

# 革新的ソフトウェア・プロジェクトを目指して

— 無形労働 (Immaterial Labor) [\*] の視点から —

-

岸田 孝一

SRA

E-mail: k2@sra.co.jp

[\*] Immaterial Labor の訳語は、社会学関連の文献では普通「物質的労働」(Material Labor) に対比して「非物質的労働」となっていますが、ここではその訳語に含まれる政治的ニュアンスを避けるために、伊藤昌夫さんの提案にしたがって「無形労働」という訳語を使用しています。

Immaterial Product は「無形プロダクト」。音楽やソフトウェアは CD という形態をとっていますが、そこに記録された「かたちのない」情報が主体です。介護そのほかのサービスのよう、もともと「かたちのない」プロダクトを提供する無形労働も存在します。

## ABSTRACT

これまでのソフトウェア工学は、物質的労働 (Material Labor) のパラダイムにもとづく規律 (Discipline) を中心に構成されてきた。この論文では、ソフトウェアの開発や保守の仕事を 無形労働 (Immaterial Labor) としてとらえ、新しいプロジェクト・マネジメントのあり方を探る。

## Keywords

無形労働, ソフトウェア・開発, プロジェクト管理, 技術革新

## 1. はじめに

ソフトウェア工学の概念が提唱されたのは、1960年代末にヨーロッパで行われた2回の NATO ワークショップでの討論の結果である [1]。当時は参照すべき既存の工学は、機械工学や電気工学などの物質的労働を対象とするものしかなかったので、ソフトウェア工学の概念的枠組みもそうした工業生産のパラダイムに沿って、工程の標準化や分業、作業の属人性の排除などの概念を用いて構築されることになった。

このアプローチは、あらかじめ仕様が確定した(あるいは比較的安定した)ソフトウェア・プロダクトの開発にさいしてはそれなりの成功を収めた。しかし、ハードウェアの小型化・低価格化やネットワーク技術の進展にともなう、社会のさまざまな分野にコンピュータの利用が浸透してきた現在、われわれが扱うソフトウェア・システムは、多様なユーザのさまざまな文化や情報を内包するプロダクトになり、われわれの仕事、学習、コミュニケーション、娯楽などの活動を定義しなおすようなプロダクトと化している。当然

のことながら、その仕様はあらかじめ確定することはできず、また周囲の社会環境の変化に対応して絶えず変化し続ける。

伝統的な工学パラダイムには、そうしたソフトウェア・プロダクトの開発や保守のプロセスを考える上で、基本的な何かが欠けているといわざるを得ない。われわれはいま、根本的に新しいパラダイムを検討すべき時に来たと考えられる。

## 2. 無形労働 (Immaterial Labor)

音楽レコードやビデオCDが商品として市場に出回り始めたとき、多くの人びとはそれらを知的労働の成果にもとづく新しいタイプの工業製品だと受け止めた。しかし、マウリツィオ・ラッツァラート [2] は、状況を別の視点から眺めて、今日のポスト工業社会における支配的な労働形態に関する考察をおこなった。現在起こっている社会現象は、文化や情報を内包するさまざまな商品の生産や消費が、全世界の経済システムに浸透し、影響を与えつつあるということなのである。それは、これまでの工業的生産のプロセスが、文化や情報が本質的な役割を占める無形労働のプロセスに置き換わりつつあるということの意味している。そうしたプロセスによって、ファッション、趣味・嗜好、消費傾向、芸術的標準、さらには世論といったものが形成され決定される時代が到来したともいえる。

無形労働のプロセスを構成する文化や情報に関する知的活動は、これまで、一部のブルジョア階級の手慰みとして扱われ、工場労働と同列の労働とは認められていなかった。しかし、今日の状況は、ITやネットワークの発達にともない、ほとんどの人びとがなんらかのかたちで無形労働に従事するようになったといっても過言ではない。

無形労働は生産と消費の関係を変えてしまう。従来は、生産がニーズに対応し、プロダクトの消費は価値を生み出すというかたちであった。しかし現在、無形労働が作り出すのは、既知の消費ニーズに応えるためのプロダクトではない。音楽やファッションなどの「かたちのない」プロダクトは、消費者の物質的ニーズに対応するのではなく、生産者からの刺激によって生み出された人びとの非物質的な欲望を満足させることを意図している。つまり、無形プロダクトが経済的な価値を持ちうるのは、その背後に含まれているアイデアがはっきりと消費者に伝えられ受け入れられた場合だけなのである。そして、消費はさらに新しいニーズを作り出し、それを生産者の側にフィードバックする。こうした社会化のプロセスが、そうしたプロダクトの特徴だといえよう[3]。

