



慶應義塾大学ビジネス・スクール

「ネットスケープ モジラプロジェクト(A)」

5 “われわれは昨年、私が達成しようと思っていた目標を達成できなかった。モジラプロジェクトを、ネットスケープ社が数ある協力者の中の1つに過ぎないような、ネットワークによる共同プロジェクトにしなかった。さらに、エンドユーザー向けのソフトの出荷もしなかった。私にとって出荷は非常に重要だ。だから私はこのプロジェクトから手を引く”

10 1999年4月、ネットスケープコミュニケーションズ社の創業メンバーのひとりで、優秀な技術リーダーであったジェレミー・ザウインスキー(Jamie Zawinski)氏は自らが率いてきたモジラプロジェクト(Mozilla Project)の1年間の評価をこう語り、AOL・ネットスケープ社を退社する決意を表明した。1998年のプロジェクト開始当初、市場を制覇しつつあるマイクロソフト社のインターネットエクスプローラー巻き返しのためにネットスケープ社がう

15 ったモジラプロジェクトは、ソフトウェア会社にとって最も重要な知的財産である製品ソースコードを外部、それもボランティアな開発者コミュニティに公開するという、ブラウザ製品開発にとって起死回生の最終手段として業界中の注目を浴びていた。だが、実際にはプロジェクト開始後1年がたっても、成果物としてのソフトウェアが一向に出荷される見通しはたっていないかった。

20

WWW とネットスケープ社

インターネットに火をつけたふたつのキラーアプリケーションは、電子メールと

このケースは慶應義塾大学ビジネス・スクールの浅川和宏教授の指導の下、慶應義塾大学ビジネス・スクールMBA学生(M26)の佐藤恭平が公開資料に基づき作成したものである。当ケースはクラス討議の資料として作成され、経営管理上の適否を例示するものではない。

本ケースは慶應義塾大学ビジネス・スクールが出版するものであり、ケースの複製等についての問い合わせ先は慶應義塾大学ビジネス・スクール(〒223-8523 神奈川県横浜市港北区日吉本町2丁目1番1号、電話 045-564-2444、e-mail case@kbs.keio.ac.jp)。また、ケースの注文は <http://www.kbs.keio.ac.jp/case/index.html>。慶應義塾大学ビジネス・スクールの許可を得ずに、本ケースのいかなる部分の複製、検索システムへの取り込み、スプレッドシートでの利用、またはいかなる方法(電子的、機械的、写真複写、録音・録画、その他種類を問わない)による伝送は、これを禁ずる。

Copyright©2004 は浅川和宏、佐藤恭平が保有する。

WWW(World Wide Web)、このふたつであるといわれている。そのうち WWW を世界中に普及させたブラウザ¹というソフトウェアの草分けが”Mosaic (モザイク)”である。Mosaic は 1993 年、イリノイ大学アーバナ・シャンペーン校内・米国立スーパーコンピュータ応用研究所(NCSA: National Center for Supercomputing Applications) で当時学生だったマーク・アンドリーセン (Marc Andreessen)を中心に開発された。Mosaic がブラウザとしても

5 っとも優れていたのは、これまでは文字しか表示できなかった WWW(World Wide Web)を画像表示までを可能にした点である。画像表示ができることによって、WWW はこれまでのアカデミック領域の人だけを対象とした無味乾燥な学術論文検索ツールから、より一般的な大衆も使うことのできる情報発信・収集ツールとして姿を変えた。Mosaic の爆発的な伸び

10 を見たイリノイ大学は、Mosaic の権利を自らの大学の管理下におこうとする。これに反発した中心的開発者であるアンドリーセンらはイリノイ大学を去り IT 産業のメッカ・シリコンバレーに渡って、Mosaic を超えるブラウザの開発に着手することを決意する。そして、シリコングラフィックス社の CEO もつとめベンチャーキャピタリストとして活動していたジム・クラーク (Jim Clark)と共に、新しいインターネット社会を創造するためのベンチャー事業を興した。会社を興すにあたり、当初社名を”Mosaic Communications”とする予定

15 だったが、”Mosaic” の名称の使用にイリノイ大学からクレームがつき、これに憤慨したアンドリーセンらは「Godzilla (ゴジラ) のように Mosaic を食べつくしてしまう」という意味もこめて、”Mozilla (モジラ)”という造語を作り出した。一時は、”Mozilla Communications”という名前も検討したが、結局 “Network” + “Landscape”の造語から”Netscape

