**論　説**

# 知識創造プロセスとしてのオープンソース・ソフトウェア開発[1)]

竹　田　昌　弘

目　次



## はじめに

Linux オペレーティング・システムの開発と普及は，自発的にプロジェクトに参加する開発者の集合であるオープンソース・ソフトウェア開発コミュニティが，マイクロソフトなどの企業が開発するオペレーティング・システム製品に匹敵する機能，品質，信頼性を持つソフトウェアを開発しうることを証明した。ソフトウェアは，一種の知識とみることができる。さらに，オープンソース・ソフトウェアは，誰でもが自由に使用することができる公共財である。すなわち，オープンソース・ソフトウェア開発コミュニティは，公共財としての知識を創造するメカニズムと見ることができる。また，オープンソース・ソフトウェア開発コミュニティでは，コミュニティに参加することによって，個人はソフトウェア開発の能力向上の機会が与えられ

1）本稿は，2004 年 11 月 14 日に名古屋工業大学で開催された経営情報学会秋季全国研究発表大会において報告した「オープンソース・ソフトウェア開発コミュニティにおける知識の創造」の内容に，報告の際の討議内容をふまえて大幅に加筆したものである。報告に際し貴重なご意見を頂戴した先生方に，この場を借りて謝意を述べさせて頂きます。

る。これは，メンバーが入れ替わりつつも，全体としてのソフトウェア生産能力を維持していることからもわかる。すなわち，オープンソース・ソフトウェア開発コミュニティは知識を生産するための知識，すなわち，メタ知識としてソフトウェア開発能力を学習する機能も有している。このようなコミュニティは，実践コミュニティと見なすことができるだろう。

本稿では，これらのメカニズムを整理した上で，企業組織における知識創造へのインプリケーションを提示する。

## 1. 公共財としてのオープンソース・ソフトウェアとそのマネジメント

オープンソース・ソフトウェアは，ソースコードを公開し，誰でも自由に利用できる。しかも，単にソフトウェアをハードウェア上で実行するというだけでなく，ソースコードを読むことによってソフトウェアの内容を学習し，さらに，自らの必要に応じて改変を行い，再配布することが自由にできる。したがって，オープンソース・ソフトウェアは，公共的に，誰もが自由に利用できる資源であり，公共財と考えることができる。しかも，その公共財の充実を誰もが担える状況になっている[2)]。

公共財を整備するためのマネジメントには，いくつかの留意点がある。まず，一般的な公共財のマネジメントについてまとめた上で，オープンソース・ソフトウェアの開発につながる公共財のマネジメントの事例をいくつか説明しよう。

### （1）一般的な公共財のマネジメント

これまでの一般的な公共財の多くは，住民に対する税や，町内会費のように利害関係者の間で広く薄く負担を分担し，公的機関が整備を行っている。道路網の整備や，マンションの管理など，広く見られることである。これは，利害関係者すべてに広く共有される財のマネジメントを利害関係者の自主性に任せていたのでは，「誰か他の人がやってくれればいい」と考えるフリーライダーの出現によって，当初は自主的な貢献をしていたものの動機付けさえ失わせることがあり，整備が困難になるからである。さらに，有形の公共財の場合，誰かが使用，あるいは占有してしまうと他者が利用できなくなる場合が多く，フリーライダーの出現は貢献意欲をそぐことになるので致命的である。また，なかには私的な利益を優先して自分に都合よく整備しようとするものが出現することも考えられる。人は自己の利益のためには積極的に行動を起こすが，利他的な行動には消極的になりがちであり，両者が並立する場合には前者を優先しが

2）ソフトウェアは一般に開発者が著作者としての権利を主張し，ビジネスの対象として商品化される。オープンソース・ソフトウェアを公共財として扱うことができるのは，GPL（General Public License：一般公用ライセンス）などのライセンスによって，オープンソース・ソフトウェア定義に示されるようなソフトウェアの自由を確保しているからである。

ちである。そこで，中立性を高め，公正に公共財を整備していくため，相応の負担を等しく求め，これを原資として活動する中立性の高い機関が必要となる。

一方で，うまくいっている地域コミュニティでは，道路の清掃や，祭りの開催などが自発的に組織されている。そこでは，住民相互の信頼と，相互の評価を重視するというコミュニティ活動の原理がうまく機能している。

（2）標準形成の事例

道路などの有形財については，税などによる解決が一般的であるが，無形財すなわち使用しても価値を減ずることなく，価値を増すことさえ可能な財については，違ったかたちのマネジメントが可能になる。その一つの例が，各界の代表者を集め協議などによってさまざまな標準化を行うプロセスである。このような無形財はある種の知識と見なすことができる。

