

開発プロセスから見たオープンソースの優位性と社会的影響力 の検討（オープンソース教育の知識領域として）

坊垣貴夫[†] 國井利泰^{†*}

近年オープンソース教育が高まりを見せている。オープンソース教育においては利用技術、開発技術の方法論を学ぶだけでなく、オープンソースの力の源泉について理解し、その社会に及ぼす影響をも理解する必要がある。

オープンソースの力の源泉はその開発プロセスそのものにある。ソースコードを特定の誰かが独占的に所有することなく共有・公開し、誰でも自由に開発に参加できる開いた開発コミュニティを作ること、規模、多様性ともに、独占的ソフトウェアの閉じた開発チームを凌駕する。このプロセスはサイバーワールド内の発展原則に従っているため、オープンソースのサイバーワールド内での重要性はますます高くなり、ひいては現実の社会においても重要性が増す。

Domination of open source based on development process and social influence (As a knowledge area of education)

Takao Bogaki[†] Toshiyasu L. Kuni^{†*}

In the education of open source, it is necessary to understand why open source has great power, and also its influence on the society. Open source communities are open, and everyone can participate in open source development freely. Nobody monopolizes source codes, and they are shared and opened to the public. As a result, open source communities surpass closed teams that develop proprietary software, in both scale and diversity. This process follows the principles of progress of cyberworlds, and hence the importance of open source in the cyberworlds rises more and more. Consequently, its importance increases in the society of the real world.

[†]金沢工業大学大学院工学研究科
Graduate School of Engineering, Kanazawa Institute of Technology
^{*}金沢工業大学 IT 研究所
IT Institute, Kanazawa Institute of Technology

1. はじめに

近年オープンソース教育が高まりを見せている。その主な対象領域は、オープンソース・プロダクトをコンピュータ・システム構築の際の部品として使いこなす利用技術の向上、必要なカスタマイズを施す開発技術の向上である。

IEEE-CS、ACM が合同で策定したコンピュータ教育カリキュラムガイドライン CC2001 では、修得すべき知識領域として「社会的、倫理的、および専門的諸問題」を独立した項目として挙げており、従来よりもこうした知識領域を意識的に学習する必要があると主張している[5]。

こうした観点に立てば、オープンソース教育においては単に方法を学ぶだけでなく、ファンダメンタルとしてオープンソースの重要性の高まりが社会に対して持つ影響力を理解することも必要となる。

本論文では[1]で示されたビジョンをもとに、オープンソースの発展が、ネットワークでつながったコンピュータ世界（以後サイバーワールドと呼ぶ）の中での発展圧力の枠組みの中でどのように理解されるのかを検討し、オープンソースが社会に及ぼす影響力について検討する。

2. オープンソース・プロジェクトのライフサイクル

2.1 開発技術教育

本論文ではオープンソースの社会的影響力について検討しようとしているが、その前にまずオープンソースのライフサイクルを概観し、その中の開発プロセスについて整理しておかなければならない。オープンソースの社会的影響力

の主要な源泉は開発プロセスそのものにあるからである。

オープンソースの開発プロセスを理解することは、もう少し実践的な観点からも重要である。オープンソースは単に無料で使えるソフトウェア・プロダクトであると捉えるのは適切でない。そのような観点に立てば、オープンソース教育における開発技術教育には、オープンソースを共有財産として維持・発展させるための再生産ループの形成を確実なものにするという使命もあると考える。すなわち、みずからオープンソース開発に参画したり、創始したりできる人材の育成を図らなければならない。

そのためには個々の要素技術もさることながら、開発プロセスの理解も重要である。オープンソースのプロジェクトは商用の独占的ソフトウェアとは異なるライフサイクルを取る。最初のプロダクトを一般に公開した後開発者コミュニティが形成され、独占的ソフトウェアの閉じた開発プロセスとはまったく異なる経緯をたどる点は、特にオープンソース開発者を目指す者としてはしっかり理解しておかなければならない。