20 Communications (ネットスケープコミュニケーションズ)”という名前に決定された。ネットスケープ社はインターネットの成長を大きく担ってきた代表的な企業の一社であり、同社のブラウザ製品であるネットスケープ・ナビゲーターは同社の技術レベルを語るものとして広く認知されることとなった。学術用に限った使用に関してという但し書きはあったものの無料でダウンロードし利用できるというこの製品は、インターネット人口が爆発的に伸びる

25 中、同社製品は一時 80%近いシェアを獲得し、Mosaic からのシェアを瞬く間に奪っていった。ネットスケープはそのスピード感にあふれた生い立ちや企業風土、まったく新しい技術を基盤にしていることより、インターネットの代名詞的な革新的企業のひとつとして市場からも注目されていた。アメリカの IT による市場牽引される新時代到来を示す、「ニューエコノミー」という言葉は、ネットスケープ社が 1995 年 8 月 9 日に新規株式公開(IPO)したと

30 きに生まれたというのが定説になっている。28 ドルで上場した株価は、その日の終わりに 2 倍の値を付けた。株は上がり続け、ジム・クラークら創始者を含む初期の投資家たちは大金

¹ Web ページを閲覧するためのアプリケーションソフト。インターネットから HTML ファイルや画像ファイル、音楽ファイルなどをダウンロードし、レイアウトを解析して表示・再生する。

を一夜にしてつかみ、ドットコム時代の幕がまさに切って落とされた。

ブラウザ戦争

しかしながら、そうしたネットスケープを中心にインターネット関連ソフトウェア市場が
まわっていた状況も長くは続かず、またもや事態は一変する。ソフトウェア業界の巨人、マ
5 イクロソフト社がインターネット 익스プローラー(IE)というブラウザ製品を開発し、同社
OSのウィンドウズ95とのバンドリングに近い強いソフトウェア連携を武器にして乗り込ん
できたのである。ウィンドウズ95発売以前、マイクロソフトのCEOであるBill Gates(ビ
ル・ゲイツ)は「インターネットはただのおもちゃだ」とその重要性を頭ごなしに否定してい
た。だが、ネットスケープ社の勢いとインターネット人口の爆発的な伸びを見せた1995年、
10 突如として同社の製品戦略をインターネットベースに切り替え、果敢なシェア獲得政策に打
って出る。全てのマイクロソフト製品はインターネットを中心に開発することを宣言し、マ
イクロソフトにとってIEはまさにそれを体現する象徴的製品であった。IEはウィンドウズ
と強く連携されるかたちで開発され、OSと一体化させるような製品戦略をマイクロソフト
はとりつづけた。多くのユーザにとって基本ソフトウェアであるウィンドウズと連携が取れ
15 ること、すでにブラウザが組み込まれていること、多くのウェブページがIEを閲覧用標準
ブラウザとして推奨することなどによって、瞬く間にネットスケープ・ナビゲーターのシェ
アは落ちていった(付表1)。後にブラウザ戦争といわれるネットスケープとマイクロソフ
トの戦いは、OSとの互換性・連携性によるユーザ便益とバンドリングを武器にしたマイク
ロソフトの勝利であり、ネットスケープ社はブラウザのシェアを再獲得すべく、巻き返し戦
20 略の策定を強いられていた。

フリーソフトウェアの潮流

古くからソフトウェアの世界には、「公共財に近いソフトウェア、特にOSに関してはフリー
でなければいけない」という哲学に近い考え方が、マサチューセッツ工科大学(MIT)人工
25 知能研究所のリチャード・ストールマン(Richard Stallman)によるFSF(Free Software
Foundation)を中心に強く存在していた。ストールマンはGNU²プロジェクトを立ち上げ、”
フリー”という意味を「無償である」ということに加えて、「変更・改造に関する自由もあた
える」という二点を強調して、商用にゆだねない公共財としてのソフトウェア開発を啓蒙し
た。ただこれはストールマン一人の強烈なキャラクターと天才的なプログラミング能力に支

² FSFが進めているUNIX互換ソフトウェア群の開発プロジェクトの総称。フリーソフトウェアの理念に従った修正・再配布自由なUNIX互換システムの構築を目的としている。GNUで開発されたソフトウェアに適用されているGPLは、「あらゆるソフトウェアは自由に利用できるべき」というFSFの理念を体現したライセンスとして知られている。GNUは、「GNU is Not Unix」の頭文字をとっており、このプロジェクト自体の再帰性を現している。

えられている部分があまりに大きかった。実際、ストールマンは、驚異的な機能を備え Unix の世界では開発ツールの標準となった emacs エディタ³や GNU C コンパイラ(GCC)⁴を実際 5 ほぼ一人の力で開発し、それを無償で公開・提供していた。だが、たった一人の天才だけに大きく依存しているプロジェクトということもあり、ゆえに上記ツールを利用しておこなわ 5 5 れるべき GNU プロジェクトによる OS 開発の速度は遅く、新たな開発プロジェクトもなかなかなか進むこともなかった。

