

ソフトウェア生産のオープン化と地域の 情報サービス産業

～オープンソース・ソフトウェアによるソフトウェア生産のモジュール化と
情報サービス産業の組織のモジュール化のマッチングの可能性～

Opening Computer Based Software Methodology and Regional
Information Technology Service Industry

野 田 哲 夫
NODA Tetsuo

はじめに

IT (Information Technology) = 情報通信技術と、企業の IT 投資 = 情報化投資はマクロ的にも日本経済再生と経済成長の起爆剤として期待されており、また企業経営革新と業務効率化のためにも IT 投資は欠かせないものとなっている。IT 産業 = 情報産業が産出する「商品」を投入する産業全体の IT 投資によって、企業の生産システムが変革され、企業（企業間）の組織が変化する。これについては野田 [2005] a において、1990年代の米国を中心とした IT = 情報通信技術の革新とそれによる SCM ~ BtoB ~ e マーケットプレイスと進化 & 深化した企業間のネットワーク化の過程と、さらにユビキタス・ネットワークの導入を求める資本の運動、そしてそれが資本にもたらす成果とを具体的に分析する中で、IT 投資を垂直的にも水平的にも資本の拡大と深化の原動力とする「情報資本主義」段階が深化していることを、現代資本主義における巨大企業（群）の存在

キーワード：ソフトウェア生産、オープンソース・ソフトウェア、モジュール、情報サービス産業、地域産業
Computer Based Software Methodology, Open Source Software, Module
Information Technology Service Industry, Regional Industry

形態を分析する中で明らかにしてきた¹。IT投資は取引をより「オープン」にさせることによって競争は激化しながら、一方で巨大企業（群）による「系列」が強化され、市場は巨大企業に包摂されていったのである。

これらの産業の情報化、IT化を支えるのはIT「商品」を産出するIT産業＝情報産業である。情報産業は一般的には、コンピュータ産業（コンピュータおよびハードウェアの開発・製造）、通信産業（通信機器の開発・製造、通信サービス）、そしてソフトウェアの開発やサービス提供を行う情報サービス産業に分類される²。特に情報サービス産業は、インターネットの普及とそれに伴う電子商取引の拡大によって市場の拡大と成長が続いてきた分野である。そして産業間の連関において、他産業への波及効果という点ではインパクトは低い（影響力係数が産業平均を1とした場合0.9）、他産業からの波及は他の情報産業と比べても極めて高い分野でもある（感応度係数が産業平均を1とした場合2.1、総務省統計局「産業連関表2000年度取引基本表」から算出）。それゆえ他の産業の景気動向、他の産業からの需要動向に左右されるという側面が強く、90年代半ばから10%前後の高い伸びを続けてきた情報サービス産業の売上高は2002年に入ってから日本経済のリセッションも反映して鈍化したものの、2004年までに10年連続で増加している³。

IT「商品」自体を供給する産業＝情報産業の技術革新と、IT投資による企業経営革新と業務効率化が好循環をもたらすものであれば、地域経済においてもそれぞれの地域の独自の伝統的な技術とITを融合し、企業の技術力や競争力を向上させる好循環を生み出すことにより、産業構造の高度化と産業活力の向上によって地域産業全体の付加価値を高めていける可能性がある。また、ITは、インターネットに代表されるネットワークの即時性や双方向性を活用することにより、時間・距離のハンディを克服することが可能である。そこで、島根県のような地域においても地理的な要因によるハンディを克服し、ソフトウェア、コンテンツ制作などを中心に知的資産・サービスを県内にとどまらず全国市場に向けて展開、生産・供給し、地域にとって貴重な「外貨」を獲得していくことが期待されている。このように、IT＝情報産業の育成・発展は日本経済のみな

らず経済の停滞感が強くみられる地域の経済再生・成長にとっても必須の課題であるが、情報産業のこれらの特性と可能性は、IT 自体の技術革新とサービスの普及・拡大を背景としており、これはネットワークの拡大とともに進む絶えざる技術革新と市場獲得の競争に地域経済も常にさらされることも意味している。島根県の情報産業にとってもこのことは例外ではなく、ITの可能性を現実性に変えていくためには、この技術革新と市場拡大に対応することのできる情報産業自体の構造変化を迫られることになる。

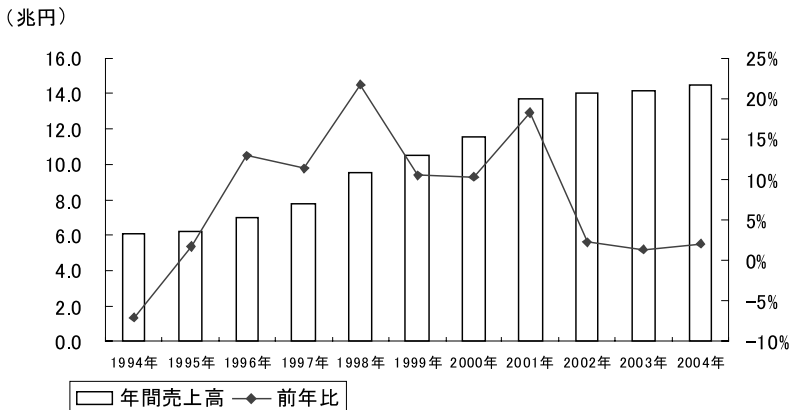
そこで本稿ではソフトウェア生産における技術革新を中心とした情報サービス産業の生産性と、地域における情報サービス産業の変化の実態を、最新の技術動向、各種資料や実態調査を通じた分析を行うことによって、地域における情報サービス産業の課題を抽出することを目的とする。

第1節 ソフトウェア生産の階層構造と生産性

1、日本の情報サービス産業の階層構造

1990年代半ばからの企業のIT投資の拡大によって情報サービス産業の売上高

図1 情報サービス産業の年間売上高推移

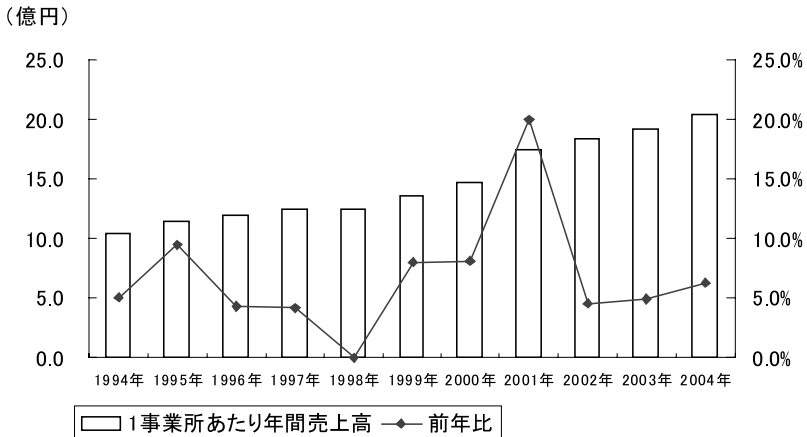


経済産業省 [2005]、『平成16年 特定サービス産業実態調査』確報より

は10%前後の高い伸びを続けてきた。2001年のITバブル崩壊や、金融・保険関連需要の一巡などもあって伸び率は鈍化しているが、年間売上高は1995年以降10年連続して増加しており、2004年の年間売上高は14兆5271億円、前年比2.5%の増加である⁴（図1参照）。

次に、M&Aなどによる企業の再編や中・小規模事業所を中心とした事業所の統廃合などによって事業所数は9年連続で減少が続き（2004年の事業所数は7110事業所で、前年比▲3.7%の減少）、その結果1事業所あたりの年間売上高は増加を続け、2004年は20億4319万円で前年比6.4%となっている（図2参照）。

図2 1事業所あたり年間売上高の推移



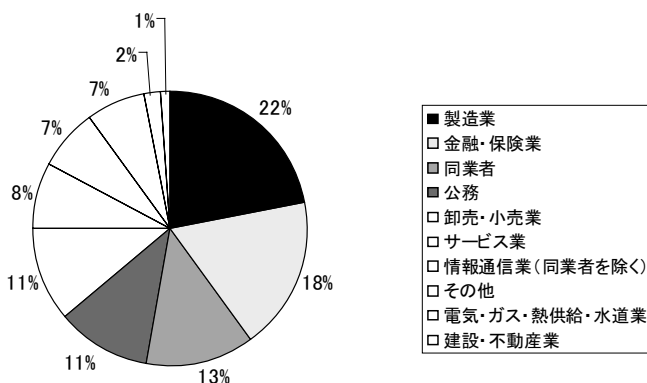
経済産業省 [2005]、『平成16年 特定サービス産業実態調査』確報より

また、年間売上高を契約先産業別にみると「製造業」(構成比21.6%)、「金融・保険業」(同18.1%)に続いて同業者(同13.3%)が目立ち、情報サービス産業における他産業の連関と同時に、産业内・業界内での連関の多重性、同業者間での委託・受託の複雑な連鎖を示している(図3参照)。2003年から2004年にかけて同業者の伸び(4.5%増)および情報通信業の伸び(22.3%増)が目立つが、特に前者に関しては事業所の統合が進んで事業所数が減じているにもかかわらず、同業者からの契約金額が伸びていることは、この産业内・業界内での委託・

表1 情報サービス産業の契約先産業別年間売上高（百万円）

契約先産業	2003年	2004年		
			構成比	前年比
計	14,170,633	14,527,057	100.0%	2.5%
製造業	3,119,535	3,138,035	21.6%	0.6%
卸売・小売業	1,472,796	1,563,344	10.8%	6.1%
建設・不動産業	177,850	174,343	1.2%	-2.0%
金融・保険業	2,639,130	2,632,580	18.1%	-0.2%
情報通信業（同業者を除く）	872,931	1,067,671	7.3%	22.3%
電気・ガス・熱供給・水道業	391,387	334,205	2.3%	-14.6%
サービス業	1,064,131	1,094,734	7.5%	2.9%
公務	1,675,885	1,611,164	11.1%	-3.9%
同業者	1,848,401	1,930,878	13.3%	4.5%
その他	908,587	980,103	6.7%	7.9%

図3 契約先産業別年間売上高の構成比



経済産業省 [2005]、『平成16年 特定サービス産業実態調査』確報より作成

受託の関係が進んでいることを示していると言えよう（表1、図3参照）。

社団法人・情報サービス産業協会が会員企業に対して行っているアンケート調査（回答193社）においても、売上高に占める同業者からの受託額の割合（同業者からの受託額/売上）が20%を超える企業が半数近くあり（48.6%）、また情報サービス受託時に開発を外部委託する割合（外部委託費/受託額）が20%を超える企業が7割以上（73.7%）を占めている⁵（図4、図5参照）。

図4 売上高に占める同業者からの受託額の割合

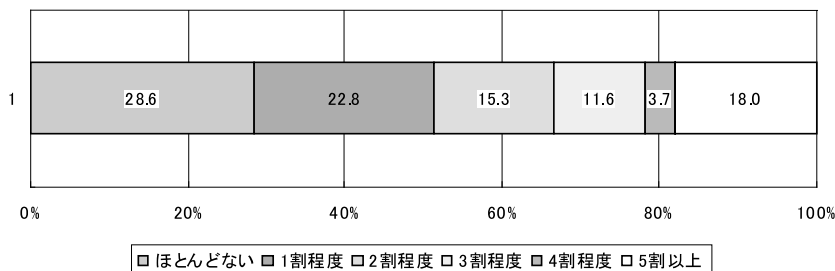
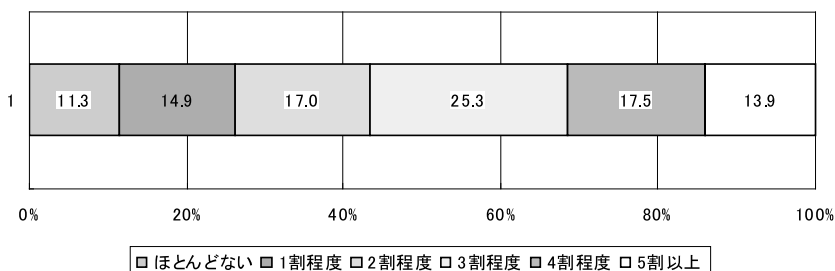


