Linux 関連の講座に譲るとして、ポイントは当時 AT&T が厳しい独占禁止法の監視下にあったという事です。

AT&T は電話を発明したグラハム・ベルが興した「ベル電話会社」を母体とし、1900 年代初頭には電話に関するあらゆる分野を独占した企業となっていました。研究開発、電話機製造、地域通信網、長距離通信網など全てを 1 社で賄っており、事実上の新規参入者は存在しませんでした。1970 年には独占を容認していた「キングスバリー協定: Kingsbury Commitment」が見直され 1984 年には地域電話会社 8 社、長距離電話会社、研究所の 10 組織に分離解体されました。それまでの間、AT&T は電話以外の事業への参入が禁止されていました。

UNIX は AT&T Bell 研の Ken Thompson が趣味の延長線上で開発し、他のメンバーが面白がって機能を拡充していったという流れがあります。研究成果として公にったのは 1974 年の Ver. 5 ですが、当時の AT&T は異業種への参入を禁止されていたため、UNIX は実費（少々の手数料とメディア代、送料のみ）での配布となりました。また Ken Thompson が母校カリフォルニア大学で客員教授として OS 理論を教えていた事も手伝い、政府機関、大学を始めとする研究機関へ一気に普及します。

ちょうどその頃、前出の ARPANET が始まり参加した研究機関の多くが、無償の UNIX を改良しネットワークへ接続していました。特に 1983 年公開の 4.2BSD UNIX はネットワーク機能(TCP/IP、ソケット)を搭載しており、インターネットを研究する機関の間で瞬く間に広がりました。

しかし、1990 年頃に入ると、独禁法が解除された AT&T がライセンスを主張[8]し、それを受け継いだ SCO 社が現在も多くの UNIX ベンダーを提訴する事になります。

UNIX の使用・改変が難しくなった 1990 年に、ヘルシンキ大学の Linus Benedict Torvalds が UNIX と同等の機能を有する OS を独力で開発し公開しました。これが Linux となりますが、詳細については多くの資料がありますので、そちらを参照してください。

フリーソフトの出現

UNIX は多くの研究機関、とりわけコンピュータ関係の研究機関・教育機関に広まったため、共通のプラットフォーム（OS と開発環境）として利用される機会が増えてきました。その中でも MIT の

[8] AT&T が UNIX のライセンスを主張したのは 1977 年の Ver.6 からですが、非常に緩やかな物でした。

Athena プロジェクト(1983～1991 年 MIT, DEC, IBM)は UNIX を使った分散処理環境の実現を目指したものでした。有名な成果物としては、X Windows System, Kerberos 認証、BIND/Hesiod ディレクトリサービスがあります。MIT は著作権を主張していますが、無償で提供されていました。他にも Sun 社が公開した NFS(Network File System, UNIX 間のファイル共有)、NIS(Network Information System, 旧 Yellow Pages, ディレクトリサービス)などが生み出されてきました。
このような無償で利用できるソフトウェアをフリーウェアと呼び、法人・個人が公開しています。ただし著作権はあるので、勝手な改変や再配布は行う事ができません。

表 6:無償ソフトウェアの種類

ソフトウェアの名称	概略
フリーウェア(Freeware)	無償で利用できるソフトウェア
PDS (Public Domain Soft)	著作権切れまたは著作権を放棄したソフトウェア
シェアウェア(Shareware)	試用については無償で利用できるソフトウェア
ドネーションウェア(Donationware)	利用者に寄付を求めるソフトウェア
アドウェア(Adware)	広告付きソフトウェア

このソフトウェアの定義に一石を投じたのが、Richard Matthew Stallman で、1985 年にフリーソフトウェア財団(Free Software Foundation, 以下 FSF)を設立し完全にフリーなソフトウェアを次々に発表しています。FSF の目的は「コンピュータで動作する全てのソフトウェアをフリーとする」事で、UNIX 互換 OS(GNU Hurd)の構築を目指し GNU(GNU's Not UNIX)プロジェクトをスタートさせました。
GNU が定義するフリーソフトは、プログラムのコピーを持つ人に対し以下を許可しています(詳しくは GNU GPL : General Public License を参照のこと)