1990 年代前半、OS についてはマイクロソフトの提供する DOS と、AT&T や HP, IBM など 10 などが開発を進めていた Unix という二つの陣営に分かれて開発が進んでいた。しかし、双方ともにソースコード（人間が実際に書いたり読んだりできるソフトウェアプログラム言語形態）を隠して、バイナリ（機械しか読み取れない実行プログラム言語形態）としていたため、個人で開発を趣味とする PC ユーザはその不満をどうすることもできなかった。その一つの 10 解決は MINIX という形で現れたかのように見受けられた。それはオランダ人の大学教授アンドリュー・トネンバウム(Andrew Tanenbaum)教授が、真のオペレーティングシステムの 15 内部の動作を学生たちに学習させるために自らの手によりスクラッチから書き進められた、Unix に非常に似た構成を持つ OS であった。MINIX は当時世界中の市場を席卷していたインテル 8086 マイクロプロセッサで動作するようデザインされていた。

ソフトウェア工学的観点から言って OS としての MINIX はそれほど卓越した新規性をも 20 ったものとは言いがたかった。だが、画期的な新規性を MINIX はもっていて、それはマイクロソフトをはじめとする当時のソフトウェアベンダーならば頑なに公開しなかったソースコード全てをユーザが見ることができるという、ソフトウェア開発に携わる全ての人間にはこの上のない特長があった。MINIX を構成する 12,000 行にもおよぶ C 言語⁵で記述された OS の中身全てを、それをみたいと思った人間全てが見ることができた。トネンバウムは 25 MINIX ソースコードを実際の題材として提供し、実働に能うオペレーティングシステムの設計にいかに芸術性が必要なのかを説き、自分が持つソフトウェア工学のあり方を自らの著

³ UNIX 環境で最も広く普及しているテキストエディタ。きわめて多機能なことで知られる。FSF を創設したリチャード・ストールマンによって最初のバージョンが作成された。後に GNU プロジェクトの一部となり、多くの開発者の手によって拡張が施された。Emacs Lisp と呼ばれる言語処理系を内蔵しており、Lisp 言語を使って機能を拡張することができる。

⁴ GNU プロジェクトによるフリーなコンパイラ。コンパイラとは、人間がプログラミング言語で記述したソフトウェアの設計図(ソースコード)を、コンピュータが実行できる形式(オブジェクトコード)に変換するソフトウェアである。GCC はほとんどの UNIX 系 OS に移植されている。

⁵ 1972 年にアメリカ AT&T 社のベル研究所で D. M. Ritchie 氏と B. W. Kernighan 氏によって開発されたプログラミング言語。もともとシステムの記述用に開発されたので、ハードウェア寄りの低水準な処理を記述することもできる。実際、UNIX は C 言語で記述されている。C の拡張版である C++言語とともに、現在もっとも普及しているプログラミング言語である。

書とソースコード、ふたつを行き来することでコンピューターサイエンスのプロフェッショナルたちやその卵たちを具体的にかつわかりやすく説明した。世界中のコンピューターサイエンスをこよなく愛するものたちは、来る日も来る日もトネンバウムの書籍を繰り返し読み返しながらソースコードと行き来することで、DOSでもUnixでもなく、自分たちの目の前

5 にあるコンピュータを動かす MINIX システムを理論的・実践的に理解することにまさに心底没頭していた。フィンランドのヘルシンキ大学に籍を置く、Linus Torvalds(リーヌス・トーヴァルズ)もそんな世界中のコンピュータを愛する学生の一人であった。

Linux の誕生

- 10 1993 年、フィンランドのヘルシンキ大学の学生だったリーヌス・トーヴァルズが MINIX を参考にして自ら作成したプログラムをインターネット上でそのソースコードごと公開し、プログラムのバグ発見と修正を依頼した。Linux とは自由に再配布することのできる、独立した Unix 系オペレーティングシステム(OS)のことを指し、その名前の由来はリーヌス(Linus)の名前からきている。1993 年、フィンランド大学の学生だったリーヌスが、MINIX
- 15 を参考にしながらかつ作成したプログラムをインターネット上でそのソースコードごと公開し、プログラムのバグ発見と修正を依頼したことから Linux の開発ははじまった。リーヌスが公開したソフトウェア自体は、当初不完全でバグも多く、仕様に関するドキュメンテーションもほとんどないままに、ソースコードというソフトウェア開発のもっとも粗野な形態のままにインターネット上におかれた。この不完全なソフトウェアは、それまでの開発者たち
- 20 が共通に持っていたフラストレーションを反映させる格好の代物であって、すぐに 100 人以上の人々が参加した。そして 1000 人、2000 人と参加者は日々増加し続けた。

