

**ECONOMIC RESEARCH CENTER
DISCUSSION PAPER**

No.172

**Governance at the development workplace in the
Japanese software companies**

by
Tokumaru Norio

March 2009

**ECONOMIC RESEARCH CENTER
GRADUATE SCHOOL OF ECONOMICS
NAGOYA UNIVERSITY**

日本のソフトウェア企業における経営管理： 技術選択および雇用・取引慣行との適合性*

徳丸 宜穂†

概要

本稿は、日本のソフトウェア産業の競争力を規定する経営管理上の「特質」「仕組み」を発見し、それらの存立のロジックと含意を検討することを目的とする。進展する国際分業の中で日本のソフトウェア産業とその雇用がどのように展開してゆくのかを考える上で、日本のソフトウェア企業が有する経営管理上の特質を体系的に理解することが重要である。本稿が依拠する予備的な聞き取り調査によれば、1) 利益の大きさは、概ね見積もり段階で決まってしまうこと、2) それは、日本企業の技術選択と関係があると考えられること、3) その技術選択は、雇用・取引慣行と整合的であること、4) その技術選択はプロジェクト組織の形態を規定しているが、その組織形態は人材育成慣行と適合的であること、および 5) 受注の方針は収益を大きく規定するが、それは人材育成および長期継続的取引という諸慣行の影響を強く受けるということが明らかになった。総じて、日本のソフトウェア企業の経営管理は、雇用・取引慣行と適合的に展開していると考えられる。

1 問題意識

本稿は、日本のソフトウェア産業の競争力を規定する経営管理上の「特質」「仕組み」を発見し、それらの存立のロジックと含意を検討することを目的とする。具体的には、筆者が今年度に行った予備的な聞き取り調査から知り得た事実を分析的に叙述し、それをもとに重要な特質についての検討を行うこととする。したがって本稿は、来るべき本格的な調査研究、なかんずく国際比較の準備作業として位置づけられる。

ソフトウェア産業が日本企業の競争力にとって重要であることは言うまでもない。それは、機器制御や情報処理でソフトウェアへの依存度をますます高めている製造業にとっても該当することである。事実、製品開発の大部分は、ますますソフトウェア開発になってきている。例えば、携帯電話などの情報通信機器類の開発で最も工数を要するのはソフトウェアである。したがって、今後の日本経済の国際競争力を考える上で、ソフトウェアを対象とすることは適切であると考えられる。

その上、ソフトウェア産業は、中国やインドでもっとも発展が著しい産業である。これら諸国を含む国際分業がどのように展開し、その中で日本のソフトウェア産業がどのように展開していくのかは、上記の問題を考える上で、中核的に重要な問題だと言えるだろう。具体的には、オフショアリングの展開によって雇用・産業が空洞化するという米国の経験と同様の経路を辿るのか否か (cf. 季・高橋 [15])、また、高品質で革新的な製品・サービスを継続的に生み出してゆけるのか否かといったことが問われている。

しかし、こうした問題に対しては、日本企業がどのような経営管理ルーチンを持っているのかを考えなくては、答えを出すことが不可能であろう。なぜならば、各々の経営管理に応じて、企業が

*いちいちお名前を挙げられないが、聞き取り調査にご協力いただき、貴重な知見を提供してくださった方々に、心より御礼申し上げたい。もちろん、すべての誤りは筆者の責任である。

†名古屋商科大学経済学部。E-mail: norio-t@mbox.kyoto-inet.or.jp

とりうる経営行動のオプションは大きく規定されるだろうし、したがって国際分業の中で占める位置は大きく規定されるものと考えられるからである。それゆえ具体的にはまず、各国企業の経営管理ルーチンを体系的に比較分析する中から、日本企業の経営管理の特徴を把握する必要がある。また、近年の国際分業の進展や、業績管理の強化と言った変化に対して、どのように経営管理が変化しているのかを分析する必要もある。

したがって長期的な課題としては、国際分業の中で日本のソフトウェア産業の占める位置を問うこととなるが、当面の課題としては、経営管理の国際比較を行いたいと考えている。しかしそのためには、比較分析の明確な視角を持っている必要がある。本稿はそれを準備するものとして位置づけられる。