図5 受託時に開発を外部委託する割合

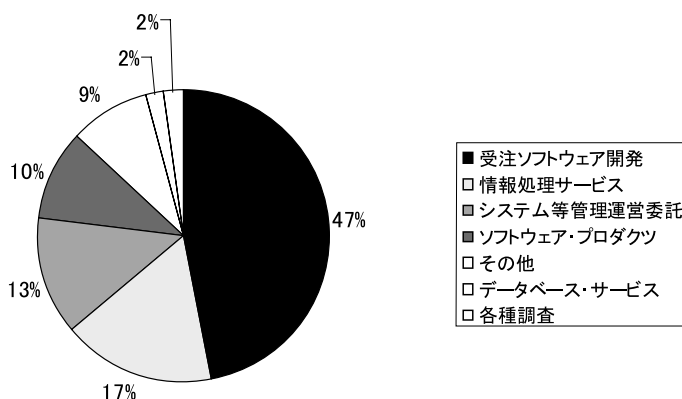


社団法人情報サービス産業協会 [2005]、『情報サービス産業白書2005』より

表2 情報サービス産業の業務種別年間売上高（百万円）

業務種別	2003年	2004年	2004年	
			構成比	前年比
計	14,170,633	14,527,056	100.0%	2.5%
情報処理サービス	2,470,928	2,438,920	16.8%	-1.3%
受注ソフトウェア開発	6,637,179	6,785,991	46.7%	2.2%
ソフトウェア・プロダクツ	1,444,426	1,506,673	10.4%	4.3%
業務用パッケージ	726,631	701,857	4.8%	-3.4%
ゲームソフト	528,275	620,548	4.3%	17.5%
コンピュータ等基本ソフト	189,519	184,296	1.3%	-2.8%
システム等管理運営委託	1,730,291	1,850,261	12.7%	6.9%
データベース・サービス	311,779	319,855	2.2%	2.6%
インターネットによるもの	148,544	178,434	1.2%	20.1%
その他	163,235	141,421	1.0%	-13.4%
各種調査	280,269	283,114	1.9%	0.3%
その他	1,293,761	1,342,242	9.2%	3.7%

図6 業務種別年間売上高の構成比



経済産業省 [2005]、『平成16年 特定サービス産業実態調査』確報より作成

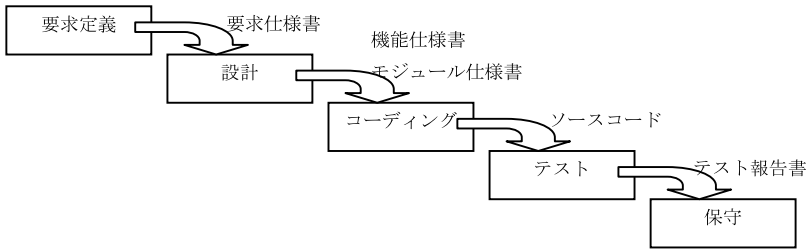
さらに、日本の情報サービス産業に特徴的なのが受注ソフトウェア開発業務であり、売上高の半数近く（2004年で構成比46.7%）を占めている⁶。

この受注ソフトウェア開発＝顧客（他産業）からの委託により電子計算機（コンピュータ）のプログラム作成及びその作成に関して調査・分析・助言などを行う事業を中心とした情報サービス産業の開発・生産体制が、上記の産業内・業界内での連関の多重性、同業者間での委託・受託の複雑な連鎖を生み出しているのである。

2、ソフトウェア生産の階層構造と情報サービス産業の階層構造

受注ソフトウェアの受注・開発過程は、大まかには①上流（システム企画・概要・設計）、②中流（詳細設計）、③下流（プログラム設計）に分けられる。またより詳細には工程として、（1）要求定義＝ユーザの要求仕様を確定しシステム化する、（2）機能定義＝システムをいくつかのモジュールに分割し、それぞれのモジュールごとの手順などを決定する、（3）コーディング＝モジュール仕様書をもとにプログラムを作成する、（4）テスト＝各モジュールを結合してシステム全体の整合性を確認する、（5）保守＝開発したシステムを実務環境

図7 ウォーター・フォール・モデルの概念図



大場充 [2003] より作成

に移し実稼動させる、といったウォーター・フォール・モデル（図7参照）が古くから（1970年代ごろから）採用されている。

後述するように、このウォーター・フォール・モデルはフィードバックの過程がないなどの欠陥からソフトウェア開発には不適切だと指摘が数多くあるが、日本の受注ソフトウェア開発において現在でも幅広く受け入れられている⁷。この理由は前項で見たように日本の情報サービス産業が同業者間での委託・受託の連鎖を作っているからであり、ウォーター・フォールの流れそのままに、発注者からソフトウェア開発を受託した企業（元請け）へ、その仕事の一部を下請けする企業（1次下請け）、さらにはその一部を下請けする企業（2次下請け、あるいは孫請け）へという流れに当てはめてきたのである。

さらに、受注した元請け企業がソフトウェア開発を下請け企業に回すのは、このウォーター・フォールの流れとともに、下請け企業の労働単価の安さが背景にある。受注ソフトウェア開発の契約では多くの場合、システム開発などの工数、仕事量を示す単位、「人数×時間（月）」で表される人月単価⁸で計算が行われており、要件定義工程では3人月で〇〇円、設計工程は10人月で△△円、製造工程では30人月で□□円、よって合計50人月で5000万円といったような工程別人月や総人月をユーザに提示する。ところが実際には下請け企業の人月単価は元請け企業の人月単価より低く、元請け企業は受注にあたっては自企業の人月単価で見積もりを行っており、この単価で受注した仕事を下請け企業に出す

のであるから、当然ながらその差額は元請け企業の利益となる⁹。

従来、企業のソフトウェア開発はその企業独自の仕様を持ち、それに応じて主にその企業系列にある大手コンピュータ・ハードウェア企業や大手ソフトウェア企業＝元請け企業が発注者のニーズ分析とそれに基づいたシステム設計を行い、プログラミング等の業務については直系子会社、独立系1次下請け企業、2次下請け企業などに下ろしてきた¹⁰。前項で見たように日本の情報サービス産業において、開発規模が多額で大量に販売できないと開発コストが回収できないパッケージ・ソフトウェアの市場規模が小さかったのと同時に、受注ソフトウェア開発業務の占める割合を高めていった要因の一つである¹¹。元請け企業が担当する上流工程は詳細設計までの場合もあるし、概要設計までの場合もあるが、残る下流工程を下請け企業が担当するのである。佐野 [2001] によれば、日本の情報サービス産業では、主として上流工程の「要求定義」や「設計」の段階は自社において自社スタッフが業務を遂行し、開発段階では外注化を行って他社のスタッフを工程に参加させるというスタイルをとってきた。いずれにしても元請けで上流工程を行い、下流工程を下請けに出すという情報サービス産業の現状の階層構造には、開発プロセスが逐次的であるウォーター・フォール・モデルが最も適していたのである。

また、日本の情報サービス産業においてはソフトウェア技術者をシステム・エンジニア (SE: System Engineer) とプログラマーに分ける傾向があり、上流工程を担当するのがSEであり、下流工程を担当するのがプログラマーと分けられ、技術面の評価においても給与の面でも階層構造が形成されている¹²。

3、ソフトウェア生産の階層構造と生産性のパラドックス

1990年代の日本経済の低迷の一つの要因として日本経済全般にわたって見られる技術進歩率、労働生産性の低下があったと考えられるが、西村・峰滝 [2002] や同 [2004] による日本と米国の産業別技術進歩率に関する比較研究によれば、情報産業においてIT製造業＝ハードウェア産業に関しては1990年代後半の日本の労働生産性の伸び率は19.5%であり、米国の23.7%、EUの13.8%と比べても

遜色ない。これに対して同時期の非製造業、特にサービス産業での労働生産性の伸び率は米国の5.4%に対して日本は0.0%となっている¹³。情報サービス産業はこの非製造業であるサービス産業に含まれるのであるから、製造業であるハードウェア産業の高い生産性との対比を示している。

また、ソフトウェア開発の能力を評価する上で国際的に認められている CMM (Capability Maturity Model: 能力成熟モデル)¹⁴によると、2000年末時点で、世界で数十社あるソフトウェア・プロセス成熟度の最高ランク＝レベル5 (技術・要件環境の違いによって、標準プロセスを最適化して用いられる段階) に入っている日本企業は1社もない¹⁵。

前項で見た日本のソフトウェア開発に特徴的なウォーター・フォール・モデルでは、上流工程から下流工程への流れにおいて、システムをいくつかのモジュールに分割し、それぞれのモジュールごとの手順などを決定して下ろしていた。このソフトウェア開発のモジュール化が日本の情報サービス産業の階層構造に従って行われるとすれば、これは元請け企業から下請け企業へのモジュールの分割と受け渡し (およびこれに付随した仕様書＝ドキュメントの受け渡し) となって表れる。

コンピュータ・ハードウェアの生産過程においては、もともとシステムを構成する機能をグループ化して分解するモジュールの考え方が一般的であり、IBM システム/360の設計で生まれたモジュール化によるハードウェアの設計と製造が生産性の向上に寄与してきた¹⁶。このモジュール化の考え方はハードから組織へと概念が拡大し、90年代における米国のシリコン・バレーを中心とした情報関連産業における集積と企業間の連関が生産性の劇的な上昇を生み出した¹⁷。生産過程のモジュール化と組織のモジュール化がマッチすることによって相乗的な生産性の上昇を生み出したのである。

しかしながら同じ情報産業でも、ソフトウェア生産＝情報サービス産業においては次節で見る「人月の神話」(あるいは「Brooksの法則」＝1人で100時間かかるプロジェクトを、100人を動員して1時間で行うことはできない) に代表されるように、コンピュータ・ハードウェアや組織における「モジュール化」

の成功をそのまま当てはめることもできない。特にソフトウェアの開発・製造においてはコミュニケーションとコラボレーションが必要とされ、この不足は生産性の低下にもつながる。

西村・峰滝 [2004] では、日本の情報サービス産業の生産性を「特定サービス産業実態調査」のデータを元にして、「モジュール化」と生産性の関連を「外注化比率」で近似的に計算することによって統計的に有意に負の効果を持っていることを算出している。情報サービス産業を4つのグループ（Group 1：期間を通じて単独企業、Group 2：期間を通じて支社保有企業、Group 3：単独事業所から支社保有企業へ転換、Group 4：支社保有企業から単独事業所へ転換）に分け、それぞれ全要素生産性変化率を従属変数、外注費比率（＝モジュール化の代理変数）、SE数、SE比率、利潤・コスト比率を説明変数として重回帰分析を行った結果、いずれのグループにおいても「外注化比率」の符号がマイナスになっており、このことから日本の情報サービス産業ではモジュール化が効率的でないと結論付けているのである。また符合の絶対値がGroup 1では低くGroup 3では最も高くなっていることから、単独事業所のほうが外注を多く活用する状況下であり、そのためにモジュール化が相対的に進展し、組織形態が不安定な企業ではモジュール化がうまく機能していないともしている¹⁸。この重回帰分析の結果から得られるのはモジュール化の非効率性ではなく外注費比率の非効率性（あえて言えば組織のモジュール化の非効率性）であって、前者の結論（モジュール化が効率的でない）は、短絡的である。一方、後者の結論（組織形態による効率性の違い）は、ソフトウェア生産のモジュール化に情報サービス産業の組織のモジュール化がマッチすれば、モジュール化の進展が生産性を上昇させることを示唆している。

インターネットを中心としたネットワークの進展や、企業の合理化によってIT投資においても企業は多額のコストを要するシステムの開発よりも、既にあるシステムを利用するかカスタマイズして使う傾向は強まるであろう。日本の場合は確かにパッケージソフトの利用やカスタマイズの比率は小さいが、受注ソフトウェア開発においても開発側もコスト削減のために既存のパッケージの

応用、カスタマイズで対応しようとする。また Java 言語¹⁹などプログラム言語によるソフトウェアの開発はネットワークの利用と分業をますます進める可能性がある。さらに、次節で詳しく見る Linux などのオープンソース・ソフトウェア (OSS:Open Source Software) の普及はこれをさらに加速化する。特に Web アプリケーション環境や Java 開発ツール分野は OSS が多く、この分野の需要の増加と併せて、情報サービス産業は OSS を活用した開発体制とそのための組織形態の変容を大きく迫られることになるであろう²⁰。次節では、このソフトウェア生産における構造的変化を詳しく見ていく。

第 2 節 ソフトウェア生産の構造変化とオープン化

1、「人月の神話」：ソフトウェア生産の生産性の古典

前節で触れたように日本の情報サービス産業における受注ソフトウェア開発の契約では、システム開発などの工数、仕事量を示す単位で、「人数×時間（月）」で表される人月単価で計算が行われている。この人月表示では「3人×8カ月」も「6人×4カ月」も計算上同じ「24人月」となるため、人数を増やせば納期を早くできると勘違いされることがあるが、ソフトウェア開発の作業は要員間のコミュニケーションの比重が大きく、人員追加はメンバー間の意思疎通の効率を低下させるため、必ずしも納期短縮につながらない。「人月の神話」で有名な Brooks は、コストは実際に人数と月数の積に比例するが、仕事の進捗はそうではないとして、ソフトウェアの構築は複雑な相互関係における作業の遂行であって、コミュニケーションの労力は大きく、分担によってもたらされた個人の作業短縮時間を上回って非効率になるとしている²¹。「遅れているソフトウェアプロジェクトへの要因追加はさらに遅らせるだけだ」という命題が成り立つのである。

一方、マサチューセッツ工科大学 (MIT) のクスマノ教授 (Cusumano, M.) が2002年に行ったソフトウェア開発プロジェクトの国際比較のための調査結果²²によると、1ヶ月に1人のプログラマが書くプログラムの行数は、日本は469行

であり、インドや米国の2倍前後となっている。この数字から見る限りでは日本のプログラミングの生産性が高いことになる。また、システムが完成してから1年間に発見されたプログラム1000行あたりの不具合の数も日本は0.020であり、インドや米国より1桁小さい(表3参照)。Cusumanoはこの結果から日本のソフトウェアの生産性が優れていることを指摘し、その要因は日本の情報サービス産業における開発モデル＝ウォーター・フォール・モデルであるとしている²³。その結果、ソフトウェア生産の信頼性においても日本の情報サービス産業はHigh Levelにある企業の比率が他地域に比べて高くなっている(表4参照)。

表3 ソフトウェア開発の生産性国際比較

	India	Japan	US	Europe & other	Total
Number of projects	24	27	31	22	104
Median output (1)	209	469	270	436	374
Median defect rate (2)	0.263	0.020	0.400	0.225	0.150

(1) No. of lines of code/(avg. no. of staff* no. of programmer-months).