- 半年もたたないうちに、Linux はもはや単なるハッカーのおもちゃではなくなっていた。UNIX の卓越したアーキテクチャを Linux はほぼ踏襲して作られていて、ソースコードを見ればほとんどの開発者は何を意図しているものかを理解することができた。GNU プロジェクトが作り出した無償の強力なプログラムやツール群の力を借りて、Linux は急激に実践レベルのソフトウェアとしての能力を持ち始めた。インターネットをベースにして、世界中のハッカー⁶が自らの技術を出し合って、リーヌスの作ったもともとのソフトウェアを改良し、さらにそれについての改良を他の世界中のハッカーに求めた。この開発コミュニティは質・
- 30 量ともに世界的に爆発的に拡大し、様々なプログラムの改良、追加的ソフトウェアモジュールの提供、バグ修正レポートなど、みるみるうちにその動きはソフトウェア業界の一大ムー

⁶ コンピュータ技術に精通した人。転じて、コンピュータ技術を悪用して他人のコンピュータに侵入・破壊を行なう者を指すことが多いが、この用法は誤用が定着したものであり、高い技術を持った人々に対する尊称である。

ブメントとなった。Linux コード管理におけるリーヌスのマネジメントは、「最高の Linux をつくる」という他の開発者との共有可能な技術的目的に向けられていた。リーヌスは多くの開発者の意見に耳を傾け、自ら書いたソースコード以上にいい品質のものを見極め、それを次バージョンへとすばやく反映させた。さらに大きなリリース前テストも行なうことなく、

5 速いペースでソフトウェア・リリースを、場合によっては一ヶ月単位でのリリースがつづくときもあった（付表 2）。このような開発サイクルのもと、Linux は GNU ジェネラル・パブリック・ライセンス(GPL)⁷のもとにライセンスされ、ソースコードはコピー、学習、そして改良に関して一切の自由を保証した。その後、商業ベンダーも Linux を広める動きに入ってきた。FSF が頑なに主張し続けていた「ソフトウェアはあらゆる意味でフリーでなければい

10 けない」という思想を超越して、Linux の世界において商業化することすらも自由であった。商業ベンダーが具体的に何をしたかと言えば、様々なソフトウェアをコンパイルし直し、他の扱い慣れた OS 同様の配布可能なフォーマットにしたのである。Red Hat、Caldera、Debian、そしてたくさんの企業が世界中のユーザからかなりの好感を得て市場参入していった。新しいグラフィカル・ユーザ・インターフェース⁸(X-window、KDE、GNOME など)とともに、商業 Linux ディストリビューションはたいへんな人気を得た。さらには、開発者コミュニ

15 ティにはまるで関心がないものの、ソフトウェアビジネスにおいて一番重要かつお金になる、保守やサポートサービスに関して、Linux はビジネス化することを奨励し、実際 IBM や hp といった巨大商業ベンダーを含めてビジネスの好機と捉えはじめていた。

20 開発者コミュニティの中においては、実際にソースコードを開発するまでのスキルはなくとも、デバッグ⁹を行い問題があることを発見したり、自らちょっとした改良を意見したりするような、あまり高度な技術を持ち合わせてはいないジュニアレベルのテスターと呼ばれる開発者が多くなっていった。デバッグに参加する開発者が多ければ多いほどバグは発見・修正されることで少なくなり、ソフトウェアの品質は向上していった。同時に、インターネット

25 インフラを利用して世界中の一流の開発者とソースコードについてのコミュニケーションを行なうことで、テスターレベルの開発者は自らどんどん一人前の開発者へと育っていった。以前にも増してリーヌスは開発から手を引き、Linux という開発者コミュニティの門番的な役割におさまった。リーヌスの主たる役目は自らがソースコードを開発することではなく、

⁷ FSF の理念に基づいて明文化されたソフトウェアライセンス体系。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を原則とし、使用者に対してソースコードを含めた再配布や改変の自由を認めている。また、再配布や改変の自由を妨げる行為を禁じている。

⁸ GUI とも略される。ユーザに対する情報の表示にグラフィックを多用し、大半の基礎的な操作をマウスなどのポインティングデバイスによって行なうことができるユーザインターフェースのこと。