0. 目的を問わず、プログラムを実行する自由
1. プログラムを研究し必要に応じて修正を加える自由
2. 身近な人を助けるため入手したプログラムを再配布する自由
3. コミュニティ全体が恩恵を受けられるよう改良点を公衆に発表する自由

上記のうち、1,3 を実現するためにソースコードへのアクセス許可(利用者の要望があれば、速やかにソースコードを提供すること)が必要となります。
一般のソフトウェア使用許諾(以下、ライセンス)と最も違うのは、上記 4 つの自由を再配布・改変した利用者にも課す点にあります。これはコピーレフトと呼ばれ、再配布をする場合、自分が付け加えたり、改良した部分を含め全ソースコードへのアクセス許可を義務付けました。
FSF の定義する「フリー」は無償ではなく、プログラムの分析・改良・再配布を自由に行ってよいという意味です。有償であっても GPL を満たせば「フリーソフト」ということができます。例えば DEC 社の Open VMS という OS は有償ですが、要求があれば実費でソースコードを入手することができました。
少し宗教がかった、精神論的な表現になりますがフリーソフトは「真の自由を実現した」ソフトウェアと位置付けられています。「言論の自由」と同じ意味と考えると分かりやすいかもしれません。
このフリーソフトに対し、ソースコードへのアクセスを禁じたプログラムを「プロプライエタリ・ソフトウェア(proprietary software)」と呼びます。proprietary は「独占的な、専属的な」とう意味で、多くのソフトウェアベンダーが提供する製品は、このカテゴリーになります。
FSF の GNU プロジェクトの成果物は非常に機能・性能とも優秀で、一時は C コンパイラを内製せ

ずに、GNU-C(gcc)を採用するコンピュータメーカがあったほどです。しかし IT 業界、とりわけエンジニアの間で評価が高かったのですが、「自由」の定義についてしばしば(特に経営層)問題[9]を起こす事がありました。
現在でも FSF は、後述のオープンソースとはこの「自由」対する考え方が根本的に違うとし、創始の Stallman は否定的な立場をとっています。