(2) No. of defects reported in 12 months after implementation /KSLLOC. We adjust this ratio for projects with less than 12 months of data.

表4 ソフトウェア開発の信頼性国際比較 ()内は比率(%)

	India	Japan	US	Europe & other	Total
Number of projects	24	27	31	22	104
Level of reliability					
High	8(33.3)	12(44.4)	8(25.8)	4(18.2)	0
Medium	14(58.3)	14(51.9)	20(64.5)	18(81.8)	0
Low	2(8.3)	1(3.7)	3(9.7)	0(0.0)	0

Cusumano, M., MacCormack, A., Kemere, C.F., Crandall B. [2003]より

しかしながら、この調査においてプログラマの生産性を計算する際に(1ヶ月に1人のプログラマが書くプログラムの行数を算出する際に)人月を基準に計算を行っている。しかし、1人のプログラマは1ヶ月に働く時間がすべての国において同一であるとは限らない。日本のソフトウェア開発の現場では残業が恒常化しており、休日出勤も多い。日本のソフトウェア開発においてトータルとして超過勤務時間が多いとするならば、人月を基準にしてソフトウェアの

生産性を比較しても意味がない。むしろこれはソフトウェア開発とその生産性を時間単位で計ることによって日本のソフトウェア開発の低生産性が浮き彫りにされるだけでなく、ソフトウェア生産の効率性の上昇＝労働強度の増大をプログラマに迫るものである。

Cusumano らによる人月計算とは別に、ソフトウェア生産における生産性の計量化は人ではなく人の生産物＝ソフトウェア自体の機能、品質を定量化していく「ソフトウェア工学」として発達してきた。

2、「ソフトウェア工学」：ソフトウェア生産の定量化と人時間計算

ソフトウェア工学 (Software Engineering) は、コンピュータのソフトウェアの開発方法を研究対象とする情報工学の一分野で、ソフトウェアの開発、運用、および保守に対して、体系的な定量的アプローチを適用する学問分野で、ソフトウェアの生産性の向上が叫ばれた1960年代末から1970年代にかけて成立してきた²⁴。ソフトウェア工学では、通常、開発対象となるソフトウェアの開発を(1) 要求仕様技術・設計技術、(2) 実装及び開発環境、(3) 品質保証技術、(4) 保守技術、(5) 再利用技術、(6) 計測・評価技術、(7) プロジェクト管理技術、(8) 文書化技術、などのフェーズに分けてこれらフェーズごとに典型的な課題と課題に立ち向かう方法を明確にしようとする。ソフトウェア自体、目に見えないものであってその機能や品質を外見的に判断することは難しいが、ソフトウェア工学はソフトウェア開発の工程を細分化することによってそれぞれの工程を定量化しようとする試みであった²⁵。この現代的な成果が1990年代に登場した組織におけるソフトウェア開発などの能力を向上させるための国際的な標準規格である CMM²⁶であり、ソフトウェア開発の能力を客観的に判断するための指標として利用されている²⁷。

谷口功 [2005] によれば、各レベルの概要は以下の通りで、

レベル1：プロセスが確立されていない初期段階

レベル2：特定のプロジェクトリーダーや技術者に依存している状態

レベル3：首尾一貫したプロセスを標準として持っている段階

レベル 4：標準化されたプロセスを定量的に測定し、洗練化していく状態

レベル 5：技術・要件環境の違いによって、標準プロセスを最適化して用いられる段階

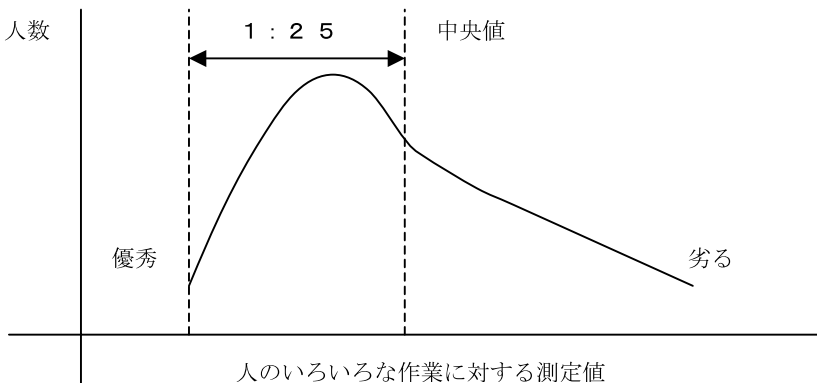
と規定され、組織の能力はレベル 1～5 で示され、各レベルで持つべきプロセスを規定している²⁸。ここで、組織の能力をレベル 1～レベル 5 へと向上させるためには、レベル 4 に見られるようにソフトウェア開発のプロセスの標準化と定量化の測定が不可欠であり、レベル 5 ではその定量化に基づいて標準プロセスを最適化することになるのである。

一方、ソフトウェア開発のプロセスを標準化しようにも、Brooks [1975] が早くから指摘しているように、プログラマの間には大きな生産性の差異がある²⁹。Demarco, T. と Lister, T. は1984年から1986年にかけて全米の企業から選抜されたプログラマ（92社からのべ600人）対象に行ったプログラミングコンテストの結果から

- ・最優秀者の測定値は、最低者の約10倍である。
- ・最優秀者の測定値は、平均的作業（中央値）の約2.5倍である。
- ・上位半分の平均測定値は、下位半分の平均の2倍以上である。

と結論を出している³⁰。

図8 プログラマの生産性の差異



Demarco, T., Lister, T., [1987]、邦訳、56頁より。

この分析においてはプログラムの優劣をソフトウェア開発の各種作業の所要時間の分布に基づいて分析し、図8のように右に長く尾をひく（作業の所要時間が長い＝生産性の低いプログラムが少数であるが存在する）という傾向を明らかにしている。

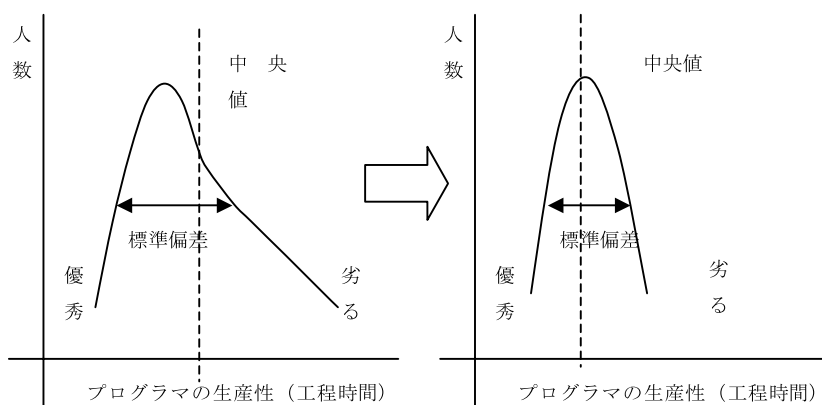
ソフトウェア工学自体は、プログラマの労働の成果物である目に見えないソフトウェア商品の機能や品質を計るものとして成立・発展してきたが、これを定量化しようとする場合、プログラマの能力を計量する必要が生じ、その能力の差異と標準化はプログラム開発工程の時間数で計られることになる。プログラマの作業はコンピュータ上で行われるため、その作業時間、コーディングなどのプログラミング作成作業、動作確認（テスト）、不具合の修正（デバッグ）などにかけた時間はコンピュータ上に秒単位で記録され、プログラマごとに日々算出されることになる。プログラマがソフトウェア開発のためにコンピュータで行った作業はすべてコンピュータに記録されるのである。そして、プログラマの能力ごとに時間単価が決められ、ソフトウェアの価値はソフトウェア生産にあたるプログラマの時間単価×総時間数の総和で算出されることになる。

もちろんユーザからの受注に対しては人月計算＝「人数×時間（月）」による人月単価で見積もりが行われているのであるから、情報サービス産業の企業内においては時間単価×総時間数＝人時間計算はソフトウェア開発のコストとして計上される。そこで、コスト削減のために時間単価を下げるか、総時間数を減らすかが課題になってくるが、時間単価の切り下げはプログラマの質の低下による開発時間の延長につながるために逆効果を生む場合が大きく、ソフトウェア開発の回転数を上昇させることも考え、総時間数の削減を目的としてプログラマの能力アップが至上命題となってくる。現実には図8の右に長く尾をひく著しく生産性の低いプログラムを生産から排除するのが手っ取り早く、そのため見直しが逐次行われることになる。

実際にも日本の「ソフトウェア工場」の典型でもある東京都大田区の（株）富士通アプリケーションズ³¹を取材したところ³²、プログラマ集団の全体の生産性の上昇（＝開発工程時間の短縮）のために、プログラミング作業を細かく時

間単位で計測すると同時に、それぞれのプログラムの開発工程時間の平均値の上昇よりも標準偏差を下げることによって（すなわち生産性の低いプログラムを生産から排除することによって）、結果的にプログラム集団としての生産性の上昇を目指していることが伺えた。富士通のソフトウェア開発のための総合システム開発体系 SDAS³³によれば、3年間でプログラマ生産性における標準偏差がそれ以前の40%に短縮したとされている。

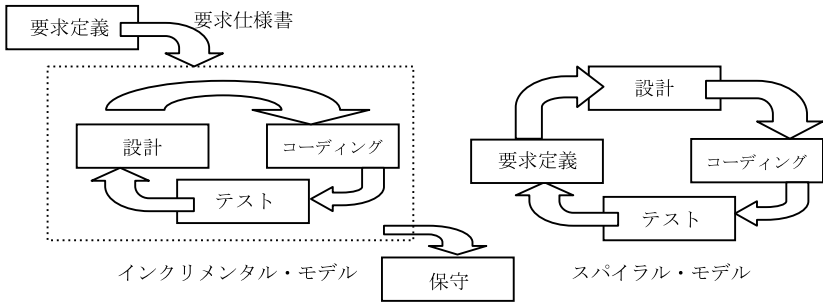
図9 プログラマ集団の生産性の改善



(株) 富士通アプリケーションズのヒアリング調査より

また「要求定義」、「設計」、「コーディング」、「テスト」の各段階における生産性上昇＝時間短縮を行うために従来のウォーター・フォール・モデルではなく、設計～コーディング～テストの段階を同時・並行的に行うインクリメンタル・モデルや、さらにスパイラル・モデルに近い形で仕事の流れ（業務フロー）を作っていた（図10参照）。