⁹ コンピュータプログラムの誤り(「バグ」と呼ばれる)を探し、取り除くこと。

参加する開発者たちの関心を育て、プロジェクトを自立させる方向へとマネージすることとなった。個々の参加者たちは自らの技術的な手腕を試し、名声を得るために開発プロジェクトに参加していたし、さらにそれがコミュニティの結束を硬くしていた。彼らの名声は、たとえば開発仕様書（ドキュメンテーション）をしっかりと残すことで Linux コミュニティの中に蓄積された。自ら開発したソフトウェアが他の開発者が開発したソフトウェアとどれほど連携され使用されているのか、ということでも開発者の名声は獲得されていた。

このような動きは、社会学者であり自らも Linux 開発者のひとりである Eric Raymond (エリック・レイモンド) によって「オープンソース」と命名された。Linux は利益追求を目的とした企業体ではなく、ボランティアなコミュニティによって開発されたオープンソース型 OS の代表としての理解をされていた。レイモンドは論文「伽藍とバザール」¹⁰を Web 上に発表した。この論文の中でレイモンドは、誰かが全体の見取り図をきちんと書いて、それにあわせてチームの面々が割り当てられた部分のコードを書き、それをまたリーダーが集めてしっかり緊密に構築して、ミスがないかを細かく一人でチェックして、公開するというやりかたを伽藍(Cathedral)型とした。中央集権的な伽藍型の開発では、リーダーが承認しないような拡張は勝手にやってはいけないし、またリーダーは問題がほとんどなくなったと思えるまで成果をリリースしたりしない。それに比較して、Linux のようなオープンソースソフトウェア(OSS)開発は、あるソースコードがあってそれにみんなが勝手に機能追加やバグ取りを行なっていくような手法で、それをバザール(Bazaar)型とした。ここには決まった方向性などはなく、プロジェクトのリーダーはいるが全体の進み方を指示するのが仕事ではない。開発者たちが出してくる様々な機能改善提案やバグ取りを取りまとめて、矛盾がないようにしていく調整役の任務を果たす。リーダーはリードをしない。これがバザール型開発の大きな特徴である。

通常ソフトウェア開発論では、バザール型開発というのは規模の小さなソフトウェアであればある程度実現可能でも、大規模なソフトウェア開発では現実的に不可能であるといわれていた。たとえば、2,3 名で開発したソフトウェアを考えてみる。この規模のソフトウェア開発であれば、開発者がどんなソフトウェア開発の提案をしようともそれを反映・調整することはそんなに大きな問題ではない。ただ、Linux のような OS に代表される巨大で複雑なシステムでは、そうした変更は一部のソフトウェア変更が他の部分のソフトウェア変更にもつながることとなり、最終的にまるで整合性が取れず収集がつかなくなり、非常に難しい

¹⁰ 「The Cathedral and the Bazaar」、Eric S Raymond, <http://www.catb.org/~esr/writings/cathedral-bazaar/>

ものと考えられていた。Linux におけるオープンソースソフトウェア開発はそれを見事に覆し、見事な製品を作り上げていた。

1998 年年末、ネットスケープ社はマイクロソフトのインターネットエクスプローラーに対抗した巻き返し戦略の切り札として、ネット上の開発者コミュニティに自社の製品戦略の一部をかけるという意思決定を行う。つまり、ソフトウェア企業にとって生命線とも言えるソースコードをネット開発者コミュニティに公開し、インターネット上の無名無数の開発者にこれからの開発を委託することにしたのである。

10 モジラプロジェクト(Mozilla Project)

1999 年 3 月 31 日、ネットスケープの次世代ブラウザ製品である、ネットスケープのソースコード¹¹が、専用サイト”mozilla.org”を通じて無償公開された。ネットスケープ創設時のスピリットを再度取り戻すため、プロジェクトは「モジラプロジェクト」と命名された。開設当初、単純なスタンドアローンのブラウザに関するものであったとはいえ、このサイトには膨大なアクセスが世界中から集まり、モジラのソースコードはひっきりなしにダウンロードされた。サイトへのページビュー（集客数）、実際にダウンロードされたソースコード数からいって、世界中の数万を超える開発者がこれまでのネットスケープの知的財産を一晩にして手に入れることができた。このソースコードを元ネタにして、Linux のようなバザール型のコミュニケーションを主体とした開発が自然と起こっていくことをネットスケープ社はもちろん望んでいた。マイクロソフトのウィンドウズに対抗して出来上がった Linux の成功を見ていたために、反マイクロソフトという開発者たちの感情的な要因も巻き込んで、同じマイクロソフトのインターネットエクスプローラー(IE)を対抗としているネットスケープ製品も同じようなムーブメントが起きることを、レイモンドの論文を読み Linux の開発論に感化されていたネットスケープの経営陣は確信していた。モジラプロジェクトはネットスケープ社から切り離れた形で行ないネットスケープ社はソースコードを提供するものの、あくまでその運営はモジラプロジェクトというコミュニティに任せられることになっていた。