技術や標準は，一企業が設定し占有することで戦略的な優位性を獲得する場合もある。しかし，たとえば，ネットワーク接続や，情報システムの開発プロセスなど，広く共有することによってはじめて意味をなす場合が多い。このような標準の形成にあたっては，大きく二つのプロセスが想定される。一つは，利害関係者の協議によって形成するもので，もう一つは，誰かが単独で作り出したものを広く共有することによるものである。

前者の事例は，業界団体や各種標準制定機関で一般的に行われているプロセスである。税による公共財の整備と同じように，各利害関係者から広く代表を参加させることで標準の形成が進められる。しかし，必ずしも代表として参加したものたちが中立な立場から協議を行うという保証はなく，出身母体の利益を代表して自己利益につながるような標準へ誘導することもある。この駆け引きが適当なところに落ち着かなければ，標準形成に至らなかったり，できあがった標準は一般的な内容に終始してしまったりすることもあり得る。たとえば，ネットワーク接続に関する標準では ISO（国際標準化機構）が OSI（開放型システム間相互接続）という参照モデルを標準化した。一方で，その制定の間に DEC，Intel，Xerox が共同開発したイーサネットが市場を席巻し，これを基本として制定された IEEE 802.x 規格が OSI 参照モデルに準拠しているが，必ずしも一対一に対応はしない形で整備されて事実上の標準となっている。

後者の事例には，ビクターの開発した VHS ビデオレコーダー規格をあげることができる。ビクターは独自にビデオレコーダーシステムとして VHS を開発したが，これを同業者に公開し共同で VHS を育てていくことを提案した。一方で，先行してベータマックスを市場に投入したSONYは,独自の技術を占有し,同業者にはライセンスによって技術の提供を行っていた。VHS は，同業者がフロントローディング機能，倍速再生機能などさまざまな技術を VHS に付加し，その技術を相互に交換することで，高機能化を果たし結果的にビデオレコーダー市場を

世界的に占有するに至っている[3)]。

ビクターが VHS の技術を占有していたら，今のような反映はなかっただろう。参加者が企業ではあるが，技術（知識）を公開（オープン）することによって，新たな技術（知識）を獲得し，蓄積していった VHS はオープンソース・ソフトウェアとして開発された Linux にたとえることができるだろう。一方，技術を占有して囲い込んだ SONY はソフトウェアのライセンスを厳しく管理しているマイクロソフトにたとえることができるかもしれない。ビクターで VHS 開発の中心にいて，技術の公開を強く押し進めた高野鎭雄は，Linux における Linus Torvalds といってもよいだろう。（図１）

ここで注意しなければならないのは，ビクターおよび VHS グループに参加した各社が，必ずしもビデオの統一規格を作るという「社会のため」の利他的な行動として VHS を進めたとはいえないところである。ビクターにとっては，業界内の協力なくしては VHS を世に出すことはできず，開発経費を無駄にしてしまうことになった。参加した同業者にしても，VHS か，ベータマックスか，という選択の中での自己利益追求のため行動であった。すなわち，それぞ

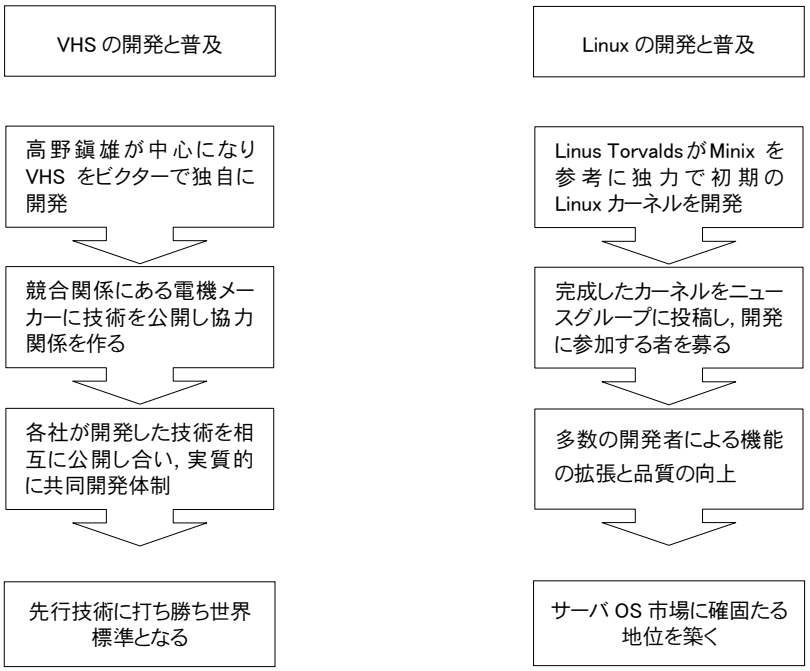


図１　VHS と Linux の開発と普及

3）VHS が同業者の共同によって機能強化されていく経緯は，佐藤正明，『映像メディアの世紀---ビデオ・男たちの産業史』，日経ＢＰ，1999 に詳しい。

れの組織の持つさまざまな自己利益追求のための行動の集合が結果として成果を上げたことになる。オープンソース・ソフトウェアもまさにこのような多種多様な自己利益を追求する個人の行動の集合で形成されている。