図10 インクリメンタル・モデルとスパイラル・モデルの概念図



(株) 富士通アプリケーションズのヒアリング調査および大場充 [2003] より作成

この場合、それぞれの段階が同時・並行的に行われることによってソフトウェア開発の期間が大幅に短縮されるだけでなく、各工程間のフィードバックが行われることによってプログラミングのミスが修正される。また、要求仕様作成の段階で各工程全体を見渡した仕様書作成といったエンジニアの高い能力が要求される。さらに「要求定義」まで含めたスパイラル・モデルにおいてはユーザのチェックを受け、問題点の抽出を受けて要求仕様自体の改善が図られる。いずれの場合にも「設計」、「コーディング」、「テスト」の各段階でモジュール化が行われるとしても、常に各工程間のフィードバックが必要なために組織的にも「近接した場」が求められる。プログラマが集中管理されることによってその作業が時間単位で計測することが可能になり、プログラマの開発工程時間の標準偏差を下げ、ソフトウェア開発の生産性を上げることも可能になる。一方、「要求定義」に関してはユーザとの接点・フィードバックなどが求められるため、この段階ではユーザとも「近接した場」が求められる。実際にもヒアリングを行った(株)富士通アプリケーションズは全国の富士通グループ内および富士通パートナー企業が受注したソフトウェア開発を集中して受託し、上記の開発作業を行っている。そしてプログラマは集中して時間単位で管理され、人時間計算によってソフトウェア開発の生産性向上に結び付けられているので

ある。

3、OSS：Linux、TRON とソフトウェア生産組織のオープン化

日本の情報サービス産業におけるソフトウェアの開発は、そのモジュール化と同時に、人時間あたりの生産性の向上を図るためにプログラマを集中管理する方向で進んでいる。前節で見たように西村・峰滝 [2004] の研究結果は日本の情報サービス産業ではモジュール化が効率的でないのではなく、「外注化」が効率的に機能していない、ソフトウェア生産のモジュール化に情報サービス産業の組織のモジュール化がマッチしていないことを示していると考えられる。ソフトウェア開発の生産性を上昇させようとするならば、それは「外注化」によるプログラマの集中管理が難しいことが要因である。また、これも前節で見たように日本の情報サービス産業の開発・生産体制が産业内・業界内での連関の多重性、同業者間での委託・受託の複雑な連鎖の元に成り立っているわけであるから、「外注化」は元請け企業のコストダウン（下請け企業の合理化）としてのみ機能し、ソフトウェア開発の生産性の上昇には結びついていないのである。

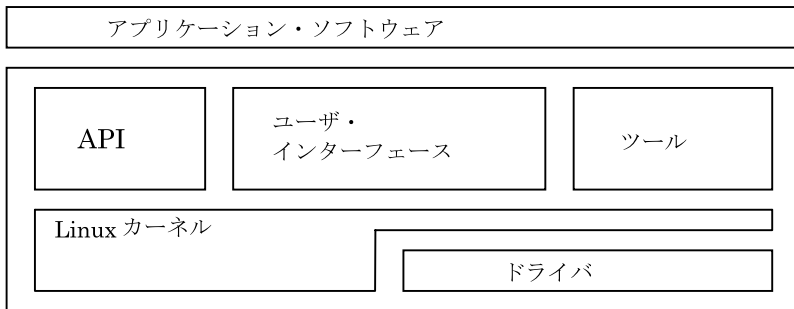
一方、インターネットなどのオープン・ネットワークの進展や、Linux に代表されるオープンソース・ソフトウェア（OSS：Open Source Software）を利用したソフトウェアの開発方式、そして日本が戦略的に位置づけている TRON の開発方式は、多数の企業によるコラボレート为前提としており、今後のソフトウェアの開発方式のあり方を特徴づけると考えられる³⁴。

フィンランドのヘルシンキ大学の学生 Linus Benedict Torvalds が大学在学中の1991年にパーソナルコンピュータ（Intel の80386 CPU の32bit PC/AT 互換パソコン上）で動く UNIX 互換 OS³⁵として開発した Linux は、GNU プロジェクト³⁶のコンパイラ GCC（GNU Compiler Collection）を利用して開発されたので、Linus は Linux のソース・コードを自由に利用できるように公開した。インターネットが本格的に普及し始めた時期でもあり、Linux はインターネット経由で世界中の開発者を引き付け、現在も改良とバージョンアップが加え続けられてい

る³⁷。Linux の普及以降、パソコン用の Web サーバアプリケーション Apache やプログラム言語 Perl、そして松江市在住のプログラマ、まつもとゆきひろ氏によるプログラム言語 Ruby³⁸など、インターネットを活用した同様の開発スタイルのソフトウェアが次々と開発された³⁹。

Linus が開発したのは OS のカーネル (kernel: 中核部分) だけであつたので、ユーザが利用するためには API (アプリケーションインターフェース) やユーザインターフェース、ツールや必要なアプリケーションを組み合わせて OS 全体を構築する必要があつた。Linux ディストリビューション (Linux distribution) はこれらをセットにして配布するもので、商用ディストリビューションなども登場し、マニュアルやサポート、商用ソフトウェアなども付属して最も盛んなオープンソースビジネスにもなっている⁴⁰。

図11 Linux ディストリビューション



また、東京大学の坂村健教授が1984年に将来のコンピュータ化された社会において協調動作する分散コンピューティング環境の実現を目指し TRON (The Real-time Operating system Nucleus) プロジェクトを開始した。TRON は情報処理用のコンピュータではなく、車のエンジン制御、工場の産業用ロボットの小型制御機器、携帯電話、ファクシミリ、デジタルカメラなどの機器に組み込まれる制御用のコンピュータのためのリアルタイム性を重視した OS⁴¹を中心としたアーキテクチャーである⁴²。TRON プロジェクトはトロン協会によって運営されているが、トロン協会は OS の仕様と、組み込み制御システムのアーキテク

チャー（どう作るか決めた仕様書などを含めて）を完全に公開している。そして TRON の実装は企業にまかせられ、これを使って誰がどのようなソフトウェアを作ってもいいし、また作ったものについて GPL のように公開を義務付けてもいないので、企業が開発に参加しやすいスタイルになっている。

従来のソフトウェア開発の作業には膨大な時間、巨額の投資が必要であった。そこでソフトウェアが簡単にコピーされるなら企業も望みだけの利益を得ることができなくなるので、企業は OS などのソース・コードの技術情報を隠すようになり、法的にコンピュータの内部情報は知的財産であるとして著作権で守られようになる⁴³。ソフトウェア生産のモジュール化が進みながらも情報サービス産業がソフトウェアの開発を集中化して行うのは生産性の問題と同様、このような社会的な背景がある。また、コンピュータやソフトウェアの普及には互換性を進めていくために規格の標準化が必要であるが、コンピュータの場合はこれが公的な場で決められるのではなく、Microsoft の Windows に典型的に見られるようによく売れたためにみなそれぞれに従うというデファクト・スタンダードが力を持ち、結果として一つの企業が市場を占有するとい覇権構造に直結してきた。

オープンソース（Open Source）という言葉の名付け親⁴⁴でもある Raymond [1997] によれば、このようなソフトウェアの開発方式はトップダウン＝「伽藍（大聖堂）型」の開発方式と呼ばれ、スケジュールと役割分担を明確にし、高層ビルを建設するように開発が進む。まさにウォーター・フォールの開発モデルなのである。そこで開発の途中でプログラマを増加してもそれ以上にコミュニケーションの手間が増えて全体の能率が上がらない、すなわち Brooks の法則が現出する。また、動作確認（テスト）や不具合の修正（デバック）など人数を増やすことで能率の上がる作業は、結局は人材コストの上昇につながる。これに対して、Linux のようなオープンソースのソフトウェアの開発のスタイルは「バザール（市場）型」として、近年情報サービス産業の関心を集めるようになった⁴⁵。また、パソコンで動作することで市場規模の拡大も期待できるため、オープンソースやこれによって開発されたソフトウェア（OSS）を情報産業全体でも

支援するようになり、多くのベンチャー企業を生み出している⁴⁶。

OSSによって開発されたプログラムはソース・コードも含めてインターネット上で公開され、修正・改良が加えられ再配布されるのであるから、ソフトウェア生産のモジュール化をオープン・ネットワーク上で、多数の企業、開発者の参加によってオープンな形態で進めることになる。日本の情報サービス産業の中心である受注ソフトウェア開発においてこのようなOSSの成果＝パッケージを使ったソフトウェア開発、カスタマイズ化が進展していく可能性は大きい。ソフトウェア生産のモジュール化に加え、このようなOSSによるソフトウェアの開発方式＝ソフトウェア生産組織のオープン化は、日本の情報サービス産業に特徴的な「外注化」の新しいスタイルを作り、生産のモジュール化・オープン化と産業内・業界内での委託・受託の関係＝ベンダーロックインを解除したモジュール化された組織として地域の中小情報サービス企業をマッチングさせる可能性がある。次節では日本の情報サービス産業におけるその具体的な実態を探っていく。

第3節 ソフトウェア生産組織のオープン化と情報サービス産業

1、日本のOSSをめぐる状況とOSSの「産業化」の可能性

Linuxに代表されるオープンソース・ソフトウェア（OSS）や、これによる新たなソフトウェアやシステムの開発はインターネットも利用して自主的に参加する人材が集まり、自由に利用できるソース・コードと、迅速な対応が可能となる。また統一した規格や標準化もオープンな場で議論し、決めることが可能であり、開発においても中小企業であってもOSSに機能を付加したり、システムを構築するなどのビジネス機会を広げることになる⁴⁷。さらにOSSは導入する企業にとってもコスト・ダウンのメリットがあり楽天市場などWebでサービスを行う多くのオンラインショップはOSSを組み合わせてシステムの構築を行っている⁴⁸。

京都大学経済学部の末松千尋教授は、オープンソースの“どこで金銭を稼ぐ

か？”を問題として

「製品（ソフト）は無料でも、サービス（展示・説明、受発注処理、決済、配送・伝送、品質保証、メンテナンス、サポート、インテグレーション、コンサルティング、教育、講演、およびそれらにより確立するブランドの活用など）を事業として課金することは全く自由だし、逆に製品が無料になれば、サービスがより重要となり発展することは十分に起こり得ることである。現実には、ディストリビューションという、全く新しいサービス業態が生まれ育っている。」

としている⁴⁹。

そこで、日本でも経済産業省や総務省を中心に、OSSを使ったソフトウェア開発の支援策が打ち出されている。OSSを推進する組織としては、独立行政法人 情報処理推進機構（IPA）が事務局を務め、NEC、日立製作所、富士通、日本IBM、NTTデータなどが幹事となっている「日本OSS推進フォーラム」⁵⁰があるが、経済産業省は2006年1月、オープンソース・ソフトウェア推進組織「オープンソースソフトウェア・センター（OSSセンター）」を設立した⁵¹。OSSセンターは上記国内主要ITベンダーと協力し、OSSの検証環境の整備やセキュリティ・ホールなど技術・事例情報の収集および提供を行う⁵²。また、これまでのソフトウェア開発がMicrosoftのOS市場独占に見られるように米国中心に回ってきたことに対して、アジア地区におけるOSSの国際協力を推進する目的から、上記の「日本OSS推進フォーラム」が中国や韓国と連携して「北東アジアOSS推進フォーラム」を第1回（2004年4月：北京）、第2回（2004年7月：北海道）、第3回（2005年12月：ソウル）と開催している⁵³。

また、電子政府・電子自治体システムの普及を目指す総務省は、2004年の4月に「電子政府・電子自治体におけるOS選定のあり方について」⁵⁴の報告においてOSの選択肢としてLinuxをリストに入れ、他の商用OSと共に評価を行い、Linuxを選択肢として事実上公認している。これに先立ち経済産業省も2003年8月に「オープンソース・ソフトウェアの利用状況調査 導入ガイドライン」⁵⁵を公表し、LinuxをはじめとするOSSを、一つの選択肢として積極的に活用する

べく導入検討ガイドライン及び法的課題の整理などについて検討を行い、IPAを通じての学校教育現へのLinuxPC導入実証実験（2004年11月～）⁵⁶や本省でのLinuxPC導入プロジェクト（2004年度）を進め、2005年度は自治体を対象に実用化検証を開始している⁵⁷。さらに政府・IT戦略本部が2005年2月に決定した「IT政策パッケージー2005」⁵⁸においてOSSに関する事項として、電子政府におけるオープンソースソフトウェアの活用促進（総務省、経済産業省）や、オープンソースソフトウェアを活用したIT人材の育成（内閣官房、文部科学省、総務省、経済産業省）などが位置づけられている。

