社会イノベーション研究
第９巻第２号（23-34）
2014年10月

2014年９月１日掲載承認

創発的組織と協働要因－オープンな開発とは－

神　戸　和　雄

拙稿において成功的なオープンソースソフトウェアの開発組織であるFreeBSDに見る協働体制を検討した[1)]。オープンソースソフトウェアの開発において参加者は無報酬でありながら高いモチベーションを保ちつつ，多くの参加者が協力するさまを観察することができた。

本稿では成功するオープンソースソフトウェア開発組織の成功要因を検討するとともに，企業がどのようにオープンソースソフトウェアで収益をあげているか，またその発展形としてのオープンな開発との関連について考察することとする。

1.　協力関係が創発的に機能するためには

オープンソースソフトウェアでは多くの人々が特に報酬を得ることもなく，高いモチベーションを維持しつつ開発に参加している。しかしながら，すべてのオープンソースソフトウェア開発組織が成功している訳ではない。一般に多くに人々がチームを構成し協力していく場合，どのような機能が働くかを考え，成功しているオープンソフトウェア開発組織との共通点および相違点を考察することとする。

- **組織に対する心理的安心**

人々が創造的な活動を行う際，評価がどのようにされるかが大きな影響をも

1)　拙稿「情報システム構築のための協働体制」十川廣國編著『経営学イノベーション③経営組織論』（第２版）中央経済社，2013年

つ。一般に人々が新しいことに挑戦する場合，評価の問題が人々の意欲に大きく影響してくる。「権限委譲し，内発的に人々をどう気づけるためには人々が新しいことに挑戦して，いささかの失敗をしてもその挑戦意欲を評価するという加点主義の人事評価を行うという方針をもっていないといけないといえる。減点主義の人事評価がとられるとすれば，人々の挑戦意欲は当然のことながら減殺されてしまうことになる[2)]」のである。

オープンソースソフトウェアにおける評価とはなんであろうか。そもそもオープンソースソフトウェアの組織では一般の組織とは評価の軸が異なると考えられる。一般の組織では評価・報酬，懲罰といった要因が軸となるが，オープンソースソフトウェアの開発組織での評価軸は異なる。営利を目的としないオープンソースソフトウェア開発組織においてはこの項目自体が存在しない。人々は賞賛と自らが好ましいと思うソフトウェアの機能を実現するために自主的に開発に参加している。オープンソースソフトウェアオープンソースソフトウェアの開発においては創発的な開発姿勢が非常に大きな意味を持ってくる。

誘因はどこにあるのだろうか。いくつかの要因を挙げることができるが，オープンソースソフトウェアで見られる特徴のうち，通常の営利組織に適用が可能であることが望ましいと思われるものを考えてみることにする。特に重要なものとして参加者間の信頼関係を挙げることができる。

一般の組織でもチームワークが機能するためには組織に対する信頼，組織の属する人々の間の信頼が重要となる。評価軸が異なるオープンソースソフトウェア開発組織では協働体制への期待と人々の間に存在する信頼の重みがさらに高くなってくる[3)]。

成功しているオープンソースソフトウェア開発組織には協働体制への期待が存在している。お互いが共通の目的に向かって努力することで開発が進む。それらを裏付けているのがお互いが共通の目的を持ち，協働することによって成果を達成できるという信頼である。その信頼は個々人の間の信頼によって裏打ちされることになる。

協働関係がうまく機能する場合，心理的要素は重要な働きをする。協働を円滑にするためには心理的に安全な環境が必要であると考えられる。心理的安全

2） 十川廣國著『経営学イノベーション①経営学入門』（第２版）中央経済社，2013年，174頁

3） SHUK YING HO and ALEX RICHARDSON, ”TRUST AND DISTRUST IN OPEN SOURCE SOFTWARE DEVELOPMENT”, *Journal of Computer Information Systems,* Fall 2013, p. 85