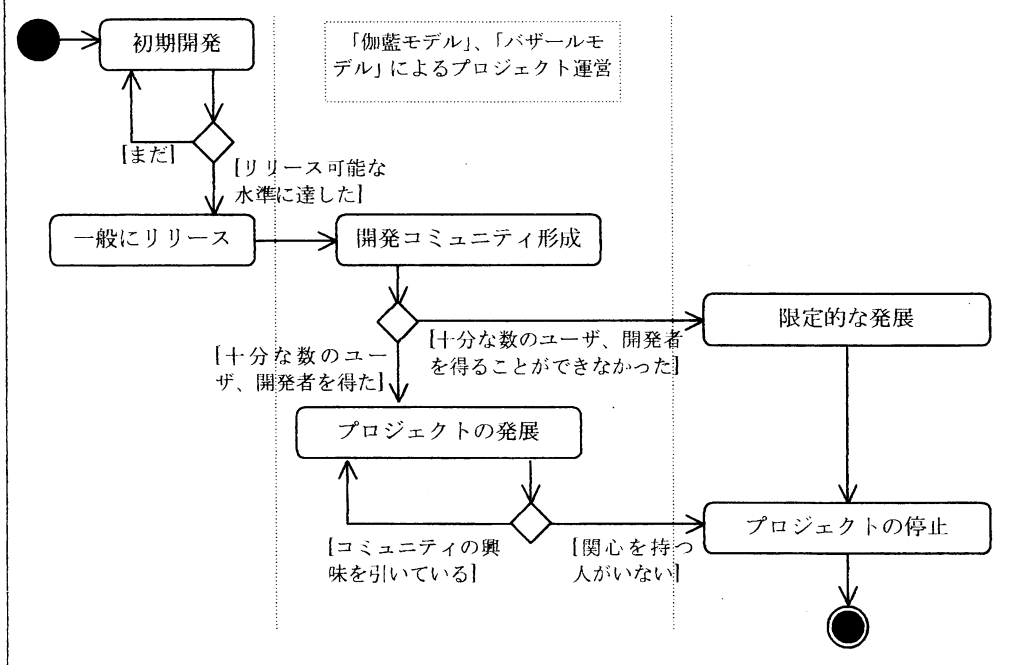
2.2 オープンソースのライフサイクル概観

まず、オープンソースのライフサイクルを概観しておく（図 1）。

(1) 開発初期

オープンソース・プロジェクトを開始するには、まずある程度の機能と品質を持った動作可能なプロダクトを作成する必要がある。動作可能な具体的なソフトウェアが何もなかったり、極端に品質が低かったりする状態では開発コミュニ

図 1 オープンソース・プロジェクトのライフサイクル模式図



ニティを形成し、プロジェクトの発展段階に入ることにはできない ([3]、[6])。

開発を開始する主体は

- 個人的な必要を満たそうとする個人
- 基盤構築を企図する大学内プロジェクト
- ある戦略をもってオープンソース化を予定した企業によるプロジェクトである。最初のプロダクトは新規に作り出す場合や、既存のオープンソースで開発者が興味を失って保守されなくなったプロダクトを譲り受けて新しくプロジェクトを起こす場合もある。

(2) リリース、開発コミュニティ形成、プロジェクトの発展

初期開発の後、プロダクトをリリースし開発コミュニティを形成してプロジェクトの発展が始まる(十分な規模のコミュニティが作れず、その後の発展が限

定的な場合もある)。図 1の中央部分がこの段階である。

開発コミュニティはいわゆる開発者以外にユーザも含まれる。

開発コミュニティの具体的な運営方法はプロジェクトによって様々だが、大きくは以下の 2 形態を取る[2]。

- 伽藍モデル：少数の開発者だけで開発し、一定の品質に達したところで次のバージョンをリリースする。
- バザールモデル：開発中のバージョンを品質にこだわらず次々とリリースし、なるべく多くの人を開発に関わらせる。

特にバザールモデルは独占的ソフトウェアとはまったく異なるプロジェクト運営形態であり重要である。

(3) プロジェクトの停止

開発者を含め、コミュニティがプロダ

クトへの必要や興味を失うとプロジェクトは停止する。独占的なソフトウェア・プロダクトではこれでソースコードが失われる可能性が高いが、オープンソースの場合、世界のどこかにソースのコピーが残っていることになる。この点も独占的なソフトウェアとの違いであるが、ここでは詳しく論じない。

2.3 開発プロセスへの注目

前節の(1)の段階である初期開発の過程は、個人の力量、厳格な開発プロセス、アジャイル開発プロセスといった、ソフトウェア工学の正統的な知識の範疇で扱うことができ、そこでのプロセスとしては独占的なソフトウェアと大差がない。