（3）オープンソース・ソフトウェアの場合

オープンソース・ソフトウェアの開発は，プロジェクトを開始するものが，個人あるいはグループでソフトウェアの原型を提供し，これをもとに機能追加や改善を行うことで進められる。このとき，プロジェクトの周囲にさまざまなかたちで貢献をしようとするものたちが集まり，協働が行われる。開発プロジェクトを開始した者がリーダーになっていることが多いが，そのプロジェクトへの参加は基本的に自発的なものであり，権威に基づいた統制は行われていない。参加の動機には，Linux プロジェクトを始め，現在でも中心的な役割を果たしている Torvalds が，「それがぼくには楽しかったから」[4] といっているように，知的好奇心を満足させるものもいれば，ソフトウェア開発能力の向上，自分の評判を高める，自分が作りたいソフトウェア，そして，コミュニティでの評価を高めたい，などが参加動機であると答えるものもいる [5]。評判を高めるためには，参加するプロジェクトが有名である必要があるが，実際のところ進行中のオープンソース・ソフトウェア開発プロジェクトは，数万ある [6]。ほとんどは，無名のプロジェクトである。したがって，多くのプロジェクト参加者たちは，自らの使いたいソフトウェアが世の中にはないのだが，自分一人で作るのは大変だ。というような考えで参加していることになる。つまり，このような動機付けを中心に，知的好奇心の追求までを含め，参加者は社会のためにソフトウェアを開発しているのではなく，参加者個人の自己利益を追求するためにプロジェクトに参加している [7]。

---

4）Linus Torvalds and David Diamond, Just for Fun: The Story of an Accidental Revolutionary, Harpercollins, 2001.（風見潤訳：リーナス トーバルズ，デビッド ダイヤモンド，『それがぼくには楽しかったから』，小学館プロダクション，2001.）

5）オープンソースプロジェクトへの参加動機を調査したものとしては，OSDN（Open Source Technology Group と Boston Consulting Group が 2002 年に報告した Hacker Survey（http://www.ostg.com/bcg/）や，International Institute of Infonomics が 2002 年に実施した Free/Libre and Open Source Software:Survey and Study（http://www.infonomics.nl/FLOSS/report/）などがある。

6）オープンソース・ソフトウェアの開発情報を管理する http://sourceforge.net に登録されているプロジェクト数は 2005 年 1 月 18 日現在で 93,863。この中には，すでに終了しているプロジェクトもあるが，このほかに，日本など地域を限定してローカルに開発されているものも多数ある。

7）オープンソース・ソフトウェア開発者の動機付けについては，http://www.rieti.go.jp/it/column/column030604. html にある佐藤一郎の「オープンソースの理想と現実」（2003 年 6 月 4 日）と，これを受けての http://japan.linux.com/opensource/03/06/28/1235243.shtml と http://japan.linux.com/opensource/03/07/31/190257.shtml にある八田真行の「オープンソースの現実，と若干の理想（上）（下）」（2003 年 6 月 28 日と 8 月 1 日）が，基本的に自己利益の追求が動機付けであり，それがオープンソース・ソフトウェアの開発を特徴づけていることを論じている。このために，オープンソース・ソフトウェアとしては，開発者が自分の使うソフトウェ

（次頁に続く）

コミュニティ活動への参加を通じた人間関係の構築は，動機付けの第一にはなっていないものの，コミュニティ活動の原理をうまく機能させるための仕組みが開発プロセスに組み込まれている。

オープンソース・ソフトウェアの開発は，その本質的プロセスから開発に関わるほとんどのコミュニケーションがメーリングリストを通じて，オープンに行われている。この中には，開発の方向性を示唆するようなメッセージや，開発の計画，提出されたソースコードの中からの選別決定などが含まれる。

開発の方向性を頻繁に広く均一に流通することは，コミュニティ全体に価値を浸透することにつながる。また，選別決定をオープンに告知することも，評価の妥当性をオープンにするだけでなく，繰り返し決定にふれることで，開発コミュニティの価値や文化をコミュニティ全体に浸透することにつながる。自己利益の追求を志向しているものの自発的参加であるから，このような価値や文化に大きな違和感を持つものは，コミュニティから離れていく。