があればより建設的に真摯で忌憚のないやりとりをできるようになる。ここでいう心理的な安全とは，なにかミスをしても，そのために他の人から罰せられたり，評価を下げられたりする可能性がない状態のことを指している。このような環境では他の人に手助けや情報を求めても相手に悪く思われないとも感じられる。このような心理的に安全な環境が構築されるためには，そこに参加する人々が互いに信頼し尊敬しあっている場合に生まれやすいと考えられる[4)]。

特にソフトウェアの開発においてプログラマは時として自己のアイディアを結実するあまり協力体制を組むのが困難になる場合がある。この困難を乗り越えるのに機能しているのが特に技術的な信頼である。

- **知識・技術に裏打ちされた信頼**

信頼を考えた場合，いくつかの類型化がされている。恐怖に基づく信頼 (deterence-based trust)・知識や計算に基づく信頼 (knowledge-based trust)・アイデンティティに基づく信頼 (identity-based trust) という分類[5)] や，客観的な事実によるどうかによって認知的信頼 (cognitive trust)・感情的信頼（情緒的信頼, affective trust）という対象に応じてさまざまな分類がとりあげられている[6)]。

オープンソースソフトウェアと特に関連が深い信頼とはどのようなものであろうか。オープンソースソフトウェアにおいては知識・技術に基づく信頼が重要な働きをすると考えられる。オープンソースソフトウェアに参加する人々の誘因の一つは自らのアイディアを結実することにあり，協働する人々に対する

4） Amy C. Edmondson, *TEAMING: How Organizations Learn, Innovate, and Compete in the Knowledge Economy,* Jossey-Bass, 2012, pp. 118-119. エイミー・C・エドモンドソン（著), Amy C. Edmondson（著), 野津智子（翻訳)『チームが機能するとはどういうことか』英治出版, 2014 年, 154～155 頁

5） Lander, Maria Cristina1, Purvis, Russell L., McCray, Gordon E. and Leigh, William , “Trust-building mechanisms utilized in outsourced IS development projects: a case study”, *Information & Management.* Mar 2004, Vol. 41 Issue 4, pp. 509-528.

6） DANIEL J. MCALLISTER, “AFFECT- AND COGNITION-BASED TRUST AS FOUNDATIONS FOR INTERPERSONAL COOPERATION IN ORGANIZATIONS”, *Academy of Management Journal,* Vol. 38,, No. 1, pp. 25-26　認知的信頼に属するであろう能力，業績といった明確に把握できる事柄に基づいて醸成される信頼であり，感情的信頼とは人々の間に存在する価値観など目に見えないものに基づいた感情的な信頼のことを指している。SHUK YING HO and ALEX RICHARDSON, “TRUST AND DISTRUST IN OPEN SOURCE SOFTWARE DEVELOPMENT”, *Journal of Computer Information Systems,* Fall 2013 における実証研究ではオープンソースソフトウェア開発における認知的信頼の重要性が強調されている。

技術的な信頼がない限り，協力関係は成り立たない。技術的に信頼ができなければ自らの意思を貫き通そうとし，協力関係は破綻する。

オープンソースソフトウェアとして成功している開発組織として FreeBSD は内部構造がある程度，明らかにされている。そこではメンバーの選出過程，組織構造の構築がこの能力に基づいた信頼関係を醸成するのに寄与している。

そこにはオープンソースソフトウェアの組織でのポイントとなる構造がある。それは高い信頼を醸成する仕組みである。選挙によってあるいは認知によって構成されているがそこには高い信頼がある。FreeBSD 開発体制において実際にソースコードを編集する権限を持つ役割を担っているのがコミッタと呼ばれる人々である。新たなコミッタは開発に積極的に寄与していると認知された人物が既存のコミッタの推薦のもと，認定される仕組みになっている。この選出方法により，コミッタとなる人物は技術的な信頼を得ていると考えられる。