無形労働によって作り出されるプロダクトの価値や品質は、その生産に携わる人びとの知識や創造性に大きく依存する。生産者に求められるのは、次の3つのスキルである。すなわち、(1) 知的スキル、(2) コミュニケーションスキル、そして(3) 自発的行動力。これらのスキルは、ソフトウェア開発者にとって必須のものだと考えられる。

## 3. 無形労働としてのソフトウェア開発

いうまでもなく、ソフトウェアは「かたちのない」ものであり、ソフトウェア開発の仕事は無形労働の一種

である。作業工程の細分化や標準化などの手法にもとづくテイラー方式の科学的管理は無形労働には使えない。マネージャにとっては、生産ラインの職長として部下を監督するのではなく、開発者の自主的行動を尊重し、協働や創意工夫を促す役割を担うが必要になる。

ソフトウェア開発組織とは、本来、相互にさまざまな関係を持つプロフェッショナルの集団であり、それぞれのメンバーは、そこで行われる開発作業の内容に関心を抱き、みずからの貢献に対してそれなりの見返りが得られる場合にのみ、その組織に参加することが理想的な人たちである。ラッツァラートは、こうした組織のあり方が今日次第に支配的になりつつあると指摘している。多様な自主的労働形態があちこちで誕生し、大小さまざまなサイズの作業ユニットが特定のプロジェクトのために結成され、仕事が完了すると、メンバーたちはまた無形労働の人材プールにもどって行く。

思想的な枠組みとしての無形労働は、ソフトウェア開発およびソフトウェア産業に対して、次の2つの点で深い関係がある。

まず第1に、われわれが現在開発し保守運用しているソフトウェア・システムは、単なる物質的プロダクトではなく、労働・学習・コミュニケーション・娯楽といったわれわれの社会生活のかたちを変革(再定義)する機能を含んでいる。その社会的な価値は、ユーザがシステムを利用することによって確定されるのである。

そして第2に、ソフトウェア・システムは、旧来の物質的労働を無形労働に変貌させる原動力として働くのだということを忘れてはならない。自動車工場や家電製品工場の製造ラインで働いている人びとの多くは、単に物質的な生産に携わっているのではなく、かなりの時間をソフトウェア・システムとのやりとりに費やしている。そうした生産支援システムを開発したソフトウェア技術者は、単にユーザにツールを提供したのではなく、ユーザの生産活動やその社会的形態を変革する役割を演じたのだといえよう。

ソフトウェア・システムの要求仕様は、もはや、そこにある情報を把握して分析すればよいというようなものではなくなった。仕様は、何らかのかたちで、われわれ開発者が創りだし設計すべきものである。ソフトウェアは、既存のニーズを満足させるためや何らかの現実世界のモデルをコンピュータに載せるために作られるわけではない。ソフトウェアは、曖昧な、あるいはまだ存在していないニーズをかたちあるものにし、それをユーザに提示して現実を認識しなおさせるためのツールなのである。すでに存在するニーズにもとづいて生産が行われるのではなく、生産と消費のプロセスを通じて、潜在的なニーズがかたちあるものになって行く。すなわち、ソフトウェア・システムは、単に工場の中で製造され、塀の外にいるユーザに投げ渡されるようなプロダクトではない。それは、生産者と消費者とのあいだの社会的な関係を確立するためのメディアである。ソフトウェア産業にとっての成功の鍵は、ユーザがプロダクトの消費プロセスを通じて生み出す新しいニーズをいかにして的確に把握し、それに応えるべき技術革新を行えるか否かだといえよう。

したがって、絶えず変化し続ける要求仕様を確定させることは、ソフトウェア工学にとっての主要な研究開発課題ではありえない。むしろそれは、潜在的なニーズを探り出し表面化させることによってソフトウェア・システムの価値を高める上での基礎だと考えられる。最近提唱された「いつまでもベータ(Forever Beta)」という新しい概念は、無形労働によって作られるタイプのソフトウェアの本質を反映している。近年盛んになってきた新しい製品リリースのアプローチは、そうした考えのもとづくものである。

