

1

OSS

OSS の進化

基
般

—コミュニティ開発のもたらすもの—

OSS とは

OSS 誕生の歴史

OSS (オープンソースソフトウェア) とは何か、それがソフトウェア開発という文脈でどのような意義を持っているのか、IT 業界にどのようなインパクトを与えているのかなどを、歴史を振り返りつつ紹介する。

ソフトウェアのソースコードをオープンにしたり配布したりすることは、大学などを中心に 1960 年代から行われていた。これが OSS の源流である。しかし、1980 年に入り、ソフトウェアの商業化が進展すると、その扱いを制限する動きが出始める。このソフトウェアの状況を大きく変えた出来事の 1 つが Richard Stallman のフリーソフトウェア運動である。

MIT (マサチューセッツ工科大学) 人工知能研究所にいた Richard Stallman は、ソフトウェアの自由を制限する動きを懸念し、1983 年 GNU (GNU is not Unix) プロジェクトを立ち上げた。GNU は、Unix 互換 OS (オペレーティングシステム) およびツール群をフリーソフトウェアとして作成するプロジェクトである。GNU プロジェクトはその名の由来どおり、Unix ではない OS、すなわち、カーネル、コンパイラ、エディタ、シェルスクリプト、デバッグなどを開発している。残念ながらカーネルは未完成だが、ほかは今も Unix コミュニティの中で広く利用されている。そして、1985 年、ソフトウェアは自由に利用し、改変し、再配布することができるものという思想のもとに FSF (Free Software Foundation) を設立し、フリーソフトウェアを定義し、運動を開始した。その運動の成果の 1 つが、GNU

■ 吉岡弘隆(楽天(株))

プロジェクトのソフトウェアライセンスとして考案した GNU GPL (GNU General Public License) である。この GNU GPL は、1980 年代は GNU プロジェクトのツールに採用される程度だったが、1990 年代に入って Linux に採用されたことにより広く使われるようになった。現在では OSS の代表的なライセンスとなっている。この Linux は「Linux の誕生」の節で、ライセンスは「OSS ライセンス」の章で詳しく説明する。

もう 1 つ、OSS 誕生そのものに大きな影響を与えた出来事として、Netscape による Web ブラウザのソースコード公開がある。これがきっかけとなり OSS という言葉が生まれた。

OSS の誕生

1998 年、Netscape が自社の Web ブラウザ Netscape communicator のソースコードを公開した。

商用ソフトウェアのソースコードを無条件に公開するということは、当時では前代未聞のことであった。Netscape のこの決断は、当時 Microsoft とのブラウザ戦争に敗北しつつあった同社が仕掛けた起死回生の奇策であった。Linux のようにソースコードをインターネットで公開すれば、第三者がさまざまな機能追加を自的に行い、Eric Raymond の言うバザール型開発 (「バザール型開発」の章で解説) が始まることを期待したのである。

Netscape はソースコード公開にあたり、Mozilla プロジェクトを立ち上げた。そのプロジェクトの戦略会議において、ソースコードをどのように公開していくかという議論があり、「オープンソースソフトウェア (OSS)」という言葉が新たに提案され

条件	
1	自由な再頒布を認めること
2	ソースコードの利用の自由を認めること（「ソースコード公開」含む）
3	派生物の利用の自由を認めること
4	原著作者のソースコードの完全性を示すこと（ソースコードが容易に入手可能であることを保証しなければならない。もしくは一切変更されていない基本ソースコードとパッチという形で頒布することを義務づける）
5	特定人物・集団に対する差別を禁止すること（たとえば「特定国家への輸出を禁ずるソフトウェア」はOSDに合致しない）
6	使用分野（Fields of Endeavor）に対する差別を禁止すること（たとえば「兵器への利用を禁ずるソフトウェア」はOSDに合致しない）
7	ライセンスの権利配分を認めること（プログラムに付随する権利は、再頒布者にも等しく認められなければならない）
8	特定製品への限定適用を禁止すること
9	オープンソースライセンスのソフトウェアとともに頒布される他のソフトウェアを制限しないこと。（著作物として明確に別物と定義できるソフトウェアは、別のライセンスであってもよい。たとえばオープンソースライセンスと商用ライセンスを一緒に頒布してもよい）
10	技術中立であること（特定技術に依存するような条項があってはならない）