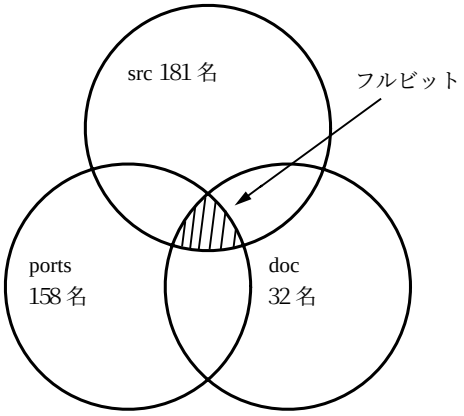
オープンソースソフトウェア開発においては参加者の参加・離脱が自由であり，開発自体が開発者の自発性の上になりたっていることから協働者間の信頼は必須である。特にまとめやくとなる者は協働者の尊敬にまで近い評価を得ることできる人物であることが必要である。開発の主体であるコミッタは既存のコミッタに信頼され推薦された者であり，また最上位のコアチームはこのようなコミッタの互選によって選ばれた人たちである。おのずと技術に裏打ちされた信頼関係が構築されており，協力関係が推進される原動力となると考えられる。

- **組織運営を円滑にするつ広い知識をもつコアパーソンの存在**

大規模なオープンソースソフトウェア開発組織ではさまざまな活動領域が存在する。FreeBSD の場合，主たる活動領域はコミッタは主に受け持つ領域が決まっており，アクティブなコミッタ 330 名の内訳は次頁の図に示すように次のようになっている。src（ソースコード）181 名，ports 158 名，doc（ドキュメンテーション）32 名である。図に示したように同一人物が複数の領域にまたがってコミッタとして活動している。またすべての分野に参加する資格を持っているコミッタはフルビットを持つと表現し，このフルビットを持つコミッタは技術的な信頼も高く全体の整合性をとることに大きく寄与していると考えられる。

オープンソースソフトウェア開発では．多くの利用者の意見をとりいれ．よりよいものを作るという共通の目標がある。この目標はことこまかに定義され

図　アクティブなコミッタ 330 名の構成



（出所） “Dru Lavigne talks about FreeBSD”, written by Bill Toulas on July 2, 2012. Posted in interviews on Unixmem (http://www.unixmen.com/dru-lavigne-talks-freebsd-interview/) を基に作成。

たものでなく．大枠を示した中で各自が自らのアイディアを寄せ合って作成していくところに機能する要点がある。その調整役としてすべての分野に通じる技術的に信頼される人物はコアパーソンとして大きな貢献をしている。

- **信頼を醸成する組織構造　創発性を高める管理体制―行動の自由**[7)]

FreeBSD では全体プロセスがあまり明確に定義されないことから，開発作業の多くは個人単位，あるいはお互いに理解しあっている複数のコミッタ等によるチーム単位で行われている。開発にあたっては Subversion というツールを用い，ソースコードの改変をお互いが常に参照できるようになっている。

実際の作業を行う際には，電子メール等を用いたコミュニケーション，ソースコードの作成，できあがったソースコードのやりとりにより，動作可能性の検証や改良点が検討される。技術的に競合する複数の案がある場合には技術的な観点から議論し，参加者が納得した上で実装が行われる。またアプリケーションあるいはその一部においてメンテナと呼ばれる「主にこのソースコードを管理する人」が定められている場合がある。メンテナが定められている場合には，コミッタがメンテナに対して確認を求め，承認が得られた時点で実際の修正を行うといった，緩やかな管理体制がとられている。個々の参加者はあくま

7） 拙稿「情報システム構築のための協働体制」十川廣國編著『経営学イノベーション③経営組織論』（第 2 版）中央経済社, 2013 年 229～231 頁および “The FreeBSD Project” https://www.freebsd.org/ja/（参照 2014 年 7 月 21 日）