本稿の構成は以下の通りである。第2節では先行研究に対する本稿の位置づけを明確にする。第3節で経営管理を分析的に叙述するための枠組を提示し、それに基づいて第4節では聞き取り調査の結果を分析的に叙述する。第4節は経営管理の制度・仕組みを単に叙述するのではなく、当事者の意図・認識を含む叙述を行う。そうすることによって、第5節での議論が可能になる。第5節では、競争力を左右すると考えられる重要な経営管理実践は開発技術の選択と深く関係しており、またそれらは日本企業の雇用・取引慣行と適合性を有していることを主張する。

2 先行研究

企業は各国の社会経済制度の中に埋め込まれており、それゆえ企業行動はそうした制度の影響を受けざるを得ないという考え方は、比較制度論 (comparative institutional analysis) 的な研究が共有する認識である。こうした仮説的な認識は、個別産業の文脈で実証される必要があるし、事実、雇用慣行や契約・取引慣行に注目した比較制度論的な実証研究が、情報通信産業やバイオテクノロジー産業などについてなされてきた (e.g., Grimshaw and Miozzo[22], Miozzo and Grimshaw[26][27], Casper[20], Casper and Vitols[19], Lam[25])。本研究も、こうした一連の研究のうちに位置づけられる。ただし、こうした研究のうちには、企業内の経営管理ルーチンを十分に解明することなく、企業外の制度、例えば雇用慣行が技能蓄積に特定の影響を与えるはずだというような理論的推論に大いに依拠して結論を導いてしまっているものが多い点で、実証水準が不十分だと思われる。このことは、経済制度が企業行動に影響を及ぼすロジックを明らかにする上で、問題だと考える。

経営管理ルーチンを明らかにする先行研究としては、管理会計論の文献を参照する必要があるが、人事管理の観点から経営管理、とりわけ部門業績管理を論じているのが今野 [3]、石田 [4]、佐藤 [7] などである。とりわけ今野 [3]、佐藤 [7] は、企業統治における近年の「株主価値重視」という動向が、部門業績管理ひいては人事管理にどのように影響を及ぼすかという問題を検討している。経営管理を体系的に理解したいという本研究の目的から見れば、「業績管理 人事管理」という連関を分析するこれら研究は、非常に示唆的であるし、こうした方法を国際比較に導入することにも大きな意味があると思われる。しかし「人事管理 業績管理」という逆の連関を分析する必要があると考える。なぜなら、人事管理は雇用慣行に規定されて各国の固有性が強く、また変容しにくい領域だとも考えられるが、それに連動して業績管理にも各国の固有性が現れると予想されるからである。

ソフトウェア開発はプロジェクト組織で行われるのが普通である。プロジェクト組織は階層的な企業組織とは異なる原理に立つ組織形態であるとの観点から、盛んに研究がなされている (e.g., Whitley[29], Ibert[24], Hobday[23])。例えば Barley and Kunda[18] は、企業組織はもはや、プロジェクトが展開する文脈に過ぎないとまで述べ、プロジェクトの実態に迫ることの重要性を強調す

る。また、戸塚・中村・梅澤 [9] は、長期的な人材育成・活用を旨とする日本型雇用慣行と、短期性・流動性を旨とする組織形態であるプロジェクト制とが矛盾する関係にあるという解釈を提出した。彼らが挙げた論点の一つは、プロジェクト制によって人材育成が阻害されるのではないかと言うことである。しかし Whitley [29] は、企業組織とプロジェクト組織の関係は単純に背反するものではなく、いくつかの類型を考えるべきことを提案している。現に、戸塚・中村・梅澤 [9] の解釈にもかかわらず、日本企業のとりわけ開発部門でプロジェクト制が定着していることは明白である。したがって、プロジェクトの運用・組織形態とその多様な実態に迫る研究が必要だと思われるし、日本企業のプロジェクト制を上記先行研究の中に位置づける必要がある。