表-1 OSD：オープンソースライセンスが満たすべき条件の定義

た。これは、「フリーソフトウェア」という言葉が「無償のソフトウェア」という意味で捉えられ、ビジネスで用いるには、経営者や投資家あまりいい印象を持っていなかったためである。そこでそれらの人々にも受け入れやすい言葉としてOSSという言葉を考え定義を行った。Eric RaymondとBruce PerensはOSI（Open Source Initiative）という組織を立ち上げ、そのときから「フリー」という言葉ではなく、OSSという言葉を使い始めている。これがOSSの誕生である。

上記マーケティングのターゲットを従来のハッカーコミュニティではなく経営者や投資家などビジネスパーソンにしたこと、オープンソースの定義を定めたこと、OSSという用語を第三者が勝手に利用しないように商標登録したことなど、Eric RaymondらのOSSへの貢献は多大である。

なお、彼らが定義したOSD（Open Source Definition）を表-1に示す¹⁾。この定義から分かるように、ソースコードを公開するだけでは、OSSとは言えないのである。

分類	変更・再配布	ソースコード公開	対象ライセンス例
コピーレフト	できる	必要	GNU GPL
パーミッシブル	できる	不必要	MIT Apache BSD

表-2 OSSライセンスの分類

Linuxの登場

Netscape社のコード公開の手本となり、先に述べたように、GNU GPLの普及に大きく貢献したソフトウェアがLinuxである。GNUの思想を体現した最も有名なソフトウェアである。

LinuxはUnixと同じPOSIX互換のOSカーネルであり、当時フィンランドの大学生であったLinus TorvaldsによってPC向け（インテルの386アーキテクチャ用）に作られた。Linusは1991年にLinuxをGNU GPLでインターネットに公開し、世界中の開発者から送られてくる修正や新規機能（通常はパッチ形式）をどんどん取り入れて改良を行っていた。インターネットの普及に後押しされる形でLinuxの開発は急速に進み、適用が広がっていった。それに伴い、そのLinuxのライセンスであるGNU GPLも多くの人を知ることとなった。

Linuxはもう1つOSSに大きな影響を与えている。それはコミュニティを形成した開発方法である。OSS開発にはよく用いられている手法なので、詳細について「バザール型開発」の章で解説する。

OSSライセンス

コピーレフトとパーミッシブルライセンス

OSSのライセンスは、その制約の特徴により2種類に分類できる（表-2）。

1つはコピーレフト（Copyleft）と呼ばれるライセンスであり、著作権をコピーライト（Copyright）と呼ぶことをもじて命名された。代表的なものにGNU GPLがある。コピーレフトは派生物においてもコピーレフトを要求する。

もう1つは、コピーレフトほど強い制約を持た

ないパーミッシブルライセンス (permissible) である。このライセンスでは派生物のソースコード公開を要求しない。MIT ライセンス, BSD ライセンス, Apache ライセンスなどがそれにあたる。

通常, ソフトウェアは著作権法によって保護されるので, 明示的なライセンスでの許可がない限り, コピー, 変更, 再配布などはできない。しかし, OSS ライセンスはその許可など利用を行うときの条件が記されている。また, いずれのライセンスも無保証である。これらが OSS ライセンスの特徴と言える。

ポピュラーな OSS ライセンス

OSS のさまざまな情報を蓄積した世界最大級のデータベースを持つ BlackDuck 社が, 利用ライセンスを調査した結果を公表している²⁾。それによると, よく利用されている OSS ライセンスは, 表-2 でも挙げたライセンスであった。それぞれのライセンスの概要と利用率を表-3 に示す。

GNU GPL とパーミッシブルライセンスは半々で利用されているが, 最近の統計によると, GNU GPL のようなコピーレフトの利用が徐々に減って, パーミッシブルライセンスが増えてきている。

■ OSS ライセンスのコンプライアンス

OSS が広く利用されるにしたがい, OSS ライセンスの無理解によるライセンス条項の違反が散見されるようになった。

特に GNU GPL の場合, 派生物もソースコードの公開が求められるが, その認識がないまま GNU GPL の OSS を利用し, ソースコードの改変, バイナリの再配布を行った場合, ソースコードの公開をしていないと, 最悪, 著作権者から訴訟を起こされるなどのリスクが発生する。ライセンスの啓蒙と違反への対応のため, 違反利用の有無をチェックする gpl-violations.org や法的に対応する Software Freedom Law Center (SFLC) などの非営利団体も生まれた。彼らは違反を見つけるとその企業に対して警告を出し, それでも改善が行われない場合は事実を公開する。SFLC は法定代理人になって提訴を行う場合もある。