参考)
FSF によるフリーウェアの解説
http://www.gnu.org/philosophy/categories.html

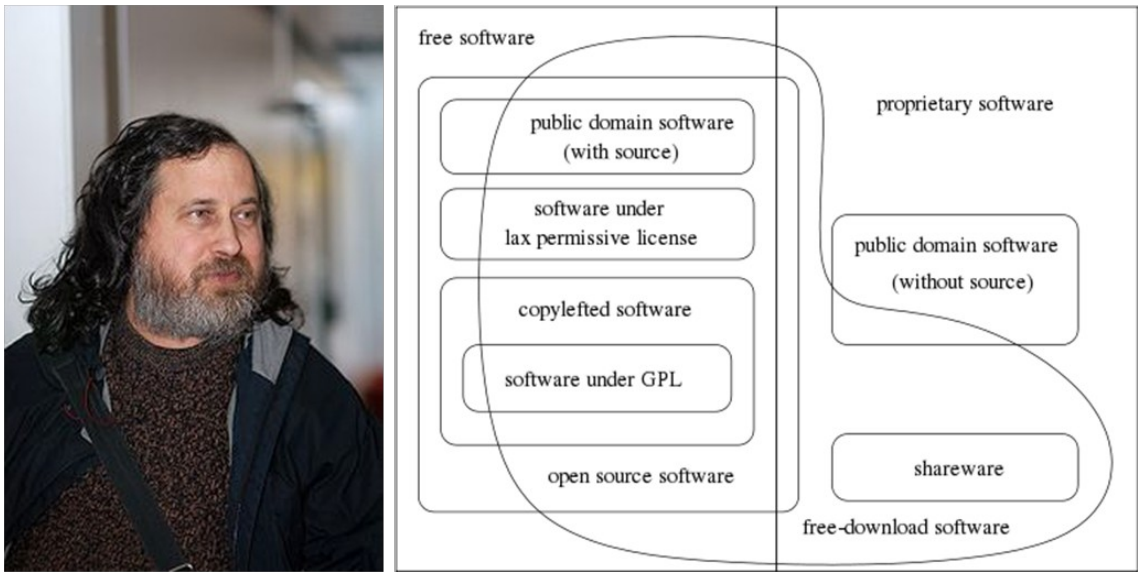


図 8:無償ソフトウェア相関図と Stallman

OSS は、全く著作権を行使しない PDS、再配布や改変などを制限をせず、逆に制限する事を禁止する Copyleft、何らかの制限を持たせた Permissive license すべてを含むことがわかります。

[9] Stallman 氏の個性による部分も多分にあったと思われます。特にプロプライエタリソフトやその企業に対し攻撃的な態度をとる事で有名です。

オープンソースの誕生

1995 年にインターネットが公開されると、メールや WWW が人気を博しました。特に WWW は多くの人に受け入れられました。WWW 自体は CERN の Tim Berners-Lee が NeXT 上で構築したのが始まりでしたが、NCSA(Center for Supercomputing Applications,イリノイ大)が UNIX 上に Mosaic ブラウザを実装した事で爆発的に普及しました。

Mosaic のプロジェクトメンバーは SGI のオーナであった James H. Clark の支援を受け、Netscape 社を立ち上げました。

その直後、Netscape 社と MS 社との間で壮絶なブラウザ戦争（シェア争い）が勃発します。MS 社は Windows OS に Internet Explorer を無償添付し、Netscape 社は自社のブラウザを「私用目的は無償」で提供する戦略をとっていました。WWW の進化とともにブラウザは多機能化していきますが、そこで MS 社の 3E 戦略が実践されます。

Netscape 社は MS への対抗処置として、ブラウザのソース公開を決断しますが、GNU GPL の考え方は若干思想的な要素を含んでおり、そのままではビジネスでの使用に躊躇する可能性がありました。

1998 年に Netscape 社は独自の定義でソースを公開する事にしました、その時の定義が「オープンソース」で、以下の内容となりました。

1. 自由な再頒布ができること
2. ソースコードを入手できること
3. 派生物が存在でき、派生物に同じライセンスを適用できること
4. 差分情報の配布を認める場合には、同一性の保持を要求してもかまわない
5. 個人やグループを差別しないこと
6. 適用領域に基づいた差別をしないこと
7. 再配布において追加ライセンスを必要としないこと
8. 特定製品に依存しないこと
9. 同じ媒体で配布される他のソフトウェアを制限しないこと
10. 技術的な中立を保っていること

これらはオープンソース 10 カ条(OSD: The Open Source Definition)と呼ばれ、現在 Open Source Initiative が管理しています。

また同じころ UNIX 上のフリーソフトについて Eric Steven Raymond が自らの体験を基にした「伽藍（がらん）とバザール」[10]と題する論文が発表されますが、それが Netscape 社をソース公開に踏み切らせたと言われています。

伽藍とバザール

オープンソースは単にソースコードを公開するだけでなく、多くのエンジニアが改良を常に加えるという特徴があります。この改良を行う、システム開発の手法に先の論文で名付けられた伽藍方式とバザール方式があります。

[10] http://cruel.org/freeware/cathedral.html

図 9:実際の伽藍とバザール

優秀な少数精鋭のプログラマが集中してシステムを構築する方法を、神社仏閣を宮大工が静かに荘厳に作る伽藍にたとえたのが「伽藍方式」です。伽藍方式ではプログラムを改変できるメンバーを限定し、有る程度の形になるまで公開しません。従来のシステム開発はほとんどこの方式をとっていて、公開する試作段階のプログラムはベータ版と呼ぶ事があります。

CDC から独立した Cray もこの伽藍方式によるスーパー・コンピュータ開発を行ってきましたし、GNU プロジェクトも Linux が登場するまで、伽藍方式を採用していました。