ソフトウェア産業に関する先行研究は、日本のソフトウェア産業については戸塚・中村・梅澤 [9]、今野・佐藤 [1]、Baba, Takai and Mizuta [17]、クスマノ [5]、古谷 [13] などがあり、国際分業の文脈に日本のソフトウェア産業を位置づけようとするものとして李・高橋 [15] などがある。また近年の後発国でのソフトウェア産業については、Arora and Gambardella [16] などがある。戸塚・中村・梅澤 [9] および今野・佐藤 [1]、古谷 [13] はいずれも部分的ではあれ、経営管理のルールを解明している点で優れているものの、国際比較の観点が無い。また Baba, Takai and Mizuta [17] およびクスマノ [5] は、日本のソフトウェア産業における需要面および品質・生産性に関する特質について、国際比較の問題意識から明らかにしているものの、経営管理に関する実証はなされていない。経営管理に関する説明が手薄であるという指摘は、李・高橋 [15] および Arora and Gambardella [16] についても該当する。

3 経営管理の分析枠組

本節では、経営管理を体系的に理解するための分析枠組を述べておきたい。第4節では、以下のような分析枠組を用いて、調査結果の叙述を進めていく。

3.1 業績管理

まず<業績管理>は、部門目標・プロジェクト目標を設定することによりソフトウェア開発業務を管理する。近年では、財務指標への過度の注目に対して批判がなされ、非財務指標をも一定程度重視したBSC(バランススコアカード)が多く採用されている。業績管理は、BSCあるいは「プロジェクト計画書」のような書類の作られ方と、その運用方法を押さえれば理解できると考えられる。具体的には例えば、KPI(Key Performance Indicator)の項目や水準はどのように決まるのか、またそれはどの段階の組織・個人までブレイクダウンされるのか、プロジェクトもKPIのブレイクダウン対象なのかどうかなどを実証する必要がある。

3.2 開発管理

業績管理で立てられた目標を達成すべく、開発現場に対して行われる管理が<開発管理>である。開発管理を抽象化し技法化したのがいわゆるプロジェクト・マネジメントの諸手法に他ならない。ここでは例えば、進捗管理の方法や、品質管理部門の関与、権限と責任の分担形式などを実証することが課題となる。

3.3 人事管理

開発に要する工数が決まれば、いかなる能力を持った技術者がどれくらい必要かが判明する。そこで、そうした技術者を適切に提供し、適切な労働を引き出すのが、ヒトの管理、すなわち<人事管理>である。人事管理は、賃金や昇進・昇格などの手段で、技術者にインセンティブを与えるのみならず、個々の技術者の能力形成を促す管理である。また、採用管理により正社員の一定数を維持することも含まれる。賃金制度、人事査定制度、キャリアパスと人材育成方法を押さえることが要諦ではないかと思われる。当然、労働市場の構造特性が違えば、人事管理も異なるだろう。ゆえに、各国別に、情報技術者労働市場の全体像をも解明する必要がある。

3.4 受注管理 / 製品企画

そもそも、開発管理が対象とする所の、開発作業内容と総量が決まるのは、<受注管理>によってである。あるいは自社製品の場合にはそれは<製品企画>となるだろう。受注管理は、仕事の分量、金額と内容を規定する、主に営業部門によって担われる管理である。したがって、営業部門の業績管理と不可分の関係にある。例えば、業績管理が売り上げのみでなされるのか、それともBSCなどで売上高以外の指標でもなされているのかによって、受注方針は大きく異なるだろう。営業部門の業績管理は第1のポイントであるから、実証したい。第2のポイントは、「契約」の形式と内容であろう。そもそも明示的な契約の形をとっているのか、契約は「請負型」「委任型」あるいは「技術者派遣」の類型がどのようなロジックで選択されるのか、また知的財産の帰属はどのように規定されているのかなど、契約の実態を実証したい。

3.5 外注管理

受注量が決まっても、自社のリソースのみで作業を出来ることはまずない。外部のリソースへの依存が不可避である。それは大きくいって「業務委託」もしくは「派遣」の形をとるだろう。こうした外部リソースの管理を<外注管理>という。外注管理によって、外注の分量、金額と内容、形態が決まる。ここでも、「契約」の形式と内容がポイントである。受注企業との「契約」の形式と内容、分担作業の切り分け、リスク分担と知財の帰属などについて、実証したい。例えば、「デザイン・イン」的な関係が請負会社との間に見られるのかどうかなどは、重要な論点である。