もちろん、これまでソフトウェア工学のメイン・テーマであった開発技法やプロセスの革新は、引き続き重要な課題であり続けるが、しかし、何を作るか (What?) よりも、どうやって作るか (How?) や、なぜ作るか (Why?), そしてどうなるのか (Where to?) のほうに焦点が移って行くであろう。

無形労働の視点からソフトウェア開発という仕事を眺めてみると、それは、ある種の設計作業であることがわかってくる。そこには2つの側面がある。ひとつは、プロダクトとしてのソフトウェア・システムの設計である。これは、従来の意味でのソフトウェア設計にあたる。そしてもうひとつは、そのソフトウェア・システムを利用(消費)するユーザとのコミュニケーションによって生まれる価値の設計である。どちらがより重要かという問題ではない。両者はソフトウェア設計の2つの側面であり、上手に統合されるべきものである。無形労働としてのソフトウェア開発に課せられた仮題は、この設計の二面性にどう対処するかであろう。その意味からすれば、Participatory Design の方法論[4]や Socio-Technical Theory[5]について、さらに深い研究が進められることが望ましい。

これまでのソフトウェア工学界における支配的な意見は、「ソフトウェア・システムを与えられた仕様を満足して正しく開発することが主要な任務であり、そのための技法が不足している」ということであった。しかし、無形労働の視点から見れば、そもそも「正しい」システムなどは存在しないのである。システムのほんとうの価値は、ユーザによってそれが使用されるプロセスを通じて、初めて確定されるものだからである。

#### 4. ソフトウェア開発プロジェクトの組織

無形労働としてのソフトウェア開発プロジェクトを組織するにあたって、特にプロジェクトの実践を通じての技術革新を考える上では、次にあげるような指導原理(Principle)が重要だと考えられる。

##### Principle 1: 動機付け(Motivation)

開発技術者たちに明確な動機付けを与えそれを維持して行くことが、まず第1のプライオリティである。開発プロジェクトにおいて主役を演じるのは開発者たちだからである。ソフトウェア・システムの価値は、その設計や具体化にさいしてのさまざまな技術的選択肢の選択によって決まる。そしてそれには、開発者たちの個人的嗜好や感覚が大いに関係している。イギリスの研究者がおこなった調査によれば、ソフトウェア開発者の動機付けに影響を与える因子は、挑戦、変化、利益、問題解決、チームワーク等であり、なかでも、変化する仕様を相手にして挑戦的な仮題に直面した場合に、開発者のモチベーションが大きく高まるということが報告されている[6]。

動機付けは、技術的な革新のみならず、ソフトウェア開発に固有な社会的協同作業にとっても重要な意味を持つ。開発プロセスの任意の時点において、開発者たちは、それぞれの技術的ニーズを相互に交換し合い、情報の共有を円滑におこなうための、臨機応変なメカニズムを必要としている。

20世紀ドイツの美術界を代表するコンセプチュアル・アーティストとして知られたヨーゼフ・ボイスは、かつて「すべての人間は芸術家である」と発言した[7]。このコトバは決して、だれもが画家や彫刻家にな

れるという意味ではない。人間のすべての活動には何らかの創造的機会が潜んでいるという事実を指摘しただけのことである。ソフトウェア・プロジェクトのマネージャは、そのことを常に留意し、メンバーの動機付けを心がけなければならない。

## Principle 2: 多声 (Polyphony)

近年その評価が高まりつつあるロシアの思想家ミハイル・バフチンが提唱した「多声」(Polyphony) および「非完結性」(Unfinalizability) の概念は、ソフトウェア・プロジェクトの技術的マネジメントにとって、きわめて重要な意味を持つと考えられる。バフチンは、ドストエフスキーの文学を綿密に分析して、この2つの原理を発見した[8]。

ドストエフスキーの小説には、さまざまな人物が登場して、人間が生きることについての「深遠な」真理の問題をめぐる討論を続けて行く。作者としてのドストエフスキーは、これらの人物の誰か1人に肩入れしたりすることなく、全員を平等に取り扱っている。小説のストーリーはどこかの時点で終わるのだが、それまでに続けられてきた対話や討論は結論には到達しない。世界で最初の「対話型小説」としてのドストエフスキーの作品のユニークさはそのあたりにある。

バフチンは次のように述べている；

世界の究極の問題である真理は、単独の個人的意識の中では解明できないと、ドストエフスキーは考えていました。真理は、1つの意識には収まりきれないのです。真理は、常に部分的にですが、多くの対等の意識が対話的に交流するプロセスの中で解明されて行きます。この対話は、真理について考え、真理を求める人間が存在する限り、終わることも完結することはありません。対話の終わりは人類の破滅を意味します。