Linux では「バザール方式」を用いています。いろんなエンジニアがそれぞれ自分が興味ある分野・部分に改良、追加を行う手法で、いわゆるバザール(市場)のように見える事からその名が付けられました。Linux の場合はそもそもプロジェクトとして計画的に開発を進めたわけでなく、自然発生的に興味のある人々が集まったコミュニティで、プログラムを作成、テストの実施、ドキュメントの作成と各国言語への翻訳といった参加者の貢献によって機能拡張・性能向上が進んでゆきました。あまりにも無計画・無秩序だったため当初は失敗すると言われていましたが、Linux の成功をみると「バザール方式」でも十分な効果が得られる事が分かりました。

Raymond によると、バザール方式には以下の特徴があります。

- 参加者(開発者、利用者)を限定しない
- 参加者の独自性を尊重する
- 階層的な組織構造を持たない
- 開発過程を公開する(早めのリリース、しょっちゅうリリース、未完成でも公開)

特に開発過程を公開することで、開発者全員が問題点を共有でき、早い段階でバグを発見、対応できるため全体としての開発スピードと品質の向上がもたらされます。

特に品質について、利用者を巻き込む事が重要で、Linus はこれを「目玉の数さえ十分あれば、どんなバグも深刻ではない」と言っています。

Linux の成功から多くのオープンソース・プロジェクトが、この「バザール方式」をとっています。

現在の Linux(RedHat Enterprise Linux)は、ソースコードが約 5 億行で、普通であれば 30 万人の開発者が必要といわれています。(実際の RedHat の開発者は 5,000 人)

カーネルの開発には 1,000 人以上のエンジニアが参加していますが、その 75%は IT 業界からの参加で、個人のボランティアは 18%ほどです。

主なオープンソース・プロジェクト

オープンソースに従って開発されているプロジェクトの代表的な物を列挙すると、以下の様になります。

- GNU (GPL)
  本来は UNIX 同等の OS を作る事を目的として開始され、開発環境や各種コマンド、サーバ類を提供しているが 25 年たった現在もまだカーネル(GNU Hurd)を提供できていない。カーネル部分に Linux を採用した Debian GNU/Linux を提供している。
- Firefox / Thunderbird (MPL/GPL/LGPL)
  オープンソースを採用した初のプロジェクト Netscape から派生した Web ブラウザのプロジェクト。Mozilla Foundation により運営されている。また電子メールクライアント Thunderbird も提供している。
- Apache Web Server (Apache)
  世界で最も利用されている Web サーバ。LAMP(XAMP)の A。Apache Foundation が運営している。Apache Foundation は他にもデータベースやアプリケーション間連携など 100 を超えるプロジェクトを運営している。最近では Oracle から OpenOffice が譲渡された。
- MySQL (GPL/Commerical)
  RDBMS で、LAMP の M。最も利用されているオープンソース・データベースと言われる(mixi、Yahoo!、Facebook、Twitter などが利用)。Sun が運営していたが同社の買収に伴い Oracle 傘下となっている。
- PHP: Hypertext Preprocessor (PHP)
  LAMP の P。Web アプリケーション用プログラミング言語、サーバサイドで実行される。The PHP Group が運営している。PEAR とよばれるライブラリ公開・共有システムがあり、プログラミングの省力化に効果がある。
- Perl (GPL/Artistic)
  LAMP のもう 1 つの P。Larry Wall による汎用プログラミング言語、本来はインタプリタだがコンパイラも存在する。The Perl Foundation が運営し、CPAN によるライブラリ提供を行っている。
- Ruby (BSD/Ruby)
  日本人「まつもとゆきひろ」による汎用スクリプト言語。純粋なオブジェクト指向言語で 2011 年 JIS 規格が公開されている。合同会社 Ruby アソシエーションが法人への普及活動を行っている。GPL と Ruby 独自ライセンスを選択できる。
- OpenOffice (Apache)
  MS Office と同様のオフィススウィート。Sun が運営していたが 2011 年 6 月 Apache Foundation へ寄贈され、現在は Apache 傘下である。移管されるまでの間、紆余曲折があり Libre Office という別プロジェクトが分岐している。
- Word Press (GPL)
  PHP で開発された Blog ツール。CMS(Contents Management System)としても利用可能。