このようにOSSの導入・普及は政府（経済産業省や総務省を中心として）と国内主要ITベンダー＝大企業を中心に産官学連携で進められているが、OSSの行政や企業における導入といった視点が中心であり、前節で見たLinuxなどのOSSの開発方式＝バザール型を日本の情報サービス企業のソフトウェア開発方式や、企業間組織のあり方にどう取り入れるか、さらに本稿の課題である地域の情報サービス産業の技術革新への対応や需要拡大にいかに関わりつづけるかという視点はあまり見られない。例えば、総務省の電子自治体へのOSS導入の課題は予算上の効率化の側面と同時に、セキュリティ確保やサポート体制が中心としてあげられ⁵⁹、また経済産業省のレポート「導入ガイドライン」においてもコスト面や、導入におけるライセンス契約、GPLに関する法的問題などが中心になっている⁶⁰。これは、「日本OSS推進フォーラム」のメンバーがOSSの開発者というよりはむしろユーザの側面が強い（あるいはOSSを囲い込んで自社のハードウェアや自社開発ソフトウェアと組み合わせてライセンス供与する⁶¹）大手ITベンダーから構成されているのと同時に、これまで見てきたようにOSSが商用ソフトウェアの開発・市場化のアンチテーゼとして始まったフリー・ソフトウェア運動に起因していることが大きな原因として考えられる。実際に大学を中心とした研究者の間には科学的知識の産物であるソフトウェアがライセンスという特定の企業の管理下に置かれるのは間違っており、その成果は公開され自由に活用されるべきである、との考え方がその核にある⁶²。それ故に、OSSによるソフトウェア開発がいかに情報サービス産業に市場と利益創造をもたらしながら

ら、その組織形態の変化、すなわちウォーターフォール・モデルを基準とした「伽藍型」からソフトウェア生産のモジュール化・オープン化と組織のオープン化がマッチングした「バザール型」への転換に結びつくのかこそが解き明かすべき重要な課題である⁶³。

2、地方自治体の OSS 活用と地域の情報サービス産業

このような OSS の導入が行政や企業の効率化という観点でだけ進められるならば、OSS 自体の市場は拡大しても、それは情報サービス産業の「バザール型」への転換にはつながらない。例えば中央政府レベルで進められる電子政府・電子自治体の取り組みは IT 企業がコンソーシアムを組んで参加・入札する方法で進められているが、実際には IT ゼネコンと呼ばれる大手 IT ベンダーを中心に「企業連合」が生まれ受注を競い合っている。特に電子政府に必要なセキュリティ技術、認証技術、暗号化技術などは高度な技術的・専門的知識が要求されると同時に、「個人情報保護法」によって大手 IT ベンダーはプログラム開発者を自社＝東京に集中させる傾向がある。第 2 節であげた東京大田区の「Java 工場」などはその典型である。

一方、地方自治体レベルでの OSS の導入は、自治体の IT システムの導入/運営コスト削減とあわせて、地元の情報サービス産業振興も目的に進められている。代表的なのは長崎県における「電子県庁システム」の構築で、長崎県は 2001 年に民間から CIO（Chief Information Officer：企業において情報に関する資源を統括する最高責任者）を招聘し、文書管理（決裁）、申請、入札の 3 機能を持つ電子県庁システムの基盤技術にオープンシステムを採用し、地場の中小ベンダーをシステム開発に参加させることを促進した。長崎県が打ち出した戦略「ながさき IT モデル」では、①システムの詳細な仕様書を県の職員が中心となって作成する、②プログラム開発を外部委託する際は、一案件の予算を減らすために分割発注する、③システム構築時に利用する技術を決める、が骨子となっている⁶⁴。この方式によって従来の大手 IT ベンダーによる不透明な提案と見積もりを排除し、現場に必要なシステムを安いコストで導入することを可能にした。

長崎県庁情報政策課での取材⁶⁵によると、

県の電算関係の経費は平成14年度9.91億円から平成15年度6.94億円と大幅に削減。

また、分割発注方式によって地場の情報企業もシステム開発に参加することが可能になり、その結果

平成14～15年度 100件中48件を地場企業が直接受注
(金額ベースで15.1%)

平成16年度 96件中73件を地場企業が直接受注
(金額ベースで32.7%)

と、地域の情報企業の参入、育成にも大きな成果をあげている。

このことであった。また、担当職員による仕様書作成の過程も

第1ステップ システム画面のラフスケッチ → WEB デザイナーに仕上げ

第2ステップ 画面デザインの確定 → SE に DB テーブルのフォーマット

第3ステップ DB テーブルの確定 → 設計書（仕様書）の作成

と、それぞれの過程で地元企業に委託し、仕様書を作成する過程に参加させている。地元企業は受注を増やすだけでなく、今まで大手 IT ベンダーの「役割」であった企画・仕様書策定の過程に参入することによって企画力、プロジェクトマネジメント力をアップさせているのである。

長崎県は2004年の11月にはメインフレームを8年間でLinuxに移行する計画を発表し、同年には特定非営利活動法人 OSCAR アライアンスからビジネスにおけるオープンソース活用の推進において、その年に最も活躍した企業・団体・個人を表彰する「2004年オープンソース・ビジネス・アワード」を受賞⁶⁶、また2005年10月21日には同県の電子県庁システムの「休暇システム」と「WEB 職員録システム」、「文書保管システム」の3システムのソースコードを公開した⁶⁷。OSSを自治体システムに利用するだけでなく、OSSの開発方式に基づき成果物を積極的に公開することによって、システム開発に参加した地元情報サービス産業への市場拡大の可能性を与えるものである。

また、北海道江別市では小麦製品や乳製品、食肉加工品などの市の情報を Web

で紹介するサイト「江別ブランド事典」⁶⁸において、従来の検索エンジンのような、カテゴリやキーワードのみで情報を探し出すシステムと違い、人に話しかけるような感覚で、知りたい情報が探せる「自然言語ナビゲーション」を札幌学院大学電子ビジネス研究センターと共同で開発し、オープンソース化している。

新潟県からオープンソース・ソフトウェアを全国に向けて発信しようという有志によって2005年05月02日に設立された NPO 法人・新潟オープンソース協会では「Niigata Linux」（オープンソースの開発環境を簡単に構築できる CD、Linux ディストリビューションであると同時に、Windows 上に PHP と PostgreSQL, Apache を導入するインストーラ）と「NPO 総合管理システム」（NPO 法人向けの業務システムをオープンソースを活用して構築する Web アプリケーション）を開発し、「日本で1番オープンソースが盛んな県は新潟県！」を目指している⁶⁹。新潟オープンソース協会の湯川高志・長岡技術科学大学助教授は「オープンソースは下請けからパッケージ・ベンダーへと脱皮するための足がかりになる。ライセンス料が無料のため資本が少ない企業でも取り組みやすい。たとえパッケージが売れなくとも金銭的な損失は小さく、労力がかかったというだけですむ」と OSS の利点を強調している⁷⁰。

このように、地方自治体や NPO レベルでは OSS の導入を地域の情報サービス企業の育成、市場拡大とあわせて振興している事例がある。自治体のシステム構築において自治体はユーザ＝発注者として地元情報サービス産業の開発した OSS を採用するだけでなく、その成果物が OSS としてインターネット上に公開されれば地元の情報サービス産業にとって全国的に市場を拡大できるチャンスが与えられるのである。特に Linux やなどの OS やプログラム、PostgreSQL などのミドルウェアレベルではなく、これらの OSS を使って開発されたシステム、パッケージをさらにオープン・ソースにすることによって末松教授が指摘するようなディストリビューションモデルを作り、市場の創造・拡大を図っているのである。そして、長崎県の事例に見られるように、県の職員が仕様策定、すなわちソフトウェア開発における「要求定義」や「設計」のプロセスに参加

し、その仕様に基づいて分割発注すなわち「モジュール化」による「外注化」を行うならば、前節で見たような大手 IT ベンダーによるウォーター・フォールの開発モデルを前提としたトップダウンの開発方式＝「伽藍（大聖堂）型」（あるいは「ベンダーロックイン」）を解除し、分割発注されたモジュールを地元の情報サービス産業が受注し、ソフトウェア開発に参加できる可能性を与え、実際にもその成果が表れている

ただしこの過程と成果が可能になるのは、これも長崎県の事例に見られるように、行政の職員が「要求定義」や「設計」のプロセス、すなわちソフトウェア開発の上流工程に参加することが前提である⁷¹。ソフトウェア開発における「要求定義」や「設計」はユーザのリクエストに基づいて要求を獲得・分析し仕様化するだけでなく、ソフトウェア開発の管理全体を行う、すなわち前節で見たような下流工程におけるプログラマの集中的な時間単位での管理、そして人時間計算に基づいた見積もり作成までを行う非常に高度で複雑な作業である。上記のソフトウェア受注額の削減効果には、県の職員がこの「要求定義」や「設計」に関わった「人月」は算定されていない。実際に経費の削減効果とも総合的に比較したコスト・パフォーマンスは測定不能であるが、行政による間接的な開発援助によって可能となる「バザール（市場）型」のソフトウェア開発方式である。さらに企画・仕様書策定の過程に地元の業者が参加することによってウォーター・フォールの開発モデルに欠けているフィードバックが行われ、コミュニケーションとコラボレーションが円滑に行われることによってソフトウェアの開発・製造における生産性向上と地元の情報サービス産業の企画力、プロジェクトマネジメント力のアップを達成しているのである。

3、島根県の情報サービス産業と OSS 活用の可能性

以上のように OSS の進展とその活用は、日本の情報サービス産業の中心である受注ソフトウェア開発において大手 IT ベンダーを中心とした産業内・業界内での委託・受託の関係をオープンなものに変えていく可能性はあるが、それはソフトウェア開発のモジュール自体を設計する上流工程のあり方に依存するこ

とになる。OSSの開発スタイル＝「バザール（市場）型」のソフトウェア開発方式においてコミュニケーションとコラボレーションの作業はより増加するため、ウォーターフォール・モデルを前提とするような「外注化」においては「人月の神話」に見られるような生産性のパラドックスを生じさせ、第1節で見たように日本の情報サービス産業の「外注化比率」がソフトウェアの生産性に対して負の説明変数になるのである。これを避けようとするならば、前節の「Java工場」に典型的に見られるように、下流工程のプログラマを集中管理して時間単位で作業効率を上げソフトウェア開発の生産性を上げるか、長崎県の事例に見られるようにコミュニケーションとコラボレーションを代替するプロジェクトマネジメントのプロセスが必要となってくる。

筆者が2005年1月～3月にかけて、島根県商工労働部・（財）しまね産業振興財団と共同して実施した「島根県情報サービス企業調査」⁷²においてもこの傾向が読み取れる。

まず、業務内容についてはそれぞれの回答企業における業務の比重を全体で単純平均した結果を比較した場合、全体的にも「受注システム開発」が42%と圧倒的に大きいことがわかった。これは全国的な調査（「特サビ」で46.7%）と同じ傾向を示している。一方「パッケージソフト開発」は僅かに3%である。特にこの傾向は小規模企業で高く、またタイプ別で見ても下請型（61%）、中間型（74%）で高くなっている。ソフトウェア関係の業務では、次いで「業務システム運用管理」が12%を占めているが、これは「受注システム開発」とは対照的に規模の大きい企業、そして自立型（21%）、元請型（17%）のタイプの企業で比重が高くなっている⁷³。

次に、受注先では同業の「情報サービス業」が圧倒的に多く（37%）、「行政・外郭団体」（22%）、「建設業」（9%）がこれに続く。「情報サービス業」については、この分野に特徴的な産業内・業界内での連関の多重性、同業者間での委託・受託の連鎖によるものであり、また、小規模企業ほど同業種からの受注の割合が高く、大規模企業では低くなっている。全国的には情報サービス以外の他サービス業や金融、保険業への売上げが大幅に伸びているのに対し、島根

県ではこの分野での受注が、「その他サービス業」(4%)、「金融・保険業」(1%)と極端に少ない。この分野の情報化・IT化が極端に遅れているわけではないので、受注を県外大手企業に食われていることを示している。一方、建設・不動産業などへの売上げは全国的にも減少しているのであるから、「行政・外郭団体」ともあわせてこの分野への受注にいつまでも依存するわけにはいかないであろう。特に大規模企業、そして自立型や元請型の企業においても「行政・外郭団体」や「建設業」の比重が高い点は特徴的である。

そして注目されるのが、島根県の情報サービス企業の人材と教育の状況で、回答企業が自社に不足しているスキルとして筆頭にあげたのがプログラム作成の工程に関する技術ではなく、「プロジェクトマネジメント力」(66%)であった。企業規模別では売上高が3千万円～1億円未満の中堅企業が(83%)、そしてタイプ別で元請型が低い(33%)のに対して自立型や下請型において高い(69%)。既に外注化の体制とノウハウができあがっている企業に対して、成長過程にあり組織の分化も同時進行している企業においてはプロジェクトも流動的であり、プロジェクトにおけるコミュニケーションやコラボレーションを遂行する能力や人材を切望しているのである⁷⁴。