ドストエフスキーは、ポリフォニック(多声的)な小説、つまり、究極の問題をめぐる緊迫した激しい対話として構成された小説の創始者です。作者はこれらの対話を完結させたり、作者としての自分の結論を提供したりはしません。かれは、いかなるかたちであれ、完結というものを認めませんでした。

たとえいくつかの小説(たとえば「罪と罰」)が完結しているようであっても、それはただ形式的な文学的完結にすぎません。「カラマーゾフの兄弟」はまったく完結していません。そこでは、あらゆるものが開かれたままです。あらゆる問題が問題のまま残り、一定の結論を示すものは何もありません。大切なのは、異なる人間によって具体化された思想が、多数存在することなのです。重要なのはそれらの間の対話、明らかに完結されない対話なのです。対話の完結、論争の終了は、外からの野蛮な物質的な力を導入すれば実際には可能であることを、ドストエフスキーは作品の中で一度ならず示しています。しかし、本質的には、そうした対話的思考やその意味というものは、完結されません。

われわれに与えられた課題は、文学作品を対象に展開されたバフチンの分析手法をソフトウェアに

置換えて、さまざまな視点や思惑が混在する多声的(Polyphonic)な開発プロジェクトのあり方についての理論を整備することである。本来何らかの時間的制約のもとにあるプロジェクトは、「外からの」強制力によって完結させられるが、ソフトウェア自体が内包する諸問題の解決は決して完結しないし、そのソフトウェアに関係する開発組織やユーザ組織もその問題を抱え続けることになる。ソフトウェア開発者たちは、これまでにそのプロジェクトで行われた対話や討論の中間的成果を携えて、次のプロジェクトに進んで行くのである。

バフチンはまた、フランス中世の民話文学の分析から、「カーニバル文学」という独自の視点を提示している。このメタファは、オープンソース/フリー・ソフトウェアの社会的問題を考えるさいに、エリック・レイモンドが提示した「伽藍とバザール」[9]よりも役に立つように感じられる。

### Principle 3: 差異 (Difference)

これまで、ソフトウェア工学の方法論は、規律や統制を中心とする物質的労働(工業文明)のパラダイムにもとづいて構築されてきた。たとえば、その典型的な例として、ソフトウェア開発プロセスの成熟度を示す **CMM** では、その第2レベルを「反復可能」(Repeatable)と呼んでいる：

このレベルにおけるプロセスの特性は、それが「反復可能」であり、予期される一手の成果をもたらすということである。プロセスの規律は、決して厳密不可侵なものではないが、しかし現実のプロセスが所定のかたち維持されることを確実にするものでなければならない。

**CMM** は、それぞれが反復可能であって、所定の成果をもたらすような一連の **Key Practice Area** から構成されている。そこで示されている規律は、もし似たようなプロジェクトを繰り返し実施する場合には、開発プロセス内のそれぞれの活動について、きちんとしたドキュメントが整備され、教育訓練が行われ、効率的に実施されるべきだということである。そうした規律は、工場生産における物質的労働には適しているが、しかし 無形労働としてのソフトウェア開発においては、ややもすると開発者の創造的行為を抑圧する結果をもたらす。いわゆる **SPI** 活動がしばしば現場からの抵抗に遭遇することの理由は、そのあたりにあると考えられる。

ポスト構造主義を代表するフランスの哲学者ジル・ドゥルーズは、「同じものが反復されることは決してない。差異だけが反復されるのだ」と主張している。その著書「差異と反復」の序章は次のようなパラグラフで始まっている[10]。

反復は一般性ではない。それは、いくつかの仕方で一般性と区別されなければならない。両者の混同を含む表明は厄介な代物である。たとえば、わたしたちが、「一般的なものについてしか科学は存在しない」というのと、「反復するものにしか科学は存在しない」というのを、同一視するような場合である。差異は、その本質上、反復とその類似(たとえそれがどのように近い類似であろうと)との間に存在する。