この他、SourceForge に登録されているプロジェクトは個人を含め、26 万以上が登録されている。

多様なライセンス

オープンソースではライセンスも多種多様になっており、利用する場合注意が必要です。
多くのプロジェクトでは GNU GPL を採用する場合が多いですが、プログラム言語は Artistic、Web では Apache などのライセンスが用いられています。
OSI が認定したライセンスだけでも 70 を超えていますが、主なライセンスと概要を以下にまとめました。

- GNU GPL：General Public License
  Stallman が GNU 成果物を配布するために初めてに定義した利用許諾で、オープンソースで最も利用されている。「GNU 一般公衆利用許諾書」と訳され現在は第 3 版が公開されている。特徴としては再配布時にも GPL に準拠するよう求めた「同一ライセンスの提供」がある。また用途によりアレンジされた GFDL(GNU Free Documentation License)や LGPL(Lesser ～)がある。
- X11 License(*)
  通称 MIT License と呼ばれ、X11 配布を目的に MIT が制定したもの。非常に簡単な内容で「無保証」と「著作表示の保持」を持つ。
- BSD License(*)
  カリフォルニア大学バークレー校が制定したライセンス。宣伝条項（二次的著作物の広告には、オリジナルの著作者に紹介すること）が特徴的である。この条項のない「修正版 BSD ライセンス」もある。
- Apache Software License(*)
  BSD ライセンスを元に Apache Foundation が制定したライセンスで、プロプライエタリ・ソフトウェアやクローズド・ソースの開発にも使える点が他のライセンスと大きく異なる。
- Artistic License
  Perl で採用されたライセンスで、原版名条項（再配布時に原版と同じ名称を使ってはいけない）を持つ。

*) ソースコード開示義務がないもの。

ライセンスに従わない場合、原則としてそのソフトウェアは利用できません。FSF は GNU の開発がひと段落したため、主にライセンス管理を業務としています。
数年にわたり FSF の法的代理人 SFLC(Software Freedom Law Center)は Cisco Systems 社の Linksys 製品の一部がライセンス違反を指摘していたのにも関わらず改善がなかったとして、2008 年に同社を提訴「フリーソフトウェア財団対シスコシステムズ事件」に発展します。翌年の判決で Cisco は和解金の支払いと、ライセンス準拠を保証する監査人の設置を義務付けられました。
このように OSS を活用する上ではリスクもある点を理解しておく必要があります。

オープンソースのメリット

FSF 設立時に Stallman が無償のソフトウェアでどうやって生計を立てるかについて質問を受けた事があります。彼は「ソフトウェア」は人類の財産であり、一企業や団体に独占されるべきではないという信念に立って、ソフトウェア作成は純粋な創造活動で、多くの人に利用される事が名誉だとし、実際にはソフトウェアの使い方や、改良に関するコンサルテーションや教育といった分野で儲ける事が出来ると語っていました。

現在のオープンソース・プロジェクトは多くの IT 企業から投資をうけたり、実際に IBM や Sun のように直接オープンソース・プロジェクトを運営する企業もあります。そういった企業の視点からオープンソースに投資するメリットには以下の様なものがあります。

世界規模の開発環境

まず第一に Linus のいう「沢山の目玉」理論があります。オープンソースはインターネットを通じ公開することで、興味のあるユーザや開発者が世界中から集める事ができます。優秀な開発者を数多く集める事ができ、世界中の言語に対応したソフトウェアやドキュメントを作成する事が可能となります。またテストもあらゆる視点で行われる事になるため、完成度も自然と高いものになります。

ソフトウェアを多くの言語や国に対応させることを I18N(Internationalization、最初の I と最後の N の間に 18 文字あることから)と言いますが、一企業でこれを行うには膨大な手間がかかります。単純にメニュー項目を翻訳するだけではなく、通貨(Currency)の変更、日付の表記が異なるといった修正も必要です。例えば日本では年月日曜日の順に表示しますが、日月年のような順序であったり、月を数字ではなく Jan, Feb といった略号にしたりする必要があります。また数字の表記でも、日本や米国で小数点(.)を使うところを欧州ではカンマ(,)使う場合があります。数値なのか区切り記号なのか、意味が変わってしまいプログラムの設計自体に影響する事もあります。