この「プロジェクトマネジメント力」を従来どおりに大手ITベンダーに依存するのであれば、地方の情報サービス産業はウォーターフォール・モデルを前提としたベンダーロックインを解除できないだけでなく、プログラム工程でさえ東京の「Java工場」に集中化させてしまうことになる。特に「個人情報保護法」などもあって、プログラム開発は集中化する傾向にある⁷⁵。

OSSの開発、あるいはOSSを活用したソフトウェア生産は地域の情報サービス産業においても直接受注を拡大し、またディストリビューションモデルに見られるようなOSSのオープン性=成果が市場に公開されることによって、全国的に市場を創造・拡大する可能性がある。島根県においても日本医師会の進めるORCAプロジェクト⁷⁶に参加し、OSSである「日医標準レセプトソフト」⁷⁷の開発に参加し、これを導入した医療機関に対してハードウェアの手配・設置、操作説明、リモート対応まで総合的なサポートサービスを行うことによって全

国的に受注を拡大している企業も存在する⁷⁸。この場合、日本医師会が上流工程である「要求定義」や「設計」のプロセス、すなわちプロジェクトマネジメント力を発揮したと捉えられる。前項の長崎県の事例も同様であるが、現状ではこのプロセスはパブリック（公的）な制度・支援によって可能になっているのが現状である。これが文字通りの「バザール（市場）型」のソフトウェア開発方式に転換するためには、パブリック（公的）な制度・支援を前提とするだけでなく、情報サービス産業同士のコミュニケーションとコラボレーションが必要であり、これは島根県の情報サービス産業においても重要な課題である。

おわりに

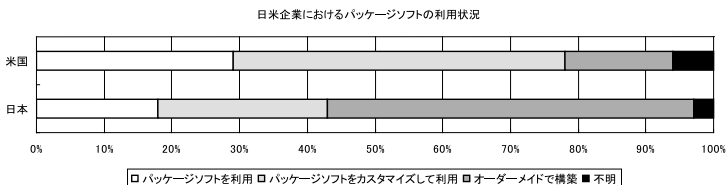
日本の情報サービス産業に特徴的な産業内・業界内での連関の多重性、同業者間での委託・受託の複雑な連鎖は、ソフトウェア開発においてはウォーターフォール・モデルに典型される上流工程から下流工程の流れとして現れ、地域の情報サービス産業は下流工程を受託する構造、ベンダーロックインを解除できないでいた。そしてソフトウェア生産を効率化しようとする試みは、人月の神話を超えてさらにプログラマの能力を計量しようとする人時間管理（あるいは分、秒単位の管理）へと進んでいる。これが可能になるのはプログラマがソフトウェア開発のためにコンピュータで行った作業はすべてコンピュータに記録されるからであり、資本による労働の管理・包摂が最高度に進む、「情報資本主義」段階を特徴付ける過程である。

一方、OSSに代表されるソフトウェア開発方式はオープン・ネットワーク上で、多数の企業、開発者の参加によってオープンな形態で進められ、また開発されたプログラムはソース・コードも含めてインターネット上で公開され、修正・改良が加えられ再配布される。これはフリー・ソフトウェア運動に見られるように即時的にソフトウェア開発の自由な管理、非市場化を進めるものではないが、情報サービス産業にとって市場や利益創造のもたらし方を変えるものであり、そしてその組織形態の変化やさらに労働のあり方、コミュニケーショ

ンとコラボレーションを含めた労働管理のあり方をオープン化させる可能性を持つ。これは「情報資本主義」段階における労働のオルタナティブの可能性を与えるものである。これはあくまで可能性であり、資本の運動は、前者の可能性、すなわち資本によるソフトウェア労働者の集中化＝オープン化と対極にあるクローズド化と人時間管理による管理強化に進むのであるが、後者の可能性は労働の管理・包摂が最高度に進む情報サービス産業の中から現出するのである。

【脚注】

- 1 1990年代の米国の産業が「オープン・ネットワーク」技術の優位性をIT投資として企業活動に応用し、SCM～BtoB～e マーケットプレイスと続くネットワーク型の生産～流通システムを完成させたのに対し、2000年代に入り日本は「オープン・ネットワーク」技術を継承しながらも「ユビキタス」技術の優位性を背景に「系列」を強化してきたのである。ユビキタス、あるいはユビキタス・ネットワークは、インターネット中心の米国のネットワーク技術の「標準化」に対する日本の「標準化」戦略であるが、このことによってユビキタス・ネットワークを導入する、すなわちIT投資を行う企業の生産過程の「標準化」も併せて行われつつある。ユビキタス・ネットワークによる生産～流通ネットワーク化が、「系列」を強化する日本型の「ネットワーク」型企業関係を完成しているのである。
- 2 総務省統計局の2000年度取引基本表（104部門）においては、情報通信機器を生産している部門として「051 電子計算機・同付属装置」、「052 電子通信機器」、通信のサービスを提供している部門として「086 通信」、そして情報サービスを提供している部門として「095 広告・調査・情報サービス」として分類されている。
- 3 経済産業省 [2004], [2005] 参照。
- 4 経済産業省 [2005] より。
- 5 社団法人情報サービス産業協会編 [2005] より。
- 6 これに対して米国では約8割の企業がパッケージソフトを利用するか、またはそれをカスタマイズして利用しており、逆に米国での受注ソフトウェアで企業の情報システムを開発している割合は2割以下である。



総務省 [2003] 資料編、資料1-2-10より。

- 7 経済産業省で情報サービス産業を所轄する情報処理振興課の嶋田ほか [2004] による

- と「ベンダーは比較的固定されたユーザを顧客とした上で、少数の大手企業が多数の中小企業を縦割りのに囲い込みつつ、多重的な下請構造を形成してきた」としている（嶋田ほか [2004]、368頁より）。
- 8 プロジェクトの大きさを表現する場合や、1人当たりの費用月額（人月単価）を掛け合わせてシステム開発の費用算定/見積りに利用される。工数計算における基礎は標準的スキルを持つ人間（エンジニア）が単位時間（1カ月）に行う仕事量だが、ソフトウェア開発においてはそれを定義する一般的な基準が存在しないため、同じ「1人月」といっても開発ベンダーによって実際に可能な仕事量は異なる。また、「月」の定義も一定でなく、22日以上稼働をもって1人月としているところもあれば、週4日程度の作業を1人月としているところもある。さらに1人月の単価もバラバラでベンダーによっては10倍以上の開きがある。
 - 9 実際に各工程の人月や総人月は、ファンクションポイント法（ソフトウェアの処理内容の複雑さなどから数を付けていきすべての機能のポイントを合計して規模や工数を導き出すもの）で見積もったり、画面や帳票などの入出力数から見積もったり、経験値から見積もったりなど、各社はさまざまな見積り方法で計算する。いずれの方法であっても、最終的には「人月」としてまとめられ、その人月に、SEやプログラマの月当り社内単価を乗じて見積もり金額を計算する。そこで使われる社内単価は、SEの職位やレベルで異なる企業もあれば、そうではない企業もある。また、人件費や管理費、そのほかの経費なども企業ごとに異なる。そして、下請けのSE、プログラマを必要とするような場合は、下請けとの契約金額に何%かを乗じて合計金額を計算する（馬場 [2002]、参照）。
 - 10 情報サービス産業が階層構造になったのは大手企業や官庁がソフトウェア開発を外注する際に大規模なシステム全体の一括発注を行い、そのため中小企業や実績のない企業は契約金額の大きな入札案件には参加することもできないことが多かったことが背景にある。したがって大型案件は大手企業が受注することが多くなるが、実際の業務は大手企業の系列下にある中小企業に流されることになる。鶏が先か、卵が先か、いずれにせよ情報サービス産業の階層構造とウォーター・フォール・モデルはこのシステムの下で再生産されているのである。
 - 11 また、ソフトウェア開発・調達プロセス協議会 [2001] によれば、国内の大手コンピュータ・ハードウェア・メーカは従来から各社独自のハードウェア仕様をもって、その仕様に熟達したソフトウェア開発企業とともにハードウェアを利用するユーザを囲い込んでしまおうとする傾向が強かったことを指摘している。これに加え、ユーザ側もまた、それぞれ独自の商慣行を尊重し、特定のハードウェア・メーカと好んで長期的な商取引を行い、ユーザに必要な仕様をメーカ側に依存してしまう体質になってしまったとしている。
 - 12 前川徹 [2004]、133頁参照。
 - 13 西村・峰滝 [2004]、156～157頁参照。
 - 14 ソフトウェア開発のプロセス改善（SPI:software process improvement）のための指標として、全世界で採用されている。組織（企業・チーム）のソフトウェア・プロセスの成熟度を示すリファレンスモデル。
 - 15 ソフトウェア開発・調達プロセス協議会 [2001] 参照。
 - 16 IBMのSystem/360はICを採用すると同時に業界で初めに本格的なOS（OS/360）を搭載していた。また、System/360は互換性をもったコンピューター・ファミリーで、

ソフトウェアはもちろん、周辺装置はどのモデルでも共通に使えるようになった。ハードウェアの仕様の違いは OS が吸収してくれるため、ある OS 向けに開発されたソフトウェアは、基本的にはその OS が動作するどんなコンピュータでも利用できることになる。その結果、ユーザの生産性は大幅に向上し、上位機種への移行も容易になった。そして、コンピュータ・ハードウェアを中心としたシステムは汎用化することによって本格的な量産化が可能になり、市場を拡大していくことになった。

- 17 野田 [2004] a 参照。
- 18 西村・峰滝 [2004]、178頁～204頁参照。
- 19 米国の Sun Microsystems 社が開発したオブジェクト指向型のプログラミング言語で、インターネット/イントラネット環境などにおいて、特定のハードウェアや OS に依存しないアプリケーションを作成することを目的として開発された。また Java アプレットは、Java プログラムのコードを Web サーバから Web クライアントに送信し、これをクライアント側コンピュータの Web ブラウザ上で実行することを前提としたプログラム・モデルで、プログラムが実行時にダウンロードされるため、実行プログラムをサーバ側で集中管理することが可能であり、かつ Java 対応の Web ブラウザが利用できる環境ならプログラムを実行できるので、さまざまな種類のコンピュータがクライアントとして存在している環境などでは効果が大きい。
- 20 さらに、2000年代に入って米国で始まった IT バブル崩壊～IT 不況によるコンピュータ需要の落ち込みや通信業界の過剰投資などもあって、コンピュータ・ハードウェア産業や通信産業もソフトウェアやサービス分野に重点を移しており、またインドや中国の IT 産業のこの分野での追い上げもすさまじい。コンピュータ・ハードウェアでは既に中国や東南アジアを中心とした部品製造の海外移転と、世界規模での生産から流通に至るまでのネットワーク化が進んでいるが、ソフトウェアの分野でも米国からインドへの大量のアウトソーシングに見られるように、オフショア（海外生産拠点）化は着実に進行している。このことは、情報サービス産業の成長とあわさったグローバルな供給能力の拡大によって、情報サービス産業も国際的な競争からは逃れられないことを意味している。
- 21 Brooks, F. P. Jr. [1995]、邦訳14頁～23頁参照。
- 22 インドからは Motorola India Electronics, Infosys, Tata Consulting, Patni Computing, 日本からは日立、NEC、IBM 日本、NTT Data、SRA、松下電器、オムロン、富士ゼロックス、オリンパス、米国からは IBM, Hewlett-Packard, Sun Microsystems, Microsoft, Siebel Systems, AT&T, Fidelity Investment, Merrill Lynch, Lockheed Martin, TRW, Micron Technology, 欧州からは Siemens, Business Objects, Nokia などが調査対象となった。
- 23 Cusumano, M., MacCormack, A., Kemere, C. F., Crandall, B. [2003], p.5～p.7参照。また Cusumano, M. [2003] によれば日本で調査したソフトウェア開発プロジェクト30件のうち、半数以上の16件のプロジェクトが純粋なウォーター・フォール・モデルを採用していたとなっている (Cusumano, M. [2003]、p.37参照)。Cusumano による日本の情報サービス産業のウォーター・フォール・モデル理解は正しいが、人月計算についてはあまりにも古典的である。
- 24 ソフトウェア工学という言葉が一般的に広まるきっかけになったのは1968年10月に NATO (北大西洋条約機構) によって開催された会議だと言われ、ここで NATO の関係者たちが大規模で複雑なソフトウェア生産と信頼性の問題に直面していたからだ