ランク	ライセンス名	概要 (特徴)	利用率
1	GNU General Public License (GPL) 2.0	ソフトウェアの実行, コピー, 変更, 再配布を許可する。二次著作物のライセンスは GNU GPL。ソースコードの公開を求める。	26%
2	MIT License	ソフトウェアの実行, コピー, 変更, 再配布を許可する。ライセンス文の添付を求めるがソースコード公開の義務はない。	19%
3	Apache License 2.0	ソフトウェアの実行, コピー, 変更, 再配布を許可する。ライセンス文の添付を求めるがソースコード公開の義務はない。仮に特許があったとしても行使しない (ライセンスフリーで利用できる)。	15%
4	GNU General Public License (GPL) 3.0	ソフトウェアの実行, 変更, 再配布を許可する。ソースコード公開の義務がある。仮に特許があったとしても行使しない (ライセンスフリーで利用できる)。	11%
5	BSD License 2.0 (3-clause, New or Revised) License	ソフトウェアの実行, コピー, 変更, 再配布を許可する。著作権の添付を求める。ソースコード公開の義務はない。	7%

表-3 OSS ライセンスの概要と普及率

ライセンスは, その内容をよく理解し正しく利用すれば恐れる必要はない。GNU GPL のソースコード公開にしても, その目的は 1 つのソフトウェアを複数の開発者によって迅速に開発し技術的なシナジーを生み出すことであり, その恩恵は大きい。ライセンス違反でそれらの恩恵を阻害しては意味がないので, よく理解して利用する必要がある。

なお, パーミッシブルライセンスの場合は, ソースコードの公開は義務づけられていないため, そのような訴訟のリスクやソースコード公開にまつわるコストは少ない。必要に応じて適切なライセンスの OSS を選択すればリスクを下げることも可能である。

バザール型開発

■ バザール型開発の特徴

OSS の開発方法について, その代表例である Linux カーネル開発を例にとって紹介する。

Linuxの開発は1991年から始まっているが、その開発方法が注目されたのは、Eric Raymondが1997年に「Cathedral and Bazaar（大聖堂とバザール）」を発表してからである³⁾。

従来、OSのような複雑なものは「大聖堂を建築するように中央集権的なアプローチで開発するもの」と考えられていたが、Linuxの開発方式はその対極にあった。開発者が限定されることはなく、独立した個人がルールや命令システムの少ない方法でオープンに開発し持ち寄ることから、市場に似ているとしてバザール型開発と名づけられた。開発した内容がメインラインに取り込まれるかどうかはコミュニティのコンセンサスによる。そのコンセンサスへの影響力はそれまでのその開発者自身の貢献に依存するというメカニズムになっている。リーダーが方向を決める方式ではないところも特徴の1つになっている。

このバザール型開発によりLinux開発は活発かつ継続的に行われてきた。Linux Foundationがまとめた「Linux Kernel Development」⁴⁾によると、Linuxカーネルバージョン3.10は前のバージョンから63日目でリリースされ、その間に13,367パッチの変更が加えられた。現在、カーネルソースコードは4万ファイルを超え、行数は1,700万行になろうとしている。

このような大規模ソフトウェアの開発は、複数の開発者によってインターネットを介して行われる。そのソースコード管理を支える技術として分散型バージョン管理システムを利用している点も、バザール型開発の特徴の1つである。Linuxで用いているgitは、2005年にLinus自身がLinux開発のために開発したものだが、今ではバザール型開発の標準的なツールになっている。ファイルの履歴情報などが格納されたデータベースであるリポジトリを分散して持つことができるため、個々の開発者が独自のブランチを自由に持ち、他の開発者の開発状況を気にすることなくさまざまな試行錯誤を加え改良を試することができる。これが開発の品質とスピードを向上させている。

バザール型開発の開発量と開発スピードは1社単独で行える規模をはるかに超えるものである。さらに、ベンダやユーザが協力して開発を進めるため、新製品適用や普及も早い。それをLinuxが証明して見せた。そして、このLinuxの成功に続くため、多くのOSSがバザール型開発を採用している。