以上のようなプロセスで，オープンソース・ソフトウェアの開発コミュニティは，健全なコミュニティを形成している。しかし，一般に健全なコミュニティであっても，フリーライダーの出現によって，自主的な参加が破綻を来すことはよくある。これは，コミュニティの維持する公共財がフリーライダーによって占有されたり，消費され尽くしたりすることによって，コミュニティ参加者の動機が維持できなくなるからである。しかし，オープンソース・ソフトウェアは公共財であるが，複製可能な無形財であり，フリーライダーとしての利用者が増えて消費が繰り返されたとしても，それによってソフトウェアの価値を減じることはなく，結果的にそのソフトウェアの知名度や社会における重要度を増すことによって，コミュニティの価値を向上することさえあるからである。

一般のコミュニティは，コミュニティの規模を維持，あるいは拡大するために，新規の参加者を勧誘するプロセスを持っているが，成功しているオープンソース・ソフトウェアの開発コミュニティには，そのようなプロセスはなく，自発的な参加者を継続的に獲得するによってコミュニティの規模を維持あるいは，拡大している [8)]。参加者にとっての魅力を維持できなければ，開発プロジェクトとその開発コミュニティは必然的に消滅することになる。

現在，ソフトウェアは私企業が所有し，排他的に利用されることが多いが，オープンソース・ソフトウェアは公共財として共有されている。知的財産は一定の範囲でその権利が保護される

---

アの開発は成功しやすいが，そうではないソフトウェアについては，成功しないことが多いと論じている。

8）コミュニティのとしてのオープンソース・ソフトウェアの開発については，Eric von Hippel and Georg von Krogh, Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science, Organization Science, Vol.14 No.2 March-April 2003 pp.209-223.など

べきであるが，保護のしすぎは社会全体の知的発展の妨げになるという議論がある[9]。公共財としてのオープンソース・ソフトウェアは，社会全体のソフトウェア開発能力の向上に貢献する側面を持っている。

## 2. メタ知識としてのソフトウェア開発能力

ソフトウェアを知識としてとらえられるとすれば，ソフトウェアの開発能力は，その知識を作り出すための知識であるからメタ知識であるということができる。オープンソース・ソフトウェアの開発コミュニティは，成果物としてのソフトウェアを生産するだけでなく，そのプロセスの中で参加者のソフトウェア開発能力を向上させている。オープンソース・ソフトウェア開発コミュニティへの参加動機にも自己の能力向上があげられている一方で，開発コミュニティの方でも，コミュニティの規模を維持あるいは拡大して開発の戦力を確保するためには，ソフトウェア開発能力をすでに十分持っているものの参加を待つだけでなく，コミュニティの中で自発的にではあるが，開発能力の向上を促していかなければならない。

### （1）ソフトウェア開発能力の向上

ソフトウェア開発能力を習得するためには，まず，開発に利用するプログラミング言語の知識を習得し，自在に使えるようになることが必須である。しかし，これは最低限必要とされる条件であって，もちろんそれだけでは不十分である。プログラミング言語を駆使して，プログラムを作成するためのさまざまな技法を身につけなければならない。

よい文章を書くためには，その文章を書くために使用する言語の語彙や文法を習得するだけでは不十分であり，ものを書く経験を積むことと，よい文章に多くふれることが重要であるように，ソフトウェアの開発能力を向上してよいソフトウェアがかけるようになるためには，多くのプログラムを実際に作製して開発の経験を積むと共に，優れたソフトウェアのソースコードに多くふれることが必要不可欠である。このような経験を通じて，自ら優れたプログラミング・スタイルを身につけることでソフトウェア開発能力を習得していくのである。また，優れたソフトウェア開発者の指導を直接受けることによって，その習得の速度を加速できるのはいうまでもない。

### （2）オープンソース・ソフトウェア開発コミュニティの場合

9）Lawrence Lessig, the future of idea, Random House, 2001.（山形浩生訳：ローレンス・レッシグ，『コモンズ』，翔泳社，2002.）

オープンソース・ソフトウェアは，Richard Stallman が提唱したフリーソフトウェア[10]を思想的な背景に持っている。オープンソース・ソフトウェアはフリーソフトウェア同様に，ソースコードを研究し，学習のために利用する自由を確保している。これは，Stallman が，ソフトウェアは人類共有の知識であり，既存のソフトウェアを新たな知識（ソフトウェア）創造のためのステップとして活用すべきと考えたからである[11]。

オープンソース・ソフトウェアに採用されるのは，世界中にいる多数の開発者たちから寄せられたソースコードの中から選別された優れたソースコードであり，このようなソースコードに日常的に触れることで，ソフトウェア開発能力を向上することが期待できる。オープンソース・ソフトウェアでは，このようなソフトウェア開発能力向上のための学習の目的でソースコードの活用が推奨されている。これは，オープンソース・ソフトウェアの開発コミュニティの中核に組み込まれたものでなくても，誰でも享受できることである。