邦訳で2段組500ページを越えるこの哲学的大著の全容を解説する能力は、残念ながらわたしには

ない。ここでは、何冊かの注釈書を参考にしつつ、ドゥルーズの思想を、ソフトウェア開発との関連を考えながら追いかけてみよう。

差異について考えるさいに陥りやすい誤りは、すでに同一のものとして特定されているものを基準にして差異を規定することだ、とドゥルーズはいう。たとえば、**A**を基準にしてそれと異なるものとして差異を考えたり、**A**と**B**の2つを取り上げて両者の差異を論じる場合が、それにあたる。このとき、差異は同一性との関連で定義されることになる。差異について考えるはずだったのに、結局、同一性が思考の基礎に据えられてしまうのである。

では、同一性を経由しない差異とはいかなるものか。それにはまず、同一性をできあがったものとして考えるのをやめ、同一性という観念や同じものとして認識される個物が、いったいどんなプロセスによって作り上げられたかを考えることである。同一性とは、ある生成プロセスの途中で生まれる結果(あるいは効果)にすぎず、決して何かの前提や原理にはなりえない。そのようにプロセスの中でとらえられる同一性は、むしろ自分自身に対して異なるものになることを、その本質としている。同一性は、ただそのように見えるだけの視覚的な特殊効果にすぎない。正確に同じ状態が、観念としてであれ、物体の状態としてであれ、再現されることはありえない。反復は、差異をともなわずになされることはないからである。

ソフトウェア開発プロセスにおいては、同じような活動が繰り返し行われるように見えることが多い。それらを同一なものとして把握し定義するというのが、いわゆるソフトウェア工学におけるさまざまな技法やモデルの原理になっており、そこからの「差異」を規制したり分析したりすることが、いつのまにかわれわれの生活習慣と化している。

ドゥルーズがいうように、同一性の反復が単なる視覚的效果に過ぎないのだとしたら、われわれは、そうした反復の原理を捨て、「差異」自体についての分析を行い、それにもとづく方法論を構築すべきであろう。振り返ってみれば、これまでのソフトウェア工学における技術的革新は、まさに開発プロセスにおける「差異」を認識することから生まれてきたのである。

#### Principle 4: リゾーム(根茎, Rhizome)

フェリックス・ガタリとジル・ドゥルーズの共著になる「千のプラトー」では、これまでの西欧哲学に共通であった「樹木」型の階層構造の思考モデル(根・幹・枝・葉)に代わるものとして、「リゾーム」(Rhizome: 根茎)のメタファが提案されている [11]。リゾームとは、植物学では、地面の下に水平に生え広がる根を指す。典型的な例は竹の地下茎である。ガタリとドゥルーズは、このリゾームが、人間の思考のモデル、さらには知識・文化や社会のモデルとして有用ではないかと主張する。

文化のモデルとしての樹木型の階層構造は、常にその出発点である「根」に注目し、そこから時系列的に立ち上がってくる「幹」や「枝」、そして「梢」や「葉」について考えるかたちになっている。このフレームワークが、これまでの西欧の現実とすべての思考を(植物学から動物学まで、解剖学、さらにまた認識形而上学、進学、存在論、全哲学)を支配してきたと、ガタリ&ドゥルーズは指摘する。

書物の第1のタイプは根としての書物である。樹木はすでに世界のイマージュになっている。根は、世界としての樹木のイマージュなのである。それは、有機的、意味作用的、主体的な美しい内面性としての古典的な書物である。

そのことは、たとえばわれわれの周囲にある **SWEBOK** ほかのソフトウェア工学文献のスタイルをみればあきらかであろう。

リゾームは、それとは異なり、水平にあちこちの方向につながった線が、どこにも中心をもたず、どこからどこへでも自由に連結し幹を立ち上がらせる。それは、さまざまな意味論的連鎖、権力構造、状況などを、芸術、科学、あるいは社会組織とランダムにつながり合わせるオープンな多様の構造体なのである。リゾーム・モデルの基本的特徴は次の通りである：

- 1) 連結性および異種混合性：リゾームは、その任意のポイントにおいて他のものそして他のポイントと連結される。そのようなことが可能でなければならない。
- 2) 多様性：リゾームは多様体である。それは、いかなる場合にも何らかの「一」なるものと関係付けられることがないという意味で多様なのである。
- 3) 断裂性：リゾームは任意の場所で断裂する可能性がある。そしてそこから、あるいは他の場所から、すぐに再生し、同じラインあるいは別の新しいラインを進んで行く。
- 4) 地図性：リゾームは、道のない地図である。どこから出発しどこへ行くのかといった問いは無用であり、始まりも終わりもなく、いつも中間である。それはいかなる構造的なモデルあるいは生成的なモデルにも帰着されない。