初期開発を終えて一般に公開した後の段階、前節の(2)の段階を以後「開発プロセス」と呼ぶことにする。オープンソースでは一般公開後の開発者コミュニティによる発展、特にバザールモデルに従う開発者コミュニティのたどるプロセスが特徴的である。このプロセスは正統的なソフトウェア工学の枠からはまったくはずれており、この点が独占的なソフトウェアとの最大の相違点となっている。

オープンソース教育においてはこの部分の方法論だけでなく、このプロセスそのものがオープンソースの力の源泉となっていることを理解することが重要であり、この後さらに検討していく。

3. オープンソースの開発プロセスと社会的影響力

3.1 独占的なソフトウェアの限界

ここでは、[1]のビジョンをもとに社

会におけるオープンソースの重要性の高まりが、ある必然性を持っていることを検討する。

[1]では、歴史的な大国が覇権を及ぼした範囲と覇権を保った期間、当時の主要な情報伝達手段を検討して、覇権の性質についての第一近似として以下の原則を置いている。

「覇権の及ぶ範囲は情報の伝達速度に比例し、覇権を保つ期間は反比例する。」

これにコンピュータネットワーク内での情報伝達速度を当てはめると、サイバーワールドでは覇権の及ぶ範囲は地球の表面積をはるかに凌ぎ、覇権を保つ期間はわずかに5分間であるという結果が示されている。

ソフトウェア作成を主な活動領域としている企業やグループにおいては、サイバーワールド内での覇権とは、ソフトウェア・プロダクトのシェアや独自の技術のシェアといったことに相当する。

実際にはソフトウェアの導入や変更にはかなりの手間を要することから、一度稼働を開始したソフトウェアは簡単には入れ替えることができない。また、ベンダーの自社のシェアを守る企業戦略が奏功するなどして、あるソフトウェアの市場占有（覇権）が現実には5分間で入れ替わるということはないが、その主役交代圧力は強烈である。

その圧力の源泉は、時代の要求の多様化、スピード化である。昨今のコンピュータプログラムの規模、複雑さの増大、バージョンアップの短期間化圧力の高まりを考えてもわかるとおりである。

このような状況下においては、特定の企業が所有する限られたリソースで対

応する独占的ソフトウェアは、要請に応じていくことが困難になっていく。

それに対して、後に述べるとおり、オープンソースでは必要に応じていくらかでも規模と多様性を増大させ得る開発者コミュニティを形成することができ、多くの分野で、要求の多様化・スピード化への対応力における優位性を増している。

現在のデスクトップPC分野での覇者である Microsoft は巨大であるとは言え、オープンソースの潜在的な開発者になり得る、世界中の技術者プールに比べると大きいとは言えない。上記の圧力がさらに高まれば状況は一気に変わりかねない。

3.2 オープンソースの開発プロセスの特徴

前述したとおり、オープンソースの開発プロセスは大きくふたつに分類される。ソースコードを公開するが、ソースの修正の権利を持つのは少人数の開発チームだけという「伽藍モデル」と、誰にでも修正の権利がある「バザールモデル」である。

「伽藍モデル」では実質的に開発に携わる人数が非常に限られているため、開発速度、多様性とも独占的ソフトウェアと同様の制限を受ける。その点「バザールモデル」ではそのような制限を受けない。

実際には「バザールモデル」も少数の中心的な開発者グループとその他という構造を取っている。しかし、「バザールモデル」の中心的な開発者グループはもっぱら構成管理に徹しており、コミュニティ内から自主的に提供された修正パ

ッチの取捨選択を行って、プロダクトの統一性を保つことを中心的な活動としている。

[1]では生命の進化史から次の原則が示されている。

「進化論的に優位に立つものは他から能力を借りたり利用したりして自らの能力を超えたことを行う」

この原則に照らせば、開発のためのリソースを、閉じたグループ内だけでまかなう独占的ソフトウェア・プロジェクトや「伽藍モデル」に従った開発プロジェクトは劣勢に立たされる圧力を受けるということになる。規模とメンバーの多様性が限られた閉じた開発グループは時代の要請の急激な変化にさらされた場合対処できない場合が出てくる。

「バザールモデル」の開発コミュニティはまったく出入り自由であり、潜在的に開発者になり得る技術者プールは実に多様である。時代の要請の急激な変化にあっても、この技術者プールの中にはそれに対処できる技術者がおり、その技術者の助けを借りて対応できる可能性がある。中心的な開発者グループはそうした「ほかの人たちのよいデザイン上のアイデアを認識できる」[3]ことが求められており、本人たちが技術的な対処そのものまでを行う必要はない。このことはまさに、他の能力を借りて自分の能力以上のことをなすとげており、上記の原則によく合致している。