3.6 技術管理

最後に与件とされがちな箇所だが、そもそも必要な技術力がないと、開発が難しい案件もあるだろう。したがって、自社の強みとなる技術力を蓄積し有効に利用する必要がある。これを<技術管理>という¹。技術管理は、新技術を開発することはむろんだが、属人的知識を組織的知識にすることが課題となる。一つの焦点は、品質管理部や技術開発部などの機能部門の活動である。また、そうして蓄積された技術能力をどのように収益に結びつけるかも、技術管理の一課題である。その代表例は知的財産権の管理である。

¹これは広義には「投資管理」と言いうる領域に属すると見ることも可能だろうが、本稿では区別しない。

4 調査内容

本節では、上記の分析枠組に依拠して、予備調査から判明した限りの経営管理実践を明らかにする。ただし「外注管理」「技術管理」については十分な調査が出来ていないため割愛する。

4.1 調査対象の概要

調査対象企業の 2007 年 3 月段階での概要は、次の通りである。ただし、C 社および D 社はソフトウェア部門を含む複数部門を持つ企業であり、資本金および従業員数は全社の数値である。聞

表 1: 調査企業一覧

企業名	系列	製品	主要事業	資本金 (百万円)	従業員数
A	メーカー系	情報系 / 組み込み系 SW	受託開発	100	874
B	ユーザー系	組み込み系 SW	受託開発	300	350
C	独立系	組み込み系 SW/HW	受託開発	2200	278
D	独立系	組み込み系 SW/HW	受託開発	2400	377
E	独立系	FA 向けパッケージ SW	自社製品	40	30
F	独立系	情報系 / 組み込み系 SW	受託開発	80	72
G	独立系	情報系 / 組み込み系 SW	技術者派遣・受託開発	30	200
H	独立系	情報系 / 組み込み系 SW	受託開発	700	283
I	独立系	情報系 / 組み込み系 SW	受託開発	200	325
J	独立系	情報系 / 組み込み系 SW	受託開発	15	59
K	独立系	組み込み系 SW/HW	受託開発	265	337
L	メーカー	情報通信機器	自社製品	5300	855
M	メーカー	情報通信機器	自社製品	3500	536

(出所) 各社ウェブページ

き取り調査は 2007 年 5 月から 2008 年 6 月にかけて、各社とも 1 回ずつ、90 分ないし 120 分で行った。いずれも、開発担当部署のマネジャーを対象としている。方法はいわゆる semi-structured interview であり、そもそもの問題・仮説を発見し深めるための exploratory case study として行った (Yin[30])。質問項目や調査の精粗には企業ごとに差があり、したがって、比較検討に耐えうる聞き取りが出来ているわけではないことを予め断っておきたい。

ソフトウェア企業は、資本系列によって「メーカー系」「ユーザー系」および「独立系」に大別される。メーカー系はコンピュータメーカーの子会社、ユーザー系は情報システムユーザーの子会社であるが、両者のいずれにも属さない企業が独立系と呼ばれている。また、調査対象企業が開発するソフトウェアは「情報系」「組み込み系」および「パッケージ」に大別される。情報系は企業の情報システムに、組み込み系は機器の制御・情報処理に用いられる。後者の例は、携帯電話内蔵ソフト、車載ソフトである。後者の場合、B 社のように組み込みソフトウェアだけを開発するケースもあれば、C 社および D 社のように、ソフトウェアとハードウェアを同時に開発し納入するケースもある。またパッケージは、汎用ソフトウェアである。

組み込み系の開発は情報系の開発に比して、極めて高い品質と機密性が要求される。例えば情報系の場合だと、しばしば「重大障害をせいぜい 1 件にしてくれ」という緩い品質基準でしかない