-

こうした特長は、特に創造性を重視したソフトウェア・プロジェクトの組織構成を考える場合にきわめて重要な意味を持つと考えられる。

## Principle 5: 開放性(Openness)

ソフトウェア・プロジェクトは、外の世界に向かって開かれた技術的な窓を持つことが望ましい。そしてその窓の役割を演ずる人間が必要である。プロジェクトは、その窓を通じて、内部の活動の成果を外に向かってアピールすることができ、また外から新しいアイデアを導入することができる。古代中国の都市国家において、すぐれた統治能力を持った諸王が、その城壁の外の荒野を旅する諸子百家の知恵を取り入れて政治の革新を図ったことは周知の事実である。秦の始皇帝の成功が、そうしたオープンな政治姿勢を徹底したことにあっただけでもない。

さらにいえば、**GNU** や **Linux** などのフリーまたはオープンソース・ソフトウェアは、まさにラッツァラートが暗示したのと同様なプロジェクトによって開発されてきた。そうしたプロジェクト・チームは、インターネット上に分散されたメンバーから構成され、各人が自分の責任において特定の問題に取り組んで行く。

協働はボトムアップに行われ、マネージャの管理下ではなく、メンバー相互の意志にもとづいて開始さ



れ管理される。すべてのメンバーが、ある意味で事業経営者なのである[12]。

それと似たプロジェクト編成は、通常の企業組織の内部でも取り入れることができるだろう。数年前の EuroSPI で、ヨーロッパのある通信機メーカーが、世界中に分散したアジャイル開発チームをそうしたオープンな形で運営し成功を収めたという事例が報告されている。それは、無形プロダクトを開発するプロジェクトの新しいスタイルだといえるだろう。

## 5. 終わりに

ソフトウェアは、そこに含まれる文化的情報を本体とする無形プロダクトであり、われわれの仕事や学習、コミュニケーション、あるいは娯楽のスタイルを変革する力を持っている。しかし、これまでに構築されてきたソフトウェア工学の方法論は、旧来の工業文明に固有の物質的労働を規律正しく管理するというパラダイムにもとづいて構築されてきた。われわれは、いま、無形労働の視点に立ってすべてを考え直すべきであろう。この小論では、技術革新を目指すプロジェクト・マネジメントについていくつかの新しい指導原理の提案を試みた。

## 6. 参考文献

- [1] Naur, P. and Randall, B., "Software engineering: A report on a conference sponsored by the NATO science committee," NATO, 1969.
- [2] Lazzarato, M., "Immaterial labour," in *Radical thought in Italy: A potential politics*, P. Virno and M. Hardt, Eds. Minneapolis, MN: University of Minnesota Press, 1996, pp. 133-147.  
<http://www.generation-online.org/c/fcimmateriallabour3.htm>
- [3] Lazzarato, M., "European cultural tradition and the new forms of production and circulation of knowledge," 2004.  
<http://multitudes.samizdat.net/article1292.html>
- [4] Ehn, P., *Work-oriented design of computer artifacts*. Stockholm, Sweden: Almqvist & Wiksell International, 1988.
- [5] Ye, Y., Yamamoto, Y., and Nakakoji, K., "A socio-technical framework for supporting programmers," in *Proceedings of 2007 ACM Symposium on Foundations of Software Engineering (FSE2007)*, 2007, pp. 351-360.
- [6] Hall, T., Sharp, H., Beecham, S., Baddoo, N., and Robinson, H., "What do we know about developer motivation?" *IEEE Software*, vol. 2008, pp. 92-94, 2008.
- [7] 水戸芸術館現代美術センター編, 「ヨーゼフ・ボイス:よみがえる革命」, フィルムアート社, 2020.
- [8] ミハイル・バフチン, 「ドストエフスキーの詩学」, ちくま学芸文庫, 1995.

- [9] エリック・レイモンド, 「伽藍とバザール」, 光芒社, 1999
- [10] ジル・ドウルーズ, 「差異と反復」, 河出書房新社, 1992.
- [11] フェリックス・ガタリ & ジル・ドウルーズ, 「千のプラトー」(上中下), 河出文庫, 2010.  
「リゾーム」はその序章(上巻に所収)
- [12] Tiziana Terranova, "Free Labor: Producing Culture for the Digital Economy  
Social Text - 63 (Volume 18, Number 2), Summer 2000, pp. 33-58  
<http://www.electronicbookreview.com/thread/technocapitalism/voluntary>