ただし、バザール型開発は、開発にかかわる人や企業の力関係のバランスが絶妙に取れていてこそ威力を発揮する。OSSコミュニティの中には、コンスタントに数千名を超える開発者、数百社を超える企業の参加があるものも少なくない。彼らがバザール型開発を理解して参加しているからこそ、成功しているのである。企業もOSS開発への参加にはいろいろ工夫や努力をしている。これについては、次節でもう少し詳しく説明する。

なお、Linuxのエンタープライズ分野への適用については、本特集5.「基幹システムを実現するLinux技術」で解説しているので参照してほしい。

■ 企業の開発参加

1990年代は、OSS開発に参加する者のほとんどは所属企業とは関係なく活動していたが2000年になると変化が生じる。たとえば、1990年代は、ベンダ企業がLinux開発者を雇用することはまれで、開発参加はほとんどなかった。しかし、2000年に入り、基幹システムを開発している企業がLinuxを採用し始めると、企業からのコミュニティ開発への参加者数が増えていった。単に開発者を送り込むだけでなく、Linuxのエンタープライズ分野への適応を加速するために、CA、富士通、日立、HP、IBM、Intel、NECなどがOSDL（Open Source Development Labs）を発足、2007年にはFSG（Free Standards Group）と合流してLinux Foundationを設立し、企業のLinuxコミュニティへの参加・貢献の基盤を作ったのである。

このFoundationや非営利団体の存在も、OSSの成功には重要である。コミュニティ開発を運用・継続していくためには何者の影響も受けない活動資金を提供する仕組みが必要であり、Foundationのよ

うな中立の組織がこの役割を果たす。企業は、この組織運営を支援することで、コミュニティへの貢献を示すことができる。

この企業参加の成功の先駆者は Apache だった。1999 年に Apache Foundation を設立し、開発のインフラの提供、企業や個人がハードウェアや人材などを安心して寄付できる法的な基盤の整備、開発ボランティアの法的訴訟からの保護や対象 OSS のブランドの保護などを目的に活動を開始した。現在もその理念を継承し、いろいろな企業と有機的につながりながら数多くの OSS プロジェクトを支援している。

一方で、各企業がコミュニティ開発に参加するにはそれなりの意図があるし、また未経験のバザール型開発への参加には壁もあった。IBM Linux Technology Center を率いていた Daniel Frye が、講演で IBM における Linux コミュニティへのかかわりの歴史を述べているので事例として紹介する。

2000 年当時、IBM などのハードウェアベンダは OS として独自 Unix を開発し顧客システムに適用していたが、徐々にマーケットシェアを下げていた。そこで、その流れに対抗するための第三の OS として Linux に注目し、本格的に参入を始めた。

開発に関しては、従来は社内で検討してからコミュニティへ回答するというフローだったが、Linux の開発スピードに対応するため、すべてコミュニティのメーリングリストでオープンに議論することにした。社内メールは一切使わない。また、独自のプロジェクトを自分たちだけで立ち上げるのではなく、既存のプロジェクトに参加し、やりたいことを説明し、他の開発者やユーザの協力を願い出る。徹底したコミュニティ開発の文化への切り替えを行ったのである。従来の文化的な障壁もあり慣れるまで時間がかかったようだが、今では何の問題もなくコミュニティ開発を行っている。

IBM はこの経験を活かし、Linux 以外の OSS コミュニティへも参加し、上手につきあっている。企業が開発に参加するためには、最初にこれくらい徹底して始めた方がよいのかもしれない。

■ バザール型開発事例

前章で述べたように、最も成功した OSS プロジェクトの Linux は GNU GPL ライセンスを採用している。では、バザール型開発は、GNU GPL だから成功するのだろうか。それともそれ以外のライセンスでも可能なのか。その疑問に対し、HP の Eileen Evans は OSCON 2013 の講演で、パーミッシブルライセンス（MIT や Apache ライセンスなど）でも可能だと答えている⁵⁾。

その良い例が、最近急速に普及しているクラウドの OSS プラットフォーム OpenStack の開発だろう。OpenStack は Apache ライセンス、すなわちパーミッシブルライセンスである。その開発方法はバザール型開発であり、開発者、参加企業、開発量など、いずれも Linux に匹敵する勢いで増加し、普及も進んでいる。すなわち、派生物の公開が義務づけられていないパーミッシブルライセンスの OSS でも、バザール型開発は十分に効果を発揮することが証明されている。