- 言われている（前川 [2005]、77頁より）。
- 25 日本でのソフトウェア工学の最前線は、2003年度から奈良先端科学技術大学院大学と大阪大学が中心となっている EASE (Empirical Approach to Software Engineering: ソフトウェア工学へのエンピリカルアプローチ <http://www.empirical.jp/> を参照)、と独立行政法人情報処理推進機構 (IPA) にソフトウェア・エンジニアリング・センター (SEC) を設置して民間におけるソフトウェア開発の事例を収集・分析し、実践的な技術開発・人材育成を行う産学連携拠点のプロジェクト (<http://www.ipa.go.jp/about/jigyoshokai/soft.html> を参照) である。
 - 26 カーネギーメロン大学で開発されたソフトウェア開発能力の成熟度を測る「ソフトウェア能力成熟度モデル: SW-CMM」が元祖である。ここから派生して、ソフトウェア調達の能力を測る「SA-CMM: Software Acquisition CMM」や人材開発能力の成熟度モデル「P-CMM: People CMM」、統合プロダクト開発成熟度モデル「IPD-CMM: Integrated Product Development CMM」などさまざまな CMM が生まれてきた。そのため1999年に、これら種々の CMM を統合した「CMMI: CMM Integration」が発表された。これらの開発は、米カーネギーメロン大学ソフトウェア工学研究所 (SEI: software engineering institute) で行われている。
 - 27 日本においても経済産業省主導で2001年1月から「ソフトウェア開発・調達プロセス改善協議会」において日本版 CMM の検討が始まっているが、品質管理に関する国際規格 ISO9000シリーズや環境管理に関する国際規格14000シリーズ同様に、認証を受けること自体が自己目的化されるという批判が情報サービス産業界から強く、日本版 CMM 策定の動きは鈍化している（前川 [2005]、87頁～89頁参照）。
 - 28 谷口功 [2005]、また Pressman [2005] などを参照。
 - 29 Brooks [1975]、邦訳26頁参照。
 - 30 Demarco, T., Lister, T. 邦訳 [1987]、邦訳54頁～62頁参照。
 - 31 富士通グループ内および富士通パートナー企業のアプリケーション・ソフトウェア開発における開発専業会社としてプログラミングを集中して受託する「下請け企業」の統合会社。Java 言語によるアプリケーションシステムの開発が中心で、システム構造設計工程～結合テスト工程までを受託し、通称「Java 工場」とも呼ばれている。<http://jp.fujitsu.com/group/fap/> を参照。
 - 32 2005年12月19日実施。富士通アプリケーションズからは佐藤代表取締役社長が対応。
 - 33 富士通が1987年に開発したソフトウェアの生産性向上を実現するため総合システム開発体系 SDAS (System Development Architecture & Support facilities) よると開発期間短縮、「短期間・高品質」のシステム開発を実現するとともに、「オープン性・国際標準」「ライフサイクル全般でのシステム最適化」「エンジニアリングとマネジメントを両輪とするプロジェクト遂行」によって、システム開発期間を従来と比べ、概ね半減することが可能になるとしている。
 - 34 小山・竹田 [2001] や池田・林 [2002]、また西村・峰滝 [2004] でもオープンソース・ソフトウェアの重要性が強調されており、通信プロトコルやオペレーション・システムのようにネットワーク外部性が大きく、かつ安定性が求められるソフトウェアについては、非営利で開発し情報を無償で公開することが望ましいとしている。ただ、これらの指摘には非営利や無償での開発の可能性は描かれてはいても、その中でのソフトウェア開発が産業として成立・発展する可能性については触れられていない。
 - 35 UNIX は AT&T B のベル研究所で開発された OS で、ベル研究所の当時の親会社 AT

- &Tは、独占禁止法によりコンピュータ産業への進出を禁止されていたこともあって、UNIXのソース・コードは世界中の大学や研究機関に非常に安価な値段（メディアのコピー代だけ）で販売され、普及していった。そのときの大学を中心に生まれたオープン＝公開の精神「ソフトウェアは人類共通の財産である」という考えが広まり、UNIXやその上で動くソフトウェアもネットワーク経由で改良が加えられながら広まっていったのである。
- 36 1984年にマサチューセッツ工科大学（MIT）のRichard Matthew Stallmanがソース・コードを公開する考え方を進めたフリーソフトウェア運動・GNUプロジェクトを開始する。このプロジェクトの目標はUNIX互換OSを開発して、そのソース・コードを自由に利用できるような公開することで、Stallmanが考えたGNU General Public License（GPL）ではGPLで作ったものをベースに新しいものを作るためには、新たに作った部分についても同じGPL条件で次の利用者にも交換することを約束させるというライセンス条件で、GNUを利用してソフトウェアを作りながら公開をしないならばライセンス条件を認めずには作ったとして著作権法違反として訴えられる仕組みになっている、というライセンス方式である。そしてコピーライト（copy right：著作権）という知的所有権を認める法律の逆手に取りその思想のエッセンスであるコピーレフトの概念を提唱した（Stallman [2002] 参照）。GNUプロジェクトによってエディタであるGNU EmacsやコンパイラGCCとデバッガGDBなど多くのソフトウェアが開発・公開された。しかしながらUNIX自体がまだ高価で、専門家が使うOSであったため、一般にはあまり関心をもたれなかった。
- 37 Linus, T., Diamond D., [2001] 参照。Linusは現在アメリカでOSDL（Open Source Development Labs）というNPOに所属し、Linuxカーネルの新バージョンの開発を続け、2005年現在も、公式のLinuxカーネルの最終的な調整役（もしくは「優しい独裁者（終身）」）を務めている。
- 38 島根県松江市に本社を置く株式会社ネットワーク応用通信研究所（NaCl: Network Applied Communication Laboratory）のまつもとゆきひろ氏（通称Matz）により開発されたオブジェクト指向スクリプト言語。従来Perlなどのスクリプト言語が用いられてきた領域でのオブジェクト指向プログラミングを実現する。Rubyは当初1993年2月24日に生まれ、1995年12月にfj上で発表された。名称のRubyは、プログラミング言語Perlが6月の誕生石であるPearl（真珠）とほぼ同じ発音をすることから、7月の誕生石のルビーを取って名付けられた（まつもと・石塚 [1999] 参照）。
- 39 これらのプログラム言語の他、リレーショナルデータベースのMySQL、PostgreSQL、WebブラウザのMozilla FireFox、Thunderbird、MicrosoftのOfficeとほぼ同等の機能を持ったOpenOffice.orgなどがある。
- 40 國領・佐々木・北山 [2000]、秋元・岡田 [2004] など参照。例えば企業向けにはRed Hat Linuxという商用ディストリビューションが有名で、個人ユーザに人気の高いTurboLinux Desktopには日本語入力ツールATOKなどが付属、ライブドアによるLindowsなどがある。島根大学総合情報処理センターのPC端末は日本語環境が充実しているVine Linuxを採用している。
- 41 情報処理用コンピュータのOSがTSS（Time Sharing：タイム・シェアリング）を基本としているのに対し、TRONは制御機器用のOSなので実時間で待たなして対応する必要（例えば車のエンジン制御ならピストンが上がるまでに点火タイミングのための計算が終わっていないとエンストしてしまう）からリアルタイム（実時間）OSとなっ

- ている。
- 42 TRON は組み込み制御用のコンピュータの OS だけでなく、コンピュータのハードウェアの規格や IC カード、非接触認証などの規格からコンピュータと人間の間のインターフェースデザインまで含めた標準化とオープン化の取り組みになっている。そのため TRON プロジェクトは、MTRON (TRON プロジェクトの目標とする分散コンピューティング環境)、ITRON (組み込みシステム向けのリアルタイム OS)、BTRON (パソコン向けの OS)、eTRON (セキュリティ規格を定めたもの、IC カード、非接触認証などの規格) などの互いに連携する多くのサブプロジェクトによって構成されている。
 - 43 アメリカでは1981年に著作権法が改正され、ソフトウェアの著作権が認められた。その翌年 FBI の「おとり捜査」で日立製作所の社員が IBM の情報をスパイしたとして摘発・逮捕される事件が起こった。日本でもプログラムが著作物として著作権の保護対象になっている (著作権法2条1項10号)。なお著作権法で保護されるプログラムはゲームのプログラムからビジネスアプリケーションまで多様であるが、著作物として保護されるには創作性が必要になってくる。この点についてプログラムは他の音楽や絵画等の伝統的な著作物に比べて実用的な性格をもっているため、保護に値する創作性のレベルも高くなくてはならないと考えられている。現に日本の裁判例ではプログラムが保護を受けるための創作性をかなり高いレベルで要求している。
 - 44 Raymond によって提唱され、1998年の Freeware Summit というイベントで投票によって決められた。
 - 45 Raymond [1997] 参照。
 - 46 1998年に Netscape 社はブラウザ Netscape Communicator のソース・コードを公開し、また IBM や Oracle といった大手の IT 企業が Linux のサポートを発表した。
 - 47 後述するように株式会社ネットワーク応用通信研究所 (NaCl) はその代表例であり、地方 (松江市) においても OSS を活用した情報サービス産業の発展が可能であることを証明している。
 - 48 最近ではこのような Web サービスだけでなく、基幹系 (横河電機、大日本インキ化学工業、アクセンチュア、大日本印刷など)、銀行系企業 (東京三菱銀行、UFJ グループ、甲府信用金庫) などでの OSS 導入が目立ってきている。なお野田ゼミナールのブログサイト <http://nodazemi.nlog.jp/> も NaCl がサポートする OSS である tDiary <http://www.tdiary.org/> を使って運営されている。
 - 49 末松千尋 [2003~2004] 参照。また末松 [2004] においてもオープンソースの価格ゼロによる普及力に着目し、オープンソースの優れた所有権開放戦略がその普及する原動力であるとして「非常に短期間で世界的な普及を可能としたメカニズムである。」(204頁) としている。問題はこの普及の中で地域の情報サービス産業にいかに関与の機会、価値創造が発生するかであろう。
 - 50 <http://www.ipa.go.jp/software/open/forum/> 参照。
 - 51 2006年1月1日付けで独立行政法人・情報処理推進機構 (IPA) に設置。
 - 52 現在、ベンダーの多くがそれぞれ同じような検証や情報収集を行っているが、重複した非効率な投資が行われていると考えられ、これらの情報を共有できるようにすることで国内 IT 産業全体での投資効率の向上が期待できる。収集した情報はデータベース化し Web サイトで閲覧、検索できるようにしている。
 - 53 http://www.ipa.go.jp/software/open/forum/north_asia/index.html 参照。
 - 54 総務省 [2004] 参照。

- 55 経済産業省 [2003] 参照。
- 56 経済産業省は、学校に Linux デスクトップ機約300台を配布し、実際に授業に使用してその実用度と課題を探るプロジェクトを2005年度後半から開始している。大蔵小学校のほか、岐阜県の2校、茨城県つくば市の6校、所沢市の1校など十数校が参加、約1000人の児童、生徒が Linux を使用している。
- 57 これに先立ち、既に IT 特区として「OSCA (Open Source Community in Awaji) プロジェクト」を推進している兵庫県洲本市では2004年1月に市内 LAN 一新にあたり、約280台のクライアント PC の統一環境として、Windows と Linux のデュアルブート対応を図り、オフィス製品としては、オープンソースの「OpenOffice.org」を Windows 側にもインストールして Linux とのドキュメント共存を実現している (<http://www.it-media.co.jp/enterprise/0401/19/eprn10.html> 参照)。
- 58 <http://www.kantei.go.jp/jp/singi/it2/kettei/050224/050224pac.html> 参照。
- 59 総務省 [2004]、13頁～21頁参照。
- 60 経済産業省 [2003]、45頁～60頁参照。なお IT 企業が OSS を導入する際の課題として、収益性や新規参入容易性 (それによるメリット、デメリット)、組み込み機器における優位性などが記述されているが (49頁～51頁参照)、OSS 自体の産業化の可能性、その手法については触れられていない。
- 61 日本での OSS の展開の可能性に触れている社団法人情報サービス産業協会編 [2005] においても IBM や Sun などの米国企業がソフトウェアやソフトウェア特許を無償公開していることについて「IBM が大量のソフトウェア特許を無償公開したとは言え、それは IBM の特許全体からすると微々たるものであり、OSS が未知の IBM ソフトウェア特許 (いわゆるサブマリン特許) を侵害して権利問題が発生することは今後も十分ありえる」(48頁参照) と警鐘を発している。
- 62 「科学的知識」が「自由に活用されるべき」と無前提に主張するのは観念論であり、科学が資本主義的経済体制 (企業) の下で、いかなる開発・生産システムといかなる組織形態によって産出され、他企業によって利用されているか=すなわち再生産されているのかを具体的に解き明かすことこそが重要である。
- 63 この課題は経済産業省 [2003]、46頁でも指摘されているが、分析は進んでいない。経済産業省 [2003] ではその原因を続けて「残念ながら、オープンソース・ソフトウェアに取り組む開発コミュニティのほとんどは海外にあり、我が国ではまだこうした開発スタイルは定着していない」ことをあげて、日本における中小企業を中心とした OSS による開発スタイルの分析を放棄している。
- 64 広岡 [2005.7]、140頁～143頁参照。
- 65 2005年11月22日実施。長崎県情報政策課からは情報政策担当・島村秀世参事監 (CIO) 電子県庁推進班・大庭茂雄課長補佐が対応。また地元情報サービス企業の株式会社ドゥアイネットの SE の取材で、分割発注方式によって地場の情報企業もシステム開発に参加することが可能になり、その結果地域の情報サービス企業の参入、育成にも大きな成果をあげていることと、ソースの公開によって県外企業からも問い合わせが始めているが、これが実際の仕事、市場の獲得に成果をあげるのはまだこれからだとの説明があった。
- 66 <http://japan.cnet.com/news/ent/story/0,2000047623,20079691,00.htm> 参照。
- 67 ソースコードの著作権は長崎県が保有する。ソースコードは、「改変時に、管理者にその内容を報告する」、「ソースコードを公開することなく汎用パッケージ製品を開発・販売する