文字も左から右へ書くだけでなく、ペルシャ語・ヘブライ語のように右から左へ書くもの、日本語・中国語のように上から下へ書く言語もあります。

こういった複雑な問題も各国の開発者が自身のノウハウをと伴に協力する事で素早く解決できるでしょう。

ソフトウェア開発で最も手間がかかり、大変なのはテストです。作った本人は構築しているうちに、何らかの暗黙の規則を作ってしまい「そんな非常識な事はないだろう」と高を括りがちです。昔 ATM の開発で、全てのボタンを押す事はないだろうと開発者が思っていたところへ、実際の主婦の方にお願いしてテストをしてみたところ、いきなり買い物袋を ATM の上において財布を取り出し「全てのボタン押下」＝リセットしたという笑えない話がありました。

こうした設計に携わらない人によるテストは意外と重要で、オープンソースでは十分なテストを行う土壌があります。

スピードのある展開

世界中の開発者が参加しているため、24 時間 365 日ノンストップで開発を続ける事が可能です。誰かがバグを見つけると、そのパッチをまた誰かが作る。Linux のカーネル開発では、1 時間に 10 個近いパッチが摘要されると言われています。このスピードはバグ対応だけでなく、新しい話題・技術を取り込む事にも波及しています。グローバル企業ではよく 3 極体制といい、北米、日本（アジア）、欧州に拠点をおけば、通常の就業時間帯で（8 時間ずつ、残業なしで）24 時間対応可能という体制を敷く場合がありますが、オープンソースではそれを限界まで突き詰めた体制といえるでしょ

う。
また開発だけではなく、無償提供による利用促進・普及の速度も圧倒的に速いというメリットもあります。Java や Linux の開発に参加しておくことでノウハウを蓄積し、SI で収益を上げるといった企業も存在します。

コミュニティの存在

インターネットを通じ誰もが開発や評価に参加できますが、そういった参加者の集まりをコミュニティと呼んでいます。どのオープンソース・プロジェクトでもコミュニティがあり、活発な意見交換がされています。インターネットの仕様である RFC はこのコミュニティの原型だといえます。オープンソースのコミュニティは、参加者を制限しません。世界中の老若男女、言語、宗教に関係なくインターネットにアクセスさえできれば誰でも平等に参加できます。
利用者のコミュニティを市場としてとらえると、企業にとって非常に魅力的なものになります。また自社の技術力向上、優秀な人材の獲得、競合他社による独占の回避、企業イメージの向上、利用者の要望を吸い上げることで市場調査にも活用できます。
このコミュニティこそがオープンソースの最大の特徴であり、メリットだといえます。

オープンソースの将来

冒頭で紹介したようにオープンソースは IT 分野で、現在最も注目されている仮想化とクラウドに活用されています。これはオープンソースの最新性とスピードが影響していると考えられます。今後もオープンソースの市場は拡大し進化していくと考えられます。オープンソースは新しいソフトウェアを生み出し、それを成長させ、世の中に浸透させています。

この特徴をプログラム以外へ応用する取り組みが始まっています。

クリエイティブ・コモンズ

プログラムだけでなく、文書や絵画、音楽、映像といった全ての著作物にオープンソースの仕組みを応用したものがクリエイティブ・コモンズ(CC:Creative Commons)です。

著作権保持(All right reserved)は全ての権利をもち、パブリック・ドメインは全てを放棄している現行の知的財産の考え方を見直し、それらの中間に位置する権限を定義しています。クリエイティブ・コモンズ・ライセンスにより以下の行為を許可・制限することができます。