また，遠隔で進められるプロジェクトの中で，優れたソフトウェア開発者の指導を直接に受けることはできないとしても，開発のプロセスはほとんどインターネット上のメーリングリストを通じたディスカッションによって進められるので，このディスカッションの経過は第三者として観察するだけでも，生々しいソフトウェア開発の実際を目の当たりにすることができるだろう。さらに，世界中から寄せられたソースコードの中から中核開発者を中心として行われるソースコードの選別プロセスを観察することによっても，よいソフトウェアとはどのようなものか，という基準を身につけることができるだろう。このようなかたちで，優れたソフトウェア開発者たちから間接的にではあるが，指導を受けることが可能である。

オープンソース・ソフトウェアの開発プロセスでは，プロジェクトが地理的に分散された環境の中で進められるので，すべてのコミュニケーションはインターネット上のメーリングリストなどを通じて行われる。結果として，原則的に開発プロセスに関連するコミュニケーションは開示する仕組みとなっている。したがって，オープンソース・ソフトウェアの開発コミュニティの内部，さらには周辺においても，個人が自発的努力によってソフトウェア開発能力を向上する機会を提供している。これは，結果として，開発コミュニティの周辺にいる参加者たちを育成し，開発コミュニティの中心へと人材を誘導するプロセスの可能性を示している。

また，開発に関連する過去のコミュニケーションも，メーリングリストのログとしてインタ

---

10）フリーソフトウェア（Free Software）の Free は，Free of Charge の Free ではなく，Freedom の Free である。Stallman は，ソフトウェアの使用，ソースコードの解析・探求，改変，再配布の自由を求めている。

11）優れたソフトウェアを元にして，よりすぐれたソフトウェアを自由に作り出していこうという考え方は，学術研究における研究論文と同じである。既存の優れた研究を引用したり組み合わせたりすることによって，新たな知としての研究論文を作り出すことができる。論文は一定の範囲内で引用が許されているのは，知を元に新たな知を作り出すためである。
よい論文を書くためには，よい論文にたくさんふれる必要がある点も似ているだろう。

ーネット上に残っている。これを活用すれば，遡及的に過去の開発プロセスを擬似的に体験することができる。これによって，途中から開発コミュニティに参加する者は，コミュニティの価値と文化を確認することができる。新規参加者に対するオリエンテーションのような手間をかけずとも，開発コミュニティへの参加に価値を見いだすものは，このような仕組みの中で既に進行中のプロジェクトの途中からでも，自発的に参加を開始することができるようになっている[12)]。

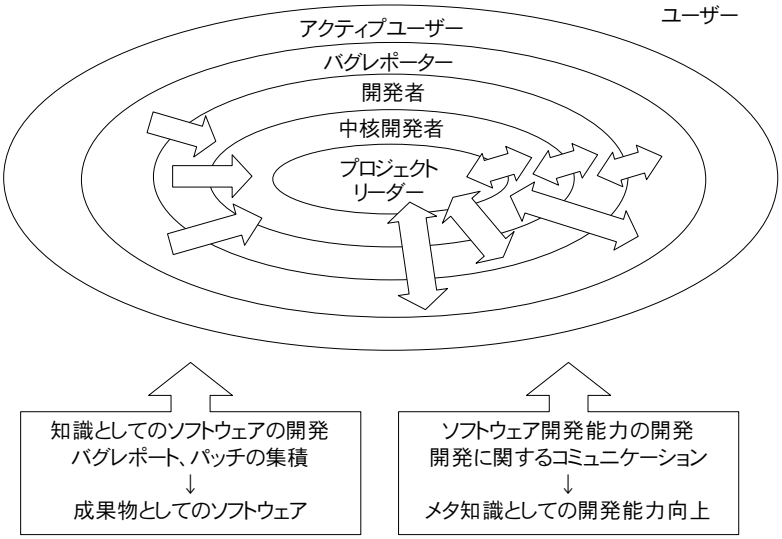


図2　オープンソース・ソフトウェア開発コミュニティにおける知識創造

## 3. オープンソース・ソフトウェア開発における知識創造

これまでに見てきたように，うまく機能しているオープンソース・ソフトウェアの開発コミュニティでは，知識としてのソフトウェアの質を向上するとともに，その知識を作り出す能力としてのメタ知識としてソフトウェア開発能力の向上が実現されている。ここでは，これらのプロセスを整理した（図2）上で，実践的な知識創造のための仕組みとして注目されている実践コミュニティの枠組みで分析を試みる。

12）ディスカッションのログを公開することによって，地理的に分散した作業を効率的に進めるプロセスについては，竹田昌弘「ネットワーク学習による意思決定過程の革新」,『学習する組織』同文舘，1993.でも，事例研究として整理している。

（1）ソフトウェア（知識）の創造プロセス

ソフトウェアは，ハードウェア上でのデータ処理の手順を記述したものであるから，ある種の知識であるといえる。したがって，ソフトウェア開発プロセスは知識創造のプロセスと考えることができる。