が、自動車や飛行機、原発に用いられる組み込み系の場合、「不良品を出してはならない」という厳しい品質基準が設定される。そのため、コストもかかるしノウハウも重要な世界である（A社聞き取りより）。また、情報系にせよ組み込み系にせよ機密性が重視されるが、組み込み系の場合、製品機能の中核を担うだけに、機密保持には一層の注意が払われている。

4.2 開発工程および組織

開発工程については、各社で名称に若干の相違があるものの、いわゆる「ウォーターフォール方式」が取られていることは共通している。典型例として、B社の開発工程を挙げると次のように進行する。

- (1) 顧客の依頼
- (2) 見積もり
- (3) 受注
- (4) プロジェクト計画
- (5) 要件分析
- (6) 基本設計
- (7) 詳細設計
- (8) コーディング
- (9) デバッグ
- (10) テスト
- (11) 納品
- (12) 検収
- (13) 請求・入金

まず(4)プロジェクト計画では、QCD等にかかわるプロジェクトの目標が立てられる。(5)から(10)までが、プロジェクトで実際になされる開発作業である。(5)要件分析で、顧客の要求仕様が分析され、(6)基本設計において、その機能を実現するためのシステム構成が決められる。(7)詳細設計ではシステムを構成する各モジュール内部が設計され、それを具体的なプログラミング言語で翻訳していくのが(8)コーディングということとなる。このように、詳細度を順次上げていく設計方法がウォーターフォール方式と呼ばれる。

ソフトウェア開発プロジェクトは、L社以外では「開発*部」などと称される機能部門の中に組織される「機能重視型」(延岡[11])プロジェクト組織である。しかし、情報通信機器を開発するL社では、機器開発の場合に通例であるように、「システム開発部」「機構設計部」「ソフトウェア開発部」など、各機能別部門からのメンバーで構成される、部門横断的な開発プロジェクトである。

通常「プロジェクト」というと、L社のようなクロス・ファンクショナル・チームのことが念頭に置かれ、開発の性質に応じて適切なメンバーが流動的に集められるという組織形態だとされる。いわば、必要な多様な人材を、必要な人数・期間だけ、機能部門とは独立に柔軟に組織するのがプロジェクトの典型的理解である(西村[10])。しかし、今回調査したソフトウェア会社に限って言えば、確かにプロジェクトと称されてはいるものの、機能部門の中で組織されることから考えても、上記の典型的理解からは乖離している。したがって、戸塚・中村・梅澤[9]は、プロジェクト制がもたらすという人材の流動性を過度に強調しているのではないかと思われる。このことは、後出「人事管理」の項目でやや詳しく論じられる。

4.3 受注管理および製品企画

どのような内容の開発案件をいかなる条件で受注できるかは、ソフトウェア企業の収益性を大きく左右する一大要因である。ここでは、誰がどのような基準で受注を決定するのか、また、仕事内容と対価の決定方法＝見積もり方法について見てゆく。その上で、受注方法には大別して「請負」と「技術者派遣」とがあるが、その選択について述べる。最後に、自社製品開発の場合の製品企画について、簡単に見ることとする。

4.3.1 受注の流れと営業方法

まず、営業を担当する営業部門が存在する場合と、開発部門のリーダーやマネジャーが営業を担当する場合とがある。その相違は、新規顧客を開拓する必要性によるものと考えられる。

全企業の営業体制に関して聞き取りが出来ているわけではないが、開発部門のリーダーやマネジャーが営業を行うとしていたのは、B社、F社およびJ社であった。

例えばB社の場合、開発部門の中に「グループ」が複数あり、グループの長であるグループマネジャーが営業をして、自分のグループの仕事を取ってくる。自グループの仕事が少なくなってくるのを予見して、仕事を取る。例えば客先で「アプリケーションの仕事はないか？」等の形で営業をする。ただし、この方法で獲得される開発案件は、関連会社からの仕事であり、関連会社以外の案件も同様な営業方法で獲得しているのかどうかは分からない。