しかし、モジラプロジェクトでの開発者コミュニティの動きは想定していたほど当初からあまり活発ではなかった。実際にコミュニティに集まってきたのは、ネットスケープ社の開発

¹¹ 人間がプログラミング言語を用いて記述したソフトウェアの設計図。そのままではコンピュータ上で実行することはできないため、コンパイラなどのソフトウェアを用いてオブジェクトコード(ネイティブコード)と呼ばれるコンピュータの理解できる形式に変換され、実行される。

者自身が多く、彼らの多くはモジラプロジェクトを構成する大きなコンポーネントのエリア監督として指名されていた。一方で、フリーの開発者や Linux の開発者などは当初の予想よりもまるで集まってはこなかった。ネットスケープ社はモジラとは別のプロジェクトとして、これまでどおりの企業内開発部門の製品でメール機能など新しい開発コンポーネントが追加された「ネットスケープコミュニケーター」という別製品の開発を継続していた。さらに、実際にオープンソースとしてネットスケープがモジラプロジェクトに公開されているソースコードが不十分であり、多くのオープンソースプロジェクトが前提とする GPL を完全に踏襲していなかった。それゆえ、ネットスケープはモジラのコードベースに対しての変更を最終的に拒絶したり、あるいは強制的な変更を加えたりする権利をいまだに保留していた。同時に、ネットスケープ社以外のコミュニティ開発者たちが実際に自ら作ったソフトウェアを確認するには別のソフトウェアライブラリのライセンスが必要であったり、実際には公開されていないソースコードの一部があったりしたことが徐々に開発者コミュニティへと明らかとなった。これらの事態が明るみになるにつれ、ネットスケープへの批判が開発者コミュニティから相次ぎ、IE の独占的な状況を是としない反マイクロソフトに共感して集まってきたハッカーの中にも脱会者がおおくなりはじめた。こうした状況下ゆえ、モジラの開発自体も遅々として進まなかった。リリースは延期に延期を重ね、商品クラスの高品質ブラウザは結局リリースされることはいつになるのかわからなくなっていた。ネットスケープは、モジラ製品開発の要としてコミュニティを活用しようとしている一方で、ネットスケープがどれほどコミュニティに対して開発イニシアチブをとるのが望ましいのか、「コミュニティ」という体質を別とする組織体といかに関わり合うのがもっとも企業組織として妥当なのかということに明確な方針をもっておらず、わからないまま模索する状況が長く続いていた。このような状況が続くうち、ハッカーたちは「ネットスケープ社は、コミュニティの持つ技術を期待しているのではなく、ネットスケープ社の株価維持の話題のひとつとして、ビジネス・広告宣伝的な話題として利用されているのではないか」という不満を公にするようになっていった。自社を揺るがすソフトウェアの脅威としてオープンソースを認識・調査していたマイクロソフトも、モジラプロジェクトに関しては冷ややかな目で分析していた（付属資料 3）。

そんな状況が長らく続いた混乱の中、1998 年 11 月に AOL によるネットスケープ買収の発表が突然報じられた。ブラウザを提供する企業として、さらにはインターネットを牽引してきたネットスケープとしての企業アイデンティティとして、今後どうなっていくのかということに議論が IT 業界の中で巻き起こった。同様に、モジラプロジェクトの中にも今後のモジラ製品開発はどうなるのか、ということが大いに議論されることとなった。この緊急事態に当たってプロジェクトリーダーであるジェレミー・ザウインスキーは次のようなコメン

トを寄せた。「オープンソース・ブラウザ開発プロジェクトであるモジラプロジェクトは、これまでもそうだが今後も AOL から、ネットスケープからも、組織として独立した存在であり続ける。ただ、今後 AOL がどのような支援をこのプロジェクトに対して施してくれるのか、大いなる懸念がある」。ザウインスキーの意図していたものに反して、ザウインスキー声明はコミュニティに対してポジティブなアナウンス効果を果たすことはなかった。巨大情報サービスコングロマリットを標榜しネットスケープのポータルおよびサーバ事業といった主に企業向けの有償ビジネスに強い関心のある AOL にとって、GPL に則ってフリーの独自ソフトウェアの開発を行うモジラプロジェクトはほとんど意味のないものであるはずだと開発者コミュニティは感じていた。製品開発スピードはますます遅くなる一方であった。