でも自発的な参加，明確な線引きのない開発プロセスとわずかな取り決めによって自由な参加が可能となっている。ほとんどのコミッタは自分の余暇を利用し開発に参加している。またボランティアであり，開発に貢献することによる直接の報酬はない。参加も離脱も自由であり．雇用契約など明示的な契約は存在せず，世界中にちらばっているため明確な役割分担を固定的に運用することは難しいため，全体プロセスはあまり明確に定義されていない。個々のコミッタが守るべき心構えや活動に関する抽象的に定義がされている Committers' Guide と呼ばれる文書が存在する程度である。唯一，FreeBSD プロジェクト全体で定められているプロセスはリリースに関わるものである。あらかじめ目標とするリリース日を掲げ，予定に間に合うように全体の作業が進められる。

以上．述べた FreeBSD の開発組織の特徴はあくまでも緩やかな管理体制の下に自発的に運用されているという点にある。ではなぜそのような組織が大規模なシステムを開発できるのであろうか。図式化すれば，一貫した体系を持つように見えてしまうかもしれないが，どの部分をとってもボランティア活動として行われ，明確な権限体系も存在しない。一般の企業組織と比較した場合，それはあたかもインフォーマルな組織の集合体と考えられるべきものである。インフォーマルな形態をとることにより，議論は活発化し，自由が発想が促される。

このインフォーマルで自発的な組織が高い信頼関係のもと，創発姓を発揮しながらよりよい成果を挙げていくところにオープンソースソフトウェア開発組織の利点が存在する。

2. オープンソースソフトウェアとビジネス

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。特に新たな開発が頻繁に行われるソフトウェアの世界ではオープンな協力関係が新たな開発を促進するのに寄与するが，本来，利益を目的としていないオープンソースソフトウェアは企業の収益とどのような関連を持つのであろうか。

・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をビジネスに適用した場合の類型としてチェス

ブロウは価値の高いものから低いものへと下記の４ステップを挙げている[8)]。

・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上げている。

「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。近年ではコマーシャル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。同社で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いれるとともに，購入したユーザはソースを改変し，より業務に適切な形での運用が可能となっている[9)]。

「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。

8） 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 頁。Chesbrough, Henry William., *Open business models : how to thrive in the new innovation landscape,* Harvard Business School Press, 2006, p. 45.

9） http://thinkit.co.jp/free/article/0607/1/1（参照 2014 年 7 月 21 日）

Oracle は MySQL, GlassFish, Java, Linux, PHP, Apache, Eclipse, Berkeley DB, NetBeans, VirtualBox, Xen といった広く利用されているオープンソースソフトウェアの開発に参加，あるいは多大な投資を行っている[10)]。特に最も普及していると考えられる Java に関して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。

チェスブ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であるが，Google 社は Android を通じて多くの収益を上げている[11)]。オープンソースの 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上げていないが，「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 の利用があってこそ，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[12)]。

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，さまざまなアイディ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

10）“ オラクルとオープンソース ”
http://www.oracle.com/jp/technologies/open-source/index.html（参照 2014 年 7 月 25 日）

11）“ Google がオープンソースの Android から利益を生み出すカラクリとは？ ”
http://gigazine.net/news/20140124-google-android−profit/の 2013/01/25 12:35 追記
「The Guardian が Google から明示された情報によると，GMS ライセンスを取得すること自体に費用はかからないが，取得のためのテストに費用がかかるとのことです。製造メーカーは，テストを行う Foxconn や Archos にテスト料を支払うことになっています。」（参照 2013 年 1 月 25 日）

12） “Google Inc. Financial Tables (Q42013)”,
http://investor.google.com/pdf/2013Q4_google_earnings_data.pdf（参照 2014 年 7 月 25 日）

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のである。

3. オープンな開発とビジネス

オープンソースソフトウェアとして開発しているものが外部企業に活用され，新たなビジネスモデルとして結実する場合が多々，存在する。オープンソースソフトウェアの使用規約には開発した成果物の公開を義務づけず，有償での販売を許すものもあるからである。

ソフトウェアに限らず，企業外の知識を活用して新たな製品を開発するオープンな開発はかなり古くから行われてきた。大きな変革としてはマウスをコンピュータに活用することによってコンピュータ技術を人々の身近な例にした例がある。