表 7:クリエイティブ・コモンズの権利

権限	意味
BY 表示	作品を複製、頒布、展示、実演を行うにあたり、著作権者の表示を要求する。
NC 非営利	作品を複製、頒布、展示、実演を行うにあたり、非営利目的での利用に限定する。
ND 改変禁止	作品を複製、頒布、展示、実演を行うにあたり、いかなる改変も禁止する。
SA 継承	元になった作品のライセンスを継承させた上で頒布を認める。

これらを組み合わせ、例えば非営利目的限定で作者表示を要求する場合は「CC BY-NC」といった組み合わせになります。

Creative Commons Japan / CC BY

図 10:クリエイティブ・コモンズの権利分布

Wikipedia やオープンソースに関する解説書などからで、クリエイティブ・コモンズが利用されています。詳しくはホームページを参照してください。

http://creativecommons.jp/

ハードウェアの仕様公開

最近ではハードウェアの仕様公開も盛んになっています。コンピュータ業界で初めて行われたハードウェアの仕様公開は、実は IBM PC[11]が始まりだと言われています。これは「オープンアーキテクチャ方式」と呼ばれています。

実際には、1981 年当時 PC を開発するプロジェクトメンバーはたった 14 人で、予算が限られていたうえ、短期間での成果を求められてたため自社制作を最初から諦めていたという背景がありました。IBM Pc について、IBM が作ったのは BIOS 部分だけで OS を含めすべてを外部から調達しました。ちなみに、その時の OS 調達先が Microsoft 社で、その後急成長する事になります。後に自社で全てを作り直した PS/2 と呼ばれるコンピュータは OS/2 が搭載されれいましたが、すでに市場は Windows が主導権を握っていたという後日談もあります。

商用 UNIX で絶大な影響力をもっていた Sun は、自社が開発した CPU である SPARC の仕様をオープンソース化しました。今では SPARC インターナショナルが管理し、主に富士通が開発にあたっています。SPARC のライセンスは事務手続き費用 99.00$だけで、誰でも製造を行う事ができるとしています。2011 年に世界最速だったスーパーコンピュータ京にも搭載されています。

最近話題の EV(Electric Vehicle、電動輸送機器)では、基礎技術を研究している SIM-Drive 社が試作においてオープンソース手法を取り入れています。

「先行開発車事業第 3 号募集のお知らせ」からの抜粋

> 先行開発車事業の特徴
>
> 先行開発車事業の大きな特徴は、オープンソースの手法にあります。電気自動車の最終製品を生産販売するのではなく、インホイールモーター技術とコンポーネントビルトイン式フレーム技術を組み合わせた電気自動車技術を普及することが当社のミッションでもあるからです。これらの技術を核に、参加機関の皆様に実際に電気自動車開発を体験していただくことで、ここで得た技術や人的ネットワークを自由に持ち帰り、各社でさらに発展させていただくことにつながります。また、電気自動車産業に関心をお持ちの全ての業種・業界の機関様にご参加いただけるプログラムになっております。

http://www.sim-drive.com/

現在話題のクラウド・コンピューティングはコンピュータ・サービスを電気やガス、水道のように必要な時に必要な量を安定供給する社会基盤へと変革させています。先の EV もこうしたオープンソース運動の一環と位置付ける事ができます。

[11] 写真は IBM PC 5150 / PC-DOS 5.0 / CC BY-SA Boffy b

今後の展開

オープンソースの考え方はコンピュータソフトウェアから、全ての創作活動へ広がりはじめていいます。このような社会現象を OSM: Open Source Movement - オープンソース運動と呼んでいます。この運動の効果について、Peter Ferdinand Drucker の NPO 観が参考になります。Drucker は NPO には 2 つのメリットがあると考えました。

1. 人間を改革する機構
   NPO の恩恵を受けた人は、それに触発され意識を改革するというもので、例えば医療関連の NPO からサービスを受け、病気が改善した人は健康に気をつけるようになるし、健康的な生活をするようになる。

2. 市民性の創造
   NPO に対しボランティアや寄付をすることで、社会貢献しているという実感を得る事ができ、市民活動が活発になり社会がよりよくなると云う「善意の連鎖」が生じる。

まさにオープンソース活動そのものであるといえるでしょう。