10

実際にそうしたコミュニティの予想は的中する。1999年3月末に、AOLは700~1000人にもなるリストラ計画を発表し、そのうちの約半数はネットスケープ関連部門であった。この段階で、リストラ対象社員のみならず創業時からネットスケープを支えてきたエンジニアや創業マネジメントメンバーの多くは同社を去り、新たなる夢を求められる企業へと移っていった。このような動きと連動するように、モジラプロジェクトは自然と縮小しプロジェクト内のコミュニケーションによる場の活気も日に日になくなっていった。

15

1999年4月1日、ついにプロジェクトリーダーのジェレミー・ザウインスキーは自らの AOL・ネットスケープ社からの退社とモジラプロジェクトからの引退を自身のホームページを通じて表明した。そこにはこうかかっている。

20

“私の救いは、私はネットスケープ社を退社するのではなく、AOL社を退社することになったということである。……（中略）…… 私が最も恐れるのは、モジラプロジェクトの失敗がオープンソースの象徴として一般的に取り上げられてしまうことである。モジラプロジェクトが持っているどんな問題でも、それはオープンソースがうまく動かないからというのが原因ではない。オープンソースは動くが、オープンソースは万能薬ではないことも確かなことなのである。”

25

■付属資料

1. ブラウザ市場のシェア(%)

	1995	1996	1997	1998	1999	2000	2001	2002	2003
Netscape	80.1	73	57.6	54	30.7	17.7	9.8	7.2	3.7
Microsoft	2.9	20	39.4	39	67.5	82.2	89.9	91.5	95.9
その他	17	7	3	7	1.8	0.1	0.3	1.3	0.4

資料:1995-1997年のデータは Dataquest、1999-2003年のデータは WebSideStory、1998

5 年のデータは ZD Market Intelligence の調査による

2. Linux の製品リリース実績

1991年 トーパルズ開発開始、バージョン 0.01 カーネルをリリース

この間、2年5ヶ月で119回のバージョンアップ

10 1994年 バージョン 1.00 カーネルをリリース

この間、2年3ヶ月で234回のバージョンアップ

1996年 バージョン 2.0 (安定版) をリリース

この間、4ヶ月で21回のバージョンアップ

1995年 バージョン 2.1.0(開発版)をリリース

15 この間、2年4ヶ月で142回のバージョンアップ

1999年 バージョン 2.2.0(安定版)をリリース

3. ハロウィーン文書

1999年10月、マイクロソフトのオープンソースに関する詳細な社内リサーチが、レイモン
20 ドによってリークされた。ハロウィーン近くにリークされたことより、ハロウィーン文
書と呼ばれる。当初ノーコメントであったマイクロソフトも、後にこれが公式な社内文書で
あることを認めており、その中でネットスケープのモジラに関しての調査結果が報告されて
いる。ハロウィーン文書はオープンソースよろしく GPL として公開されている。

25 ~以下日本語版ハロウィーン文書より抜粋(<http://cruel.org/freeware/halloween.html>)~

ネットスケープ

ブラウザ業界での信用を更新すべく、Netscape は最近になって Mozilla ソースコードをリリースして、OSS コミュニティをそのまわりにつくりあげようとしている。

5 組織とライセンス

Netscape の組織とライセンス提供モデルは、おおざっぱに Linux コミュニティと GPL に基づいているが、少しちがうところもある。まず、Mozilla と Netscape Communicator という別々のコードベースがあって、Netscape のエンジニアがそのシンクロを行っている。

- 10 ● Mozilla = OSS 版のフリー配布可能なブラウザ
- Netscape Communicator = ブランドつきの、ちょっと手を入れたバージョンの Mozilla (例: ホームページがデフォルトでは home.netscape.com になっている)。

15 完全な GPL とは異なり、Netscape は Mozilla のコードベースに対しての変更を最終的に拒絶したり、あるいは強制的な変更を加えたりする権利を保留している。そして大きなコンポーネントの「エリア監督」として (いまのところは) Netscape のエンジニアたちが指名されている。

強み

- 20 OSS コミュニティ内の反 MSFT 感情を利用する

ほかの OSS プロジェクトにくらべて、Mozilla はマイクロソフトのエスタブリッシュメントに対するもっとも直接的で短期的な攻撃であるとみなされている。この要因だけでも、開発者たちを Mozilla コードベースに引きつける要因としては大きなものになる。

25

信頼性の強化

Mozilla ソースコードが提供されたことで、ブラウザ業界における Netscape の信頼性はわずかながら上昇した。

- 30 「コードをリリースしたことで、かれらは Wordstar が消滅したような形では、この世から消滅したりはしないということを保証したことになる。Mozilla ブラウザは、今後十年間は十分に生き延びるだろう。そのユーザベースは縮小したとしても。」

大きな悩みを解決

ブラウザは広く使われ、普及している。したがって、「目の前の問題」を解決したいと思っている人や、あるいはバグをなおしたいと思っている人のブールはかなり大きいかも知れない。