マウスを最初に商用化し世に生み出したのはアップル社である。アップル社はゼロックス社のパロアルト研究所でのマウスの使用を視察し，マウスを使った Mac 開発のインスピレーションを受けたという。しかし，このマウスの利用自体，ゼロックス社のパロアルト研究所の発明ではない。マウスとキーボードの連携によるコンピュータの活用の歴史は 40 年以上前にさかのぼる。マウスはダグラス・エンゲルバート氏によって考案され[13)]，バロアルト研究所のコンピュータ開発に利用されていたのである。アップル社のビル・ゲイツとほぼ同時期にマイクロソフト社のビル・ゲイツもバロアルト研究所を訪れ，Windows の開発へとつながったことは興味深い。コンピュータの歴史を変えたマウスの利用も必ずしもオリジナルではなく，既にあった外部の技術を活用したという事実が存在している。

また，より意図的に外部からの情報を求め，新しい製品にむすびつける動きも存在する。

IBM は顧客から学ぶという目的で FOAK (First-Of-A-Kind) プログラムを通

13） 米スタンフォード大学が 1968 年に行われたマウス利用のデモンストレーションに関して詳しい資料と映像を残している。“Douglas Engelbart 1968 Demo” http://web.stanford.edu/dept/SUL/library/extra4/sloan/MouseSite/1968Demo.html（参 照 2013 年 12 月 11 日）

じて顧客との協力により研究開発を製品へと結実している。顧客とのビジネス上の問題を解決する－ために．顧客と IBM 研究員が協力し．IBM リサーチの最新の研究成果から世界初の－ソリューションを作り出すための実証研究プログラムとして活用されている。

FOAK 以外にも顧客との連携により新たな製品を開発している例は多数，存在する。協力相手は必ずしも顧客だけではなく，ライバル会社と連携する場合もある。近年はオープンイノベーションとしてより意図的に企業の境界を取り払い，外部の知識の活用によって製品開発のスピードと質を高める動きが活発である。P&G はオープンイノベーションによる製品開発に注力しており，Connect + Develop と表して広く外部から技術を募っている。自社のアイディアが使われることによる理しくよりも自社単独の開発よりも早期に新製品を世に送り出し，技術の使用料を上回る利益を獲得している[14)]。

これらは必ずしもオープンソースソフトウェア開発ではないが技法的に新たな技術を取り込む，オープンな開発の利点を活用するという点で共通点が見える。

4. むすびにかえて―オープンソースソフトウェア開発技法の活用―

本稿ではオープンソースソフトウェア開発組織を支える信頼関係について検討し，外部からのオープンな知識の獲得によってイノベーションが達成されるオープンな開発について触れた。両者の間には外部知識の活用という共通項は見られるものの，開発体制には大きな違いが見られる。

オープンソースソフトウェア開発においては Subversion や Git といった大規模にソフトウェアを共有する技術が存在し，目的も一定の方向が設定されている。それに対してオープンな開発では不特定多数の相手を探索する必要があり，そのための技法が確立されているとは考えにくい。一部には Web やサテライトを通じオープンイノベーションを促進する仕組みを整える動きもあるが[15)]，まだその技法が確立されているとは言いがたい。オープンソースソフト

14） “Learn about Connect + Develop, Current Product Development Needs, Open Innovation Partnerships - P&G Connect + Develop” http://www.pgconnectdevelop.com/home/home3.html には多くの事例が掲載され，また協力相手を広く公募している。（参照 2014 年 7 月 24 日）

ウェア開発のように同一も目的が設定されていない中で，目的自体も探ることを必要とする側面を持つオープンイノベーションでは定式的な技法を見つけることは難しい可能性はあるが，オープンソースソフトウェア開発体制のエッセンスと比較可能な部分と比較可能な方策などを実態調査することにより，ソフトウェアにとどまらず，より一般的なオープンな開発を考察していくことが必要と考えている。

15）“ オープンイノベーション情報メディア『cotas』” http://cotas.jp/（参照 2014 年 7 月 18 日）