筆者は、バザール型開発の本質は情報が公開されていることだと考える。だからこそ、企業が共同で開発をする場合、派生物においてもソースコードの公開を義務づける GNU GPL が好まれていたと考える。そのため、ソースコード公開の義務がないパーミッシブルライセンスの場合、最初は、各企業が十分な情報公開を行わず、結果として重複作業が多く発生し、イノベーションの速度が遅くなる可能性を危惧していた。しかし、OpenStack 開発の事例は、パーミッシブルライセンスにおいても情報は十分公開され、バザール型開発も活発なコミュニティ活動も可能であることを証明している。そのため、この開発方法は、今後もいろいろなライセンスや開発状況の中で進化しながら普及していくに違いない。

なお、OpenStack については、本特集 2 「OSS/OpenStack に見るクラウド基盤技術」も参照されたい。

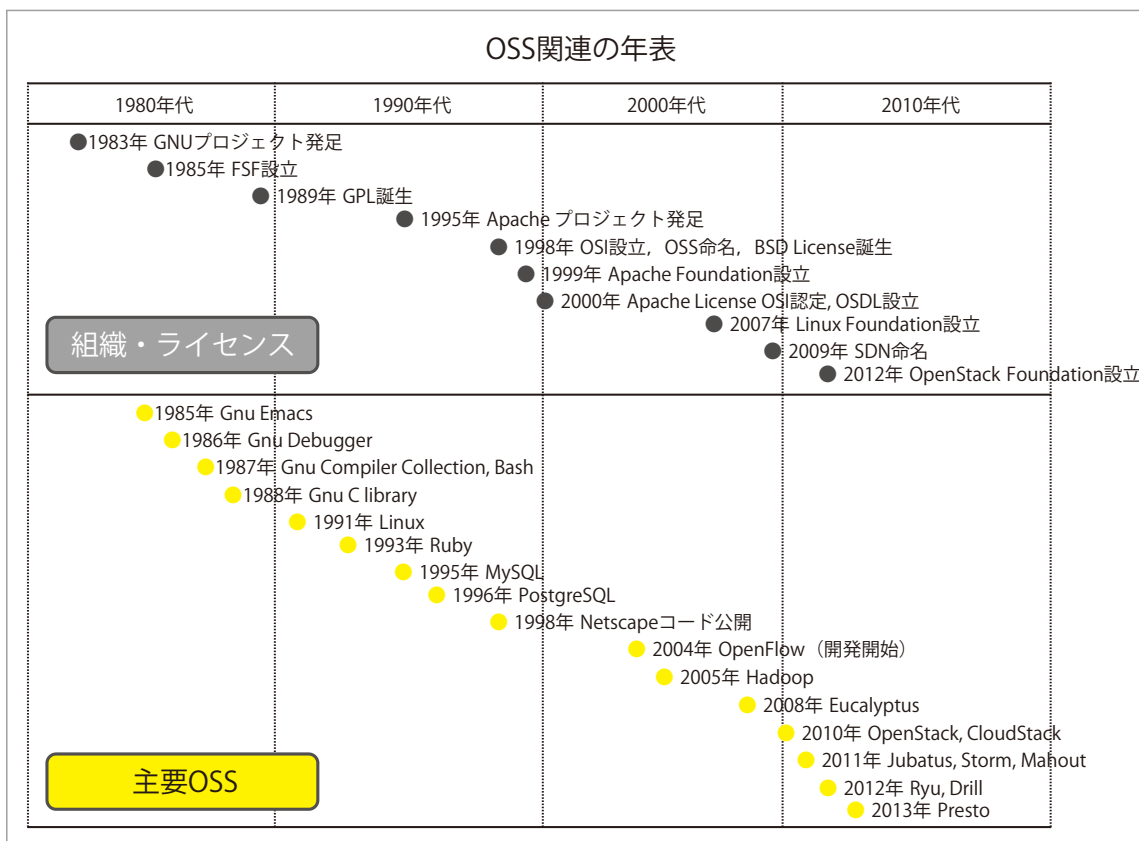


図 -1 OSS 関連の年表

OSS の利用状況と今後の動向

OSS の歴史を概観し、ライセンスや開発方法について紹介した。主要な OSS の出来事を年表にしたので参照いただきたい (図 -1)。OSS を理解する上での参考になれば幸いである。