- ことはできない」といった条件を持つ「GPL ライセンス」の考え方に基づいて公開。ライセンス管理は、長崎県下の IT 企業が組織する「オープンソースベンダーフォーラム長崎」が担当する（広岡 [2005.10] 参照）。
- 68 「江別ブランド事典」<http://www.ebetsu.city-brand.jp/top.php> 参照。また、この取り組みのポイントとして「第1のポイントとして、江別に直接来られない人や商品・素材を直接手にできない人のために市民の声を集め、これを情報として公開することにし、第2のポイントとしては「対話型自然言語ナビゲーションシステム」という、ごく普通の会話を理解し対応できる検索システムを導入、さらにこのシステムをオープンソースを利用して開発し、完成したシステムプログラムをオープンソースにすることにしました。」と紹介されている。<http://www13.ocn.ne.jp/~eknet/project1.html> 参照。
- 69 「NPO 法人・新潟オープンソース協会」、<http://www.niigata-oss.org/> 参照。
- 70 「オープンソースは下請けから脱出する足がかりになる」
http://itpro.nikkeibp.co.jp/article/NEWS/20051121/224911/?ST=itpro_print 参照。
- 71 長崎県でのヒアリングにおいても県の職員の仕様策定能力アップとそれを可能にする組織体制、最終的にはスーパー SE である CIO の能力に負うところが大きいことも判明した。
- 72 島根県内 IT 関連企業（34社）へのアンケート調査と集計、島根県内 IT 関連企業（5社）へのヒアリング調査を行った。野田 [2005] b、84頁～123頁参照。
- 73 情報サービス産業が、同業者間での委託・受託の連鎖が複雑に連なりあっていることから、委託・受託のそれぞれの割合によって【自立型】（同業者からの受託 20%未満、同業者への委託 20%以下）13社、【元請型】（同業者からの受託 20%未満、同業者への委託 20%超）3社、【下請型】（同業者からの受託 20%以上、同業者への委託 20%以下）17社、【中間型】（同業者からの受託 20%以上、同業者への委託 20%超）3社、として4つのタイプに分類した場合、【元請型】がいずれも情報サービス部門のみの売上が1億円～10億円であり、また【下請型】の売上高が全体的に少ないのを除いては、情報サービス企業のタイプと売上高との大きな相関は見られなかった。特に【自立型】では売上高がもっとも分散しており、小さな企業でも自立志向の強い企業が存在するこの分野の特性が島根県の情報サービス産業でも示されている。調査ではこの情報サービス企業の規模（売上高）と、委託・受託の類型（4つのタイプ）も基準にしなが、島根県の情報サービス企業の現状、動向、課題などを分析している。
- 74 同時に人材教育の課題については「業務多忙のため教育時間がない」（49%）や「戦略的人材教育なし」（43%）と回答している企業が多い。ヒアリング調査においても仕事内容に対応して言語を習得するなど、その場その場の対応でしどろいであることがわかり、系統的・戦略的な人材教育ができていないことがわかる。さらに下請型の企業においては「現場に教育体制がない」（50%）と回答している企業が多く、成長分野であり技術革新が急速に進む分野であるので企業内外における研修を含めた人材教育は不断に必要とされるが、同時に日常業務が多忙であるためにそのための時間と体制を割くことができない現状が浮き彫りにされている。
- 75 松江市の典型的な大手【元請型】情報サービス企業のヒアリング調査では、地元で受注したシステムのプログラム開発が東京の「Java 工場」で集中的に行われていることが明らかになった。
- 76 全国の医師、医療関係機関が誰でも無料で使え、改良できる公開ソフトウェア（オー

- ブンソース)方式でプログラムを配布するプロジェクト。ネットワークを用い、情報の配信、蓄積、点数改正などに対応できるネットワーク技術とソフトウェア公開している。「ORCAプロジェクト」、<http://www.orca.med.or.jp/index.rhtml> 参照。
- 77 日医標準レセプトソフトはオープンソースを実現するため、基本ソフト (OS) は Linux、データベースは PostgreSQL、COBOL コンパイラは OpenCOBOL、その他もすべてオープンソースとなっている。
- 78 株式会社ネットワーク応用通信研究所 (NaCl) の ORCA サービス&サポート、<http://www.netlab.jp/ORCA/> 参照。

【参考文献】

- Brooks, F. P. Jr. [1995], *The Mythical man-month: Essays in software engineering*, Anniversary edition (first edition [1975]), Addison-Wesley. (邦訳『人月の神話 ~狼人間を撃つ銀の弾はない』、ピアソン・エディケーション。)
- Cusumano, M. [2003], 「“synchronize and stabilize” ソフトウェア開発の成功例に学ぶ」 CSK XPRESS, 2003, 10~11.
- Cusumano, M., MacCormack, A., Kemere, C. F., Crandall B. [2003], “Software Development Worldwide: The State of the Practice”, *IEEE Software Nov/Dec 2003*, p. 28~34 http://ebusiness.mit.edu/research/papers/178_Cusumano_Intl
- Cusumano, M. [2004], “The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad”, Free Pr.
- Demarco, T., Lister, T. [1999], *Peopleware: Productive Projects and Teams*, Second edition (first edition [1987], Dorest House. (邦訳『ピープルウェア やる気こそプロジェクト成功の鍵 第2版』、日経 BP 社。)
- Linus, T., Diamond, D. [2001], *The Story of An Accidental Revolutionary*, (邦訳『それがぼくには楽しかったから』、小学館プロダクション。)
- Pressman, S. R. [2005], *Software Engineering, A Practitioner's Approach* 6th Edition, McGraw Hill. (邦訳『実践ソフトウェアエンジニアリング ~ソフトウェアプロフェッショナルのための基本知識』、日科技連。)
- Raymond, E. [1997], “The Cathedral and the Bazaar” (邦訳『伽藍とバザールーオープンソース・ソフト Linux マニフェスト』、光芒社。)
- Stallman, R. [2002], *Free Software, Free Society: Selected Essays of Richard M. Stallman*, Free Software Foundation. (邦訳「フリーソフトウェアと自由な社会」、アスキー。)
- 秋元芳伸・岡田泰子 [2004], 『オープンソースを理解する』、株式会社ディー・アート。
- 池田信夫・林絃一郎 [2002], 「通信政策・ネットワークにおける所有権とコモンズ」奥野正寛・竹村彰通・新宅純二郎編『電子社会と市場経済：情報化と経済システムの変容』第7章、新世社、所収。
- 大野尙郎 [2003], 「ソフトウェア工学発展史 ~ウォーター・フォール (落水) モデルの錯覚・誤用・悪用の30年」 <http://www.ivis.co.jp/prof/07.html> より。
- 大場充・堀田勝美・松瀬健司 [2003], 『ソフトウェアプロセスと組織学習 ~CMM を毒にするか? 薬にするか?』、ソフト・リサーチ・センター。
- 國領二郎・佐々木裕一・北山聡 [2000], 『Linux はいかにしてビジネスになったか』、NTT

- 出版。
- 経済産業省 [2003]、「オープンソース・ソフトウェアの利用状況調査 導入ガイドライン」
<http://www.meti.go.jp/kohosys/press/0004397/1/030815opensoft.pdf> より。
- 経済産業省 [2004]、『平成15年 特定サービス産業実態調査』、経済産業省。
- 経済産業省 [2005]、『平成16年 特定サービス産業実態調査』、経済産業省。
- 小山裕司・竹田陽子 [2001]、「ソフトウェアの開発技法と構造」藤本隆宏・武石彰・青島矢一編『ビジネス・アーキテクチャー』、有斐閣、所収。
- 佐野嘉秀 [2001]、「情報サービス業における外注化と社員の役割」電機連合総合センター編、佐藤博樹監修『IT時代の雇用システム』日本評論社、所収。
- 嶋田隆・祝谷和宏 [2004]、「ソフトウェアエンジニアリングセンター構想について」、『情報処理』vol.45 No.4、2004年4月号。
- 社団法人情報サービス産業協会編 [2005]、『情報サービス産業白書2005』、コンピュータ・エージ社。
- 末松千尋 [2002]、『京様式経営 モジュール化戦略—「ネットワーク外部性」活用の革新モデル』、日本経済新聞社。
- 末松千尋 [2003~2004]、「オープンソース戦略を探る」、CSNET Japan
<http://japan.cnet.com/column/suematsu/> より。
- 末松千尋 [2004]、『オープンソースと次世代IT戦略』、日本経済新聞社。
- ソフトウェア開発・調達プロセス協議会 [2001]、「我が国ソフトウェア開発・調達プロセスの評価指標（日本版CMM）の策定に向けて（案）」
<http://www.meti.go.jp/feedback/downloadfiles/i10626bj.pdf> より。
- 総務省 [2003]、『平成15年度版 情報通信白書』、総務省。
- 総務省 [2004]、『平成16年度版 情報通信白書』、総務省。
- 総務省 [2004]、「電子政府・電子自治体におけるOS選定のあり方について」
http://www.soumu.go.jp/s-news/2004/pdf/040428_1_a.pdf より。
- 総務省 [2005]、『平成17年度版 情報通信白書』、総務省。
- 谷口功 [2005]、『プロジェクトマネージャーの仕事と技術』、メディア・テック出版。
- 西村清彦・峰滝和典・白井誠人・黒川太 [2002]、「産業経済の変容：『ニュー・エコノミー』は日本に存在するか」奥野正寛・竹村彰通・新宅純二郎編『電子社会と市場経済：情報化と経済システムの変容』第1章、新世社、所収。
- 西村清彦・峰滝和典 [2004]、「日本の情報サービス産業の生産性」西村清彦・峰滝和典『情報技術革新と日本経済』、有斐閣、所収。
- 野田哲夫 [2005] a、「ユビキタス・ネットワークと「情報資本主義」段階の深化—IT投資としてのユビキタス・ネットワークと日本的「ネットワーク」型企業間関係の完成—」、『島根大学法文学部紀要社会システム学科編 経済科学論集』第30号、所収。
- 野田哲夫 [2005] b、「島根県情報サービス企業調査から見られる情報サービス企業の現状と課題」、島根県商工労働部編『島根県の産業と労働』、所収。
- 広岡延隆 [2005.7]、「IT戦略化の決め手 長崎県 ~職員が自ら詳細仕様を作成しIT調達コストを大幅ダウン」、『日経コンピュータ』2005年7月25日号、日経BP社、所収。
- 広岡延隆 [2005.10]、「長崎県、電子県庁システムのソースコードを公開」、『日経コンピュータ』2005年10月21日号、日経BP社、所収。
- 馬場史郎 [2002]、「人月での見積もりがエンジニアをダメにする」@IT自分戦略研究所
<http://jibun.atmarkit.co.jp/fengineer/special/ningetu/ningetu01.html> より。

前川徹 [2004]、『ソフトウェア最前線』、アスペクト。
まつもとゆきひろ・石塚圭樹 [1999]、『オブジェクト指向スクリプト言語 Ruby』、アスキー。