またF社でも、分野ごとの各開発部門にグループが複数あり、その長であるチーフリーダー(=SE:システムエンジニアでもある)が営業を行う。顧客との取引はリピートとなることが多く、顧客から慣れたチーフリーダーが指名されることが多い。逆にそうしたチーフリーダーには、当該顧客の新たな仕事を開拓することが期待されている。この事例はいわば、技術者としての能力を活かした営業方法であると言える。しかし、彼らに求められている営業は、既存大手顧客²のいわば「深耕」であって、必ずしも新規顧客の開拓ではない。加えて、チーフリーダーが統括する各グループの中堅技術者も、同様の、既存顧客に対する営業を行う。

K社では、各開発部門のマネジャーが既存顧客の営業を行い、自社・他社製品の販売を契機とした新規顧客・案件の開拓には別途部門が設けられている。組み込みソフトの場合、顧客はたいていリピートになるから、営業担当者を積極的に新規採用することが長期的には必要となるかも知れないが、直近では採用の予定はないという。

したがって、上述各社とも、営業活動に期待されている役割は、言うなれば「営業プロパー」の役割よりも小さいと言うことに留意する必要がある。

開発案件を受注するか否かを決定する基準は様々だろうが、受注を決定する全社的な会議体が開かれる企業が存在する。

D社では、営業部門が商談を行い、顧客からの見積もり依頼を受けるのだが、一定金額以上の開発案件については「受注前審査」と呼ばれる会議が開かれる。これは、営業・技術・資材部長以上の会議である。後述のように、この会議では戦略的な見地からある案件を受注するか否かを決定するのである。

G社の場合、技術者派遣の受注に関する会議は、毎週、営業部門と技術部門の間で行われている。顧客のニーズと技術者をマッチングすることを主旨とする会議だが、事実上、営業部門と技術部門の利害調整の場となっている。すなわち、営業部門は利益優先となりがちだが、技術部門は技術者の技術レベルや本人の希望を優先しがちとなるので、両者の意向を摺り合わせるのである。技術部門の観点からすれば、ある技術者をプログラマからSEへと育成しなくてはならず、したがって営業部が取ってくる派

²F社は、極力、大手顧客と直接取引をしたいという意向を持っている。それは次のような理由からである。第1に、大手顧客だと、開発を受託した部署以外の他の部署とのつながりも出来、仕事の幅が広がるという利点がある。第2に、安定した売り上げにも貢献する。第3に、受注単価も、2次以下の下請にはいるよりも、大手と直接取引をする1次下請の方がいい。ただし、多くのソフトウェア企業がF社と同様の意向を持っているのかどうかは分からない。

遣案件の要求に任せていつまでもプログラマとして派遣するわけにはいかないと言うことである。

受注に際しては、概して、短期的な収益を最大化することが目指されているとは言えない。技術蓄積や顧客との長期的な関係の構築を狙って、戦略的な受注がなされる。また、技術者の「稼働率」を維持するため、場合によっては条件がよくない案件を受注してでも受注量を確保することが行われる（4.3.3 参照）。

D社およびK社の場合、一定額以上の案件については「受注前審査」があるが、ここで、たとえ高額な投資が必要な案件であったり、リスクがかなり高い案件であっても、経営判断で受注することがある。もちろん逆に、あまりにリスクが高すぎるために受注を見送る案件もある。またB社では、「戦略的プロジェクト」という開発案件の категорияがある。例えばある案件を赤字覚悟ですることで、その顧客関係の大きな仕事が入ってくるというような場合や、赤字覚悟である仕事を取っておかないと、そのお客に参入できないなどの場合に、戦略的見地から受注した案件がそれに該当する。

D社およびB社では、個々のプロジェクトが最優先すべき目標が、作成される計画文書の一項目として明示される。あるプロジェクトが短期的収益を重視したものなのか否かは、この項目を見れば分かる。後出「業績管理」の項目（4.4）で詳述する。

短期的な収益最大化を抑制した受注方針を採らざるを得ないことは、日本のソフトウェア企業の経営にとって重要な意味を持つと思われる。したがってこの点については、第5節で改めて検討する。