完成するまで実体のないソフトウェアの開発プロセスでは，開発者相互の密なコミュニケーションが高品質な作業を進めるために必須である。にもかかわらず，オープンソース・ソフトウェアの開発コミュニティは，通常は大きな困難が伴う，地理的に分散した環境にある開発者たちがネットワークを通じたコミュニケーションのみによる共同作業によって，市場に流通する商用ソフトウェアに匹敵する機能と品質を持つソフトウェアの開発に成功している。

その背景には，企業組織で行われるソフトウェア開発とは異なったコミュニティ活動を基本とした開発プロセスがある。参加者間の相互信頼と相互評価の仕組みが機能している。その結果，開発者は自己の利益追求に動機付けされているとはいえ，基本的には無償で自発的にコミュニティに参加し，ソフトウェア開発活動に関わる。企業組織におけるソフトウェア開発は，コストを重視しなければならないため，最適化された人員とスケジュールで遂行される。一方で，オープンソース・ソフトウェアの開発では，中核的な開発者から，周辺的な開発者，開発者候補の利用者など，開発への関わりの程度に差はあるものの，全体としては過大な数の開発者が参加する。そこで，複数の開発者が同じ部分の開発を並行して進め，その中からプロジェクトの方向性に適したソースコードを選択するという，非常に冗長性の高い開発が可能となり，結果的に高品質のソフトウェア開発につながっている[13)]。

中核の開発者たちはそれぞれの担当するモジュール[14)]に対して責任を持っているものの，それ以外の開発者については，誰かが役割分担を指示するわけではないが，自分のソフトウェア開発能力に応じて，ソフトウェアの瑕疵であるバグを発見して開発グループに報告するバグレポーター，報告されたバグに対応して修正プログラムとしてパッチを作成する開発者などとして行動する。それらの作業の成果は，開発コミュニティの内側に向かって提出されるが，すべてが採用されるわけではなく，質の高いものだけが採用され成果物としてのソフトウェアに反映され，結果としてソフトウェアを高品質に維持し続けていく。（図 2 の左側）

---

13）オープンソース・ソフトウェアの開発プロセスについては，竹田昌弘，「オープンソース・ソフトウェアの開発プロセスに関する考察」，『立命館経営学』，第 43 巻第 4 号，2004.11.に整理した。

14）機械製品などと同様にソフトウェアの開発においても，ソフトウェア全体を一括して開発するのではなく，比較的関連性の低い機能ごとにモジュールという単位に分割して開発作業を進める。機械製品におけるモジュールが生産工程の都合によって分割されることが多いのに対して，ソフトウェア開発においては，開発グループを分割して開発作業の分担を容易にするために行われる。特に，インターネットを通じたコミュニケーションしかできないオープンソース・ソフトウェアの開発においては，適切なモジュール分割ができるかどうかがプロジェクトの成否にも大きく影響するといっていいだろう。

（2）ソフトウェア開発能力（メタ知識）の創造プロセス

オープンソース・ソフトウェアの開発コミュニティは，このようなプロセスを通じて成果としてソフトウェアという知識を創造しているが，それだけでなく，そのプロセスを通じて，コミュニティの内部と周辺でソフトウェア開発能力というメタ知識の学習を促進している。

すでに述べたように，ソフトウェア開発能力の向上はいくつかのプロセスによって可能になっている。第一に公開され，流通されている最新の優れたソースコード自身を探求，学習することを通じて，よいサンプルにふれることで，よいプログラミング・スタイルを習得することによって，そして，第二に開発に関連して開発者間で交わされるコミュニケーションを通じて，よいソフトウェア開発者の考え方や，判断に直接的・間接的にふれることによって，ソフトウェア開発能力の向上が促進される。ソフトウェア開発能力のように形式知化が困難な知識の移転においては，OJT などの経験を通じた学習が有効である。オープンソース・ソフトウェアの開発コミュニティでは，まさに，実際のソフトウェア開発プロセスの周辺にインターネット上のコミュニケーションを通じて参加し，実地に，あるいは仮想的に体験を重ねることによってソフトウェア開発能力の移転と向上が実現されている。

開発者相互の間でのコミュニケーションは形式化された電子メールのメッセージが中心であるが，そのコミュニケーション・パターンを繰り返し目にすることによって，ソフトウェアに関する目利きのようなことができるようになり，結果として，よいソフトウェアが開発できるようになる。（図 2 の右側）

以上のように，オープンソース・ソフトウェアの開発コミュニティは，高品質の知識としてのソフトウェアを作り出しているだけでなく，さらに知識の高品質化につながる知識創造能力をコミュニティの周辺に蓄積し，二重のループによる知識創造プロセスを実現していることがわかる。