また、「バザールモデル」では開発コミュニティ内の誰もが開発中のソースを参照することができるため、[4]で指摘されているように結果的に強力なピアレビューを受けることになる。これは

平均としてコミュニティの大きさに応じた並列度で行われ、プロダクトの規模の増大に対して品質面で容易に対処できる要素となっている。これも閉じた開発グループにない特徴である。

3.3 オープンソースの開発プロセスの強み

以上、オープンソースの開発プロセス、とりわけ「バザールモデル」に従った開いた開発プロセスの特徴がサイバースペース内での発展圧力の枠組みの中でどう理解できるかを検討し、独占的ソフトウェアの閉じた開発プロセスに対して潜在的に優位に立っていることを確認してきた。

オープンソースの力の源泉のひとつは「誰にも所有されない」ということである。3.1で検討したように、サイバースペース内では強力な主役交代圧力がかかり、適切な変化ができないと急速に存在場所をなくしてしまう。その実例としては、通信プロトコルで事実上世界で共有されているTCP/IPが残り、特定の企業の独占的プロトコルがすっかり姿を消してしまったことが挙げられる。

もうひとつは、3.2で検討したとおり、開いた開発プロセスを取ることで、世界に散在する技術者の力を、量と多様性において必要なだけ確保できることである。コアの開発者はそうすることで自らの能力を遙かに超えることが実現できる。

こうした特徴は、プロジェクトの規模が大きくなったときに特に強みを発揮する。[7]で指摘されている「ブルックスの法則」から、閉じた開発プロセスではプロジェクトが大きくなると運営が

非常に困難になる。この点はモダンPMの分野でもたびたび指摘されることである。しかし、開いた開発プロセスを取るオープンソースではプロジェクトの規模増大に対して運営コストの増加が緩やかであり、コミュニティの規模と多様性の増大に対して線形にその恩恵を受けることができる。これにより、コミュニティの規模と多様性が一定の臨界点を越えたときに、独占的ソフトウェアの閉じた開発プロセスでは到達不能な質的変革を起こすと期待される。

開発技術教育の観点から、方法論寄りの検討をしておく。これまで検討してきた強みを発揮するには、ある程度の規模の開発コミュニティを形成して「バザールモデル」のプロセスをうまく回す必要がある。そのためには[2]や[3]で議論されているように、いくつか実務的な前提条件がある。また、その前提条件を満たした後の具体的な運営方法の詳細はプロジェクトごとに異なる。さらに、CVSのようなツールの進歩、SourceForge (<http://sourceforge.net/> 日本版は <http://sourceforge.jp/>) のようにコミュニティ形成を支援するサービスも登場してきており、最適なプロジェクト運営方法も進歩している。

結局、具体的な方法論は変化していき、受講者が実際にプロジェクトを開始するときには学習した方法論が陳腐化しているかもしれない。しかし、オープンソースの開発プロセスの強みの源泉について理解していれば受講者自らが自分のプロセスを設計することができる。

3.4 オープンソースの社会への影響力

オープンソースの開発プロセスは、ソースコードの共有・公開、自由な利用、その成果の還元・共有により発展している。この過程はスケラビリティを持っていると同時に、ある時点で質的変革をもたらす。

これまでに検討してきたように、特定の技術や技術者を所有してその範囲内で開発を続けることには限界がある。それに対して、オープンソースの開発プロセスのように共有・協力をベースにすればその限界を超えることができ、いわば無限の資源を手にすることになる。

ここで言う「無限」とは、単にソフトウェアは使っても減らないということではない。ソフトウェアはある時点での特定の状況に適合するように作られている。ハードウェアの進歩や社会状況の変化により、そのままでは利用できなくなったり、目的に適合しなくなったりする。そこで、ソフトウェアを保守して新しい環境・状況に対応させる必要が出てくる。