4.3.2 受注内容・条件の確定（見積もり）

見積もりの手続きは、各社でほとんど共通である。すなわち、まず開発に要する工数（人月単位）を見積もる。その際に、どのランクの技術者が何人月必要かという形をとる。次に、見積もられた工数に、ランクごとの技術者人月単価を掛け合わせて実際の開発費用が決まる。これに間接費およびマージンを上乘せしたものが、顧客に提示される見積価格である。

技術者の人月単価設定については体系的に聞き取れていない。例えばH社では、開発のリスクを勘案して、案件ごとに技術者の単価設定をしている。またJ社での聞き取りによれば、『月刊積算資料』という雑誌があり、概ね当誌に掲載された単価の8割程度の水準でSE、PGの単価が設定されているという。技術者派遣を行うG社は、例えば「もうすぐSEになりそうな（＝能力が高い）プログラマです」といった細かい提示を顧客に対して行うことによって、技術者能力に応じた単価を設定する。しかし概して、ソフトウェア企業にとって、技術者単価水準を操作するよりも、見積もられる工数を操作する方がはるかに容易であるように思われる。こう判断するのは、単価については、技術者ランクごとに「世間相場」が流布する経路が複数存在するためである（日本労働研究機構 [12]）。

ほとんどの企業は、見積もりの精粗によって、プロジェクトの成功・不成功が大きく決定されてしまおうと考えている。その意味で、見積もりは開発プロセスの中で最も重要な工程の一つである。これは大変重要な意味を持つことだと思われるから、第5節で改めて検討する。

見積もり時に顧客が出す要求仕様は、曖昧で不明確なものであることが通例である³。I社での聞き取りによれば、「役所や大企業だと仕様書が出されない」ことすら稀ではないという。こうし

³顧客の要求仕様が曖昧なのは、日本企業に特有の現象だと見られることがしばしばあるが、必ずしもそうとは言えないようである。例えば、欧州IT産業における雇用・労働に関する国際比較である European Trade Union Institute[21]

た状況で、ソフトウェア企業の担当者がどこまで顧客の要求を明確化することが出来るかが重要である。

それでも、開発途中での仕様変更が頻発するのが現実で、そのために大きく予算超過してしまう事態も生じる。仕様変更の可能性に備えて、工数に余裕を含めた見積もりが出来れば望ましいが、現実には難しい。したがって、仕様変更によって予算超過してしまっても、営業部門が交渉力を発揮して顧客から追加予算を取れることが望ましいのだが、営業担当者の交渉能力がないと難しい。

B社では、見積もりを行うべき項目を明確にした書類である「見積もり要件書」を作成することとなっている。いったん見積もりを完成させてしまうと、後に工数が膨張してしまったからと言って再見積りをさせてくれない顧客も多い。したがって、できるだけ最初の見積もりで顧客の要求条件を明確化し、その条件での見積もりだということを明確にさせて合意させる仕組みが「見積もり要件書」である。この事例は、個々の技術者の見積もり能力を向上させるのではなく、技術者全体の見積もり能力を組織的に底上げする対応策であると言える。

他方、H社での聞き取りによれば、仕様を明確に詰めるためには、ヒアリング力と業界知識が重要だし、万が一予算超過してしまった場合にも、顧客から開発予算を取るだけの営業力が重要である。また仕様変更はまず起こりうるものと考えて、見積もり時に工数の余裕を確保することに努めているというが、競合他社との対抗上、工数を切りつめた見積もりを顧客に提示せざるを得ない場合もあるし、工数のマージンを十分に確保できないのが現状である⁴。

したがって、見積もり工程が極めて重要であることは明白だが、B社とH社の事例から判断すると、技術者（個人・全体）の見積もり能力や営業部門の交渉能力を向上させるという、一見迂遠に思われる対応策が最も現実的であるように思われる。このことは、後出「人材育成」の項目につながる。

4.3.3 派遣と請負

ソフトウェア開発を受注する場合のオプションとしては、技術者派遣（以下「派遣」）および受託開発がある。作業管理は、派遣の場合は派遣先企業が行うが、受託開発の場合は当該企業が行う。受託開発はさらに契約形態によって「請負」および「委任」に分類される。請負には瑕疵担保責任があり、完成責任があるが、委任⁵および派遣にはそれがない。委