（3）実践コミュニティとしてのオープンソース・ソフトウェア開発コミュニティ

Wenger らは，組織の中で実践的に知識を共有し，また創造するための有効な仕組みとして実践コミュニティ（Communities of Practice）を提案している [15]。その基本的な考え方は，組織における知識の有効活用には，「人と人との関係づくりが必要だ」ということである。熱意を持って自発的に参加した組織内コミュニティにおいて，参加者の持つ知識の共有とさらには新た

15）Etienne Wenger，Richard McDermott，William M. Snyder，Cultivating Communities ｵf Practice，Harvard Business School Press，2002. .（櫻井祐子 訳：エティエンヌ・ウェンガー，リチャード・マクダーモット，ウィリアム・M・スナイダー，『コミュニティ・オブ・プラクティス』，翔泳社，2002.）など

な知識の創造が有効に行われる。この実践コミュニティを組織活動に対して有効に貢献させるためには，実践コミュニティを構成しやすくするような制度や文化を組織内に構成するほか，組織の戦略，目的，目標と実践コミュニティの目的を整合させる必要がある。また，地理的に分散した実践コミュニティにおけるコミュニケーションの問題や，組織の境界を越えた実践コミュニティの可能性についても示唆している。

オープンソース・ソフトウェアの開発コミュニティは，組織を超えて個人が参加することで構成される実践コミュニティとみなすことができる。

オープンソース・ソフトウェアの開発コミュニティは，所属する組織を持たないものの，自発的な参加者の熱意に基づいて，参加者共通の目的にしたがって，成果としての知識（ソフトウェア）を開発していく。成果物を作り出すという意味では，プロジェクトなのかもしれないが，プロジェクトは当初から計画された寿命を持つ，という意味で，オープンソース・ソフトウェアの開発はコミュニティによって行われていると見るのが妥当である。一般に，コミュニティは，参加者が存在する限り存続し，オープンソース・ソフトウェアの開発も，そのように進められる。

オープンソース・ソフトウェアの開発コミュニティが明確な境界を持たず，中核的なメンバーを中心に，それぞれの熱意と事情に応じて参加の度合いを選択できるところも実践コミュニティの特徴である。階層化されたコミュニティの構造は，メンバーの中心に向かっての育成を示唆している。実践コミュニティを機能させるためには，メンバーを継続的に勧誘するための仕組みも強調されているが，オープンソース・ソフトウェアの開発コミュニティではこの仕組みなしに，メンバーの補充が行われている。これは，メンバーの母体層が固有の組織に限定されずに，無尽蔵であるからだろう。また，もしメンバーの補充がなくコミュニティが消滅したとしても，それを関知する組織もない。

以上のように，細部に置いては，Wenger らの提案する実践コミュニティには当てはまらない部分もあるが，オープンソース・ソフトウェアの開発コミュニティは知識を流通し，さらに創造する仕組みとして，実践コミュニティの一形態と見なすことができる。

## 4. 企業組織へのインプリケーション

ナレッジ・マネジメントの本質は，組織内の知識を移転したり，共有したりすることで有効に活用し，さらに新たな知識を創造していくことであるが，知識を組織内の公共財として考えたとき，企業組織内の公共財を充実させるためには，一般の公共財の場合と同じように，公的機関による整備とコミュニティ活動の充実による方法とが考えられる。これらの仕組みなしに，通常の企業組織の活動モードだけでは，組織的な知識の活用と創造はなかなか進まない。

公的機関に相当する機能としては，知識の流通・創造を積極的に行う専任の担当者，担当部

署を設定することができる。この機能は，組織にとってオーバーヘッドとなるが，企業の目指す方向での知識の流通・創造を進めることができる。また，投入に相当する成果も期待できる。知識の活用と創造は有効に進むが，一方で，組織構成員のメタ知識を充実することは困難かもしれない。

オープンソース・ソフトウェアの開発コミュニティのように，実践コミュニティの活動を充実することで，互いの信頼の中で自発的に知識の流通・創造活動への参加を促すこともできる。自発性に依存するために，期待される成果は不確定であるが，アドホックに現場のニーズに応える知識創造活動が期待できる。また，メタ知識の充実も促進されると考えられる。Wenger が指摘するような制度や文化を整備することによって，実践コミュニティが自発的に構成され成果を上げる組織を作り出すことも可能である。

たとえば，主に製紙に関連したエンジニアリング企業であるバックマン・ラボラトリーズでは，オンラインのフォーラムと呼ばれる電子掲示板上でのディスカッションを通じて，世界中に分散する従業員の間で，知識の流通と創造が有効に行われている。そこでは，同僚を助けることに価値を置き，また，問題解決を楽しむ文化を共有すると共に，その活動が公式に評価される仕組みを整備することで，地理的に分散された従業員から構成されるコミュニティ的な活動が促進されている[16)]。