5 弱み

① 飽和点をすぎた開発

Mozilla はすでに、IE4/5 とほとんど肩をならべる水準に達している。したがって、開発チームを暗黙のうちにまとめるための、追いかけるべき強い前例が存在しない。

10

Netscape は、自社の最高の開発者数名を、Mozilla コードベース管理の仕事にフルタイムであたらせている。Mozilla が新しい地平を開拓するにあたり、これがどう貢献するか（あるいはまったく貢献しないか）興味深いところである。

15

② ノウアスフィア（開発者が新たに開発することのできる先進的な領域）が小さい

おもしろい弱点として、OSS ブラウザにとって残された「ノウアスフィア」の規模があげられる。

20

1. スタンドアロン型のブラウザは、基本的にもう終わっている。

スタンドアロン型のブラウザで、今後開発すべき大きな目立つ領域は、もはや残っていない。言い換えると、Netscape はすでに問題のおもしろい 80% 分を解決してしまっているのだ。Netscape のコードの残り 20% をデバッグ・修正したところで、エゴの満足はほとんど・まったく得られまい。

25

2. Netscape の商業的な利害が、ノウアスフィア貢献者の効果を低下させる。

Linux コードベースにおけるリーヌス・トーヴァルズのマネジメントは、最高の Linux をつくるという目的に向けられていると言えるかもしれない。ところがそれに対して Netscape は、Netscape の商業的・ビジネス上の利害に基づいて、コードマネジメントに関わる決断を行う権利をはっきりと留保している。開発者のコードは、重要な製品をつくるためにではなく、Netscape の株価に奉仕させられているわけだ。

30

③統合化のコスト

おそらくは Mozilla 的試みにおける唯一最大のつまずき石は、ユーザはブラウザの機能が高度に統合されていることを期待するというところだろう。すでに述べたように、統合開発・試験は並列処理不可能な活動であり、したがってこれは、OSS プロセスがむしろ有害に作用する分野である。

ときに、IE5+ における新たな作業のほとんどは、単に新しいコンポーネントをブラウザそのものに統合することだけでなく、むしろそれを OS 内に統合する作業に費やされている。これに対抗するのはとても苦勞を強いられるはずだ。

将来予想

以上から論じられるのは、いまのところ非常に成功している Apache や Linux とはことなり、Netscape の Mozilla の試みは以下のようなになるということである。

- Linux など一部の UNIX では支配的なブラウザとなる
- 長期的にはずっと IE の後塵を拝し続ける

ソースコードがリリースされたのがほんのしばらく前（1998 年 4 月）だったことを考えると、Mozilla に対する関心はすでに薄れつつある気配が見られる。まったく科学的とはいえないながら、そうした証拠を 4 月から 6 月にかけての Mozilla メーリングリストのボリュームの低下などに見ることができよう。

Mozilla メーリングリスト	1998 年 4 月	1998 年 6 月	減少率 (%)
追加機能の要望	1073	450	58%
UI 開発	285	76	73%
一般的な議論	1862	687	63%

なお、Mozilla メーリングリストの社内ミラーは <http://egg.Microsoft.com/wilma/lists> にある。

参考資料・文献

1. 「Mozilla Project Homepage」、<http://www.mozilla.org>
2. 「resignation and postmortem」、<http://www.jwz.org/gruntle/nomo.html>, 1999
3. 「インターネットヒストリー オープンソース革命の期限」、Neil Randall (村井純監訳)、オライリー出版、1999
4. 「Linux はなぜビジネスになったか～コミュニティアライアンス戦略～」、佐々木裕一・北山聡 (國領二郎監修)、NTT 出版、2000
5. 「モジラはなぜ失敗したか」、<http://www.mochioumeda.com/archive/ss/990419.html>
6. 「Mozilla.org はネットスケープではない」、Chris Oakes、<http://www.hotwired.co.jp/news/news/Technology/story/1688.html>、1999
7. 「モジラのリーダーが敗北宣言」、Chris Oakes、<http://www.hotwired.co.jp/news/print/2252.html>、1999
8. 「『Mozilla』プロジェクトは継続」、Chris Oakes、<http://www.hotwired.co.jp/news/news/1677.html>、1999
9. 「History of Linux」、Ragib Hasan、<http://ragib.hypermart.net/linux/>、2002
10. 「The Cathedral and the Bazaar」、Eric S Raymond、<http://www.catb.org/~esr/writings/cathedral-bazaar/>

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

sample

不許複製 慶應義塾大学ビジネス・スクール 2004 Jun.

コンテンツワークス株式会社 BookPark サービス