最後に、OSS の利用状況について、筆者なりの分析を紹介する。企業や個人において OSS の利用状況を 4 種類のレベルに分けて分析したものである。レベル分けの指標は OSS 利用の成熟度である。

1. 発見 (find)

OSS を発見し利用する。本格的な導入というよりも試用のレベル。この段階で問題が見つかり本格的な導入には至らない。逆に、問題がなければ、ライセンスフィーがかからないので次のレベルに進む場合が多い。

2. 利用 (use)

試用してみて意外と便利なのが分かり、本格的に利用してみようというレベル。対象 OSS のユ

ーザが多いとインターネットには情報があふれているので、それを参考にもっと高度な使い方へと進む。ちょっとしたトラブルはインターネットで検索して解決する。

3. 参加 (participate)

本格的に利用を始めてしばらくするとさまざまな問題にぶつかり、開発コミュニティに参加するレベル。最初はメーリングリストなどで質問を行う程度だが、解決しない場合は、回避策を試したり、時にはパッチを書いて問題を解決する。その活動が進むとコミュニティでの存在感も徐々に増す。

4. 革新 (innovate)

自社のサービスに OSS を積極的に組み込み、それを競争力の源泉にするレベル。オープンイノベーションを実践するエンジンとして OSS を戦略的に活用する。コミュニティでの開発もリーダー的な存在となる。

IT 関連企業における OSS 利用レベルは、以下のような状況にあると考えている。

OSS 専業ベンダは、OSS を自社のサービスの中核に位置づけ競争力の源泉としている。つまりレベル 4 の段階にあり、OSS を自社の競争力にしている。これは今後も維持されていくだろう。

OSS 利用に先進的なハードウェアベンダはレベル 3 であり、一部ではレベル 4 の動きが出てきている。たとえば Linux や OpenStack の開発に積極的に参加しているベンダはレベル 4 に近い。

システムベンダ (SI ベンダ) は、多くはレベル 2 と言えるが、開発に参加しているベンダはレベル 3 ~ 4 になっている。

ユーザ企業はまだ発見して利用する段階、すなわちレベル 1 ~ 2 にある。しかし Web 系の企業で OSS を自社サービスのエンジンに利用するなど、徐々にその数は増えている。また、開発のしやすさから、ユーザ企業自身が必要な機能の開発や障害対応に取り組み始めており、レベル 3 になる日も近いかもしれない。

以上の分析はあくまで筆者の観測に基づくものであり、より定量的なサーベイが今後必要になる。

どの企業でも、OSS を採用することにより、ベンダロックインを回避できるだけでなく、いろいろな企業とパートナーシップを組みやすいため新たな価値を生み出せる可能性がある。また、ソースコードがあるおかげで技術力があれば自社で問題を解決でき、すばやく対応できる。さらには世界の優秀な工

エンジニアとコラボレーションをすることを通じて自社のエンジニアの育成をはかることもできる。これらはいずれも商用製品にはないメリットである。

本特集でも多くの OSS とそのメリットが紹介されている。その適用先や種類には限界がない。今後もいろいろな OSS が生まれ、技術を進化させていくだろう。OSS がどのような新しい世界を切り開くのか、楽しみである。

参考文献

- 1) The Open Source Definition, <http://opensource.org/osd>
- 2) Top 20 Open Source Licenses, <https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses>
- 3) Raymond, E.: The Cathedral and the Bazaar, <http://www.catb.org/~esr/writings/cathedral-bazaar/>
- 4) Linux Kernel Development (Sep. 2013), <http://www.linuxfoundation.org/publications/linux-foundation/who-writes-linux-2013>
- 5) Evans, E.: Licensing Models and Building an Open Source Community, OSCON 2013, <https://www.youtube.com/watch?v=o82QmitU4XE>

(2014 年 10 月 14 日受付)

■ 吉岡弘隆 hyoshiok@gmail.com

楽天 (株)、開発部技術理事。1984 年慶應義塾大学大学院修了、同年日本デジタルイクイップメント研究開発センタ入社、1994 年日本オラクル入社、2000 年ミラクル・リナックス社取締役 CTO。2009 年楽天入社、現職。