また，繊維素材のゴアテックスで有名な W. L. ゴア・アンド・アソシエイツでは，その社名の示すとおり社長のゴア以外はすべての従業員はアソシエイトとして，仲間同士の立場でプロジェクトを作り出し仕事をしている[17)]。そこでは，その文化を維持するために，採用にあたって重点的な面接を行い，価値を共有し共に仕事ができる人材であるかどうかを中心に判断されている。また，採用時に特定の所属部門を決めないことも，コミュニティ的に共同作業を可能にすることにつながっている。

企業組織におけるナレッジ・マネジメントの実践は，知識の収集，活用を目的とした担当部署や専門システムの導入によって進められることが多かったがその成果は必ずしも期待通りのものではなかった。そのような状況の中で Wenger らは，人と人とのつながりによって知識を交換し，また創造する場が構成されることを指摘した。オープンソース・ソフトウェアの開発コミュニティは，誰にでも見えるところで活動しているので，Wenger らの実践コミュニティのモデルに必ずしも一致するものではないが，企業組織が実践コミュニティを導入しようとする場合に，わかりやすい事例のひとつとして参照することができる。

---

16）水越伸 NHK「変革の世紀」プロジェクト編，『NHK スペシャル 変革の世紀①市民・組織・英知』，NHK 出版，2002，119-125 ページ。

17）http://www.gore.com/en_xx/aboutus/culture/index.html

## ま と め

本稿では，オープンソース・ソフトウェアの開発コミュニティにおけるソフトウェア開発を公共財としての知識を創造するプロセスとして整理した。その上で，公共財としての知識創造を企業組織に適用する可能性を探った。

オープンソース・ソフトウェアの開発コミュニティの参加者はほとんどが，本業として何らかの企業においてソフトウェアの開発に携わっている。彼らは，余暇の時間を使って，本業では実現できない目的を達成するために，オープンソース・ソフトウェアの開発に参加している。その意味では，オープンソース・ソフトウェアは，開発コミュニティ参加者の勤務先企業に依存していると見ることができる。しかし，本稿で例示した標準制定の事例とは異なり，所属組織の利益を代表したかたちでの活動は行われていない。

最近では，IBM などの企業が，競争優位獲得のための戦略として従業員をオープンソース・ソフトウェアの開発に職務として投入し始めている。本来の職務とは，独立にオープンソース・ソフトウェアの開発に携わっているものたちの動機付けがこのような状況で変化するのか否かは，現在進行中の事態ではあるが，今後のオープンソース・ソフトウェアの開発を考える上で非常に興味深い。

オープンソース・ソフトウェアの開発を巡っては，これ以外にも研究に値する興味深いプロセスが散見する。今後は，本稿で取り上げた以外のプロセスも整理し，企業経営へのインプリケーションをさらに探っていきたいと考えている。

### 参考文献

八田真行「オープンソースの現実，と若干の理想（上）」（2003 年 6 月 28 日）http://japan.linux.com/opensource/03/06/28/1235243.shtml

八田真行「オープンソースの現実，と若干の理想（下）」（2003 年 8 月 1 日）http://japan.linux.com/opensource/03/07/31/190257.shtml

Eric von Hippel and Georg von Krogh, Open Source Software and the “ Private-Collective ” Innovation Model : Issues for Organization Science, *Organization Science*, Vol.14 No.2 March-April 2003

Lawrence Lessig, the future of idea, Random House, 2001.（山形浩生訳：ローレンス・レッシグ,『コモンズ』，翔泳社，2002.）

水越伸 NHK「変革の世紀」プロジェクト編,『NHK スペシャル 変革の世紀① 市民・組織・英知』, NHK 出版，2002.

佐藤一郎「オープンソースの理想と現実」（2003 年 6 月 4 日）http://www.rieti.go.jp/it/column/column030604.html

佐藤正明,『映像メディアの世紀---ビデオ・男たちの産業史』，日経 BP，1999

竹田昌弘,「オープンソース・ソフトウェアの開発プロセスに関する考察」,『立命館経営学』， 第 43 巻第 4 号，2004.11.

Linus Torvalds and David Diamond, Just for Fun: The Story of an Accidental Revolutionary, Harpercollins, 2001.（風見潤訳：リーナス トーバルズ，デビッド ダイヤモンド,『それがぼくには楽しかったから』，小学館プロダクション，2001.）

Etienne Wenger，Richard McDermott，William M. Snyder，Cultivating Communities of Practice, Harvard Business School Press，2002.（櫻井祐子訳：エティエンヌ・ウェンガー，リチャード・マクダーモット，ウィリアム・M・スナイダー,『コミュニティ・オブ・プラクティス』，翔泳社，2002.）

http://www.gore.com/en_xx/aboutus/culture/index.html

http://www.infonomics.nl/FLOSS/report/

http://www.ostg.com/bcg/

http://sourceforge.net

（URL はすべて，2005 年 1 月 18 日にアクセスしたものである）