そこで適切な保守作業がなされないとそのソフトウェアは「なくなる」（よく、保守されていない古いソースコードを「腐っている」と言う）。

オープンソースの共有・協力プロセスでは誰かがプロダクトを必要としている限り適切な保守がなされる。その意味で「無限」資源であると言えるのである。

オープンソースのライフサイクルの初期開発を見てわかるとおり、オープンソースは誰かの必要を満たすために開始されるので、既存のプロダクトの保守

だけでなく、新しいニーズに応えるプロダクトも生み出されることになり、その意味でも「無限」資源たり得る。

コンピュータ、インターネットの一般への普及とともに、サイバーワールドは現実の世界に多大な影響を及ぼすようになった。コンピュータネットワーク上で行われる金融取引の金額は一国のGDPを凌ぐものになり、企業活動においてはコンピュータ・システムが必須のものとなっており、利用者を介して現実の世界とサイバーワールドは相互に影響し合っている。

こうしたことから、オープンソースの無限資源を生み出すメカニズムが現実の世界にも影響を及ぼす。

サイバーワールド内での無限資源の生成は国家の枠を超えて行われる。すなわち、サイバーワールド内では世界の相互依存度が高まる。この状態で国家間の争いなどで無限資源生成を実際に担っている人々を傷つければ、サイバーワールドの無限資源生成能力を直ちに傷つける。そうすると国家間の争いそのものでは勝った側にとっても不利益を被ることになる。

このことはサイバーワールド内の無限資源の重要性が高まるにつれ、現実の世界での有限資源争奪の相対的な利益が小さくなっていくことを示す。

このようにサイバーワールド内での無限資源生成が現実の世界の変革につながるためには、無限資源生成に関わる人々、すなわちオープンソースの開発者が特定の国に集中するのは好ましくなく、弱者と見なされている国々でのオープンソースへの貢献度を高めることが

必要である。オープンソース教育を政策レベルで考える場合にはこうした観点も必要である。

4. 課題

オープンソース教育のカリキュラムとして、オープンソースの社会的意味を論じる際には本論文で検討した観点以外にも経済的、法律的な観点等も必要である。

特に、法律的観点では現在の知的財産権重視の流れとオープンソースの発展過程は多少齟齬を生じる。例えば、特許は内容が公開されることを保証するものの、特許の所有者以外がその知識を利用しようとすると多額の特許料を支払う必要があり、「自由な利用」には歯止めがかかる。また、著作権も「自由な利用」はできない。そして、両方とも知的財産の「独占的所有」を保証するものであり、オープンソースの共有原則とは相容れない。

このように、現在の知的財産権保護のしくみはオープンソースの力の源泉とは、やや矛盾する性質を持っている。しかし、教育としてはこのような課題も含めて伝えなければならない。

5. 参考文献

- [1] Toshiyasu L. Kunii, "The Potentials of Cyberworlds -An Axiomatic Approach-", Proceedings of International Conference on Cyberworlds, 18-20 November 2004, pp. 2-7, Tokyo, Japan, IEEE Computer Society Press, Los Alamitos, California, U. S. A.
- [2] 藤枝和宏：『オープンソースソフトウェアの開発スタイルとその変遷』, 情報処理学会誌 2002 年 12 月 Vol.43 No.12 pp. 1325
- [3] Eric S. Raymond : "The Cathedral and the Bazaar": 山形浩生訳『伽藍とバザール』, 1999, <http://cruel.org/freeware/cathedral.html>
- [4] Eric S. Raymond : "The Magic Cauldron": 山形浩生訳『魔法のおなべ』, 1999, <http://cruel.org/freeware/magicpot.html>
- [5] IEEE-CS, ACM コンピューティングカリキュラム共同タスクフォース : 情報処理学会情報システム教育小委員会訳 : 『コンピューティングカリキュラム 2001 (ドラフト)』 pp. 18, 2000.11.15 改訂, <http://www.bunkyo.ac.jp/~miyagawa/ISkyoiku/>
- [6] Karl Fogel, Moshe Bar : 竹内里佳訳、でびあんぐる監訳『CVS CVS によるオープンソース開発』, 第 8 章「オープンソースプロセス」, オーム社, 2002 年 6 月 25 日第 1 版第 1 刷, 2003 年 7 月 20 日第 1 版第 2 刷
- [7] Frederick P. Brooks Jr. : 滝沢徹他訳 : 『人月の神話 (原著発行 20 周年記念増訂版)』, 第 2 章「人月の神話」, アジソン・ウェスレイ・パブリッシャーズ・ジャパン, 1996 年 2 月 15 日初版第 1 刷, 1996 年 4 月 15 日初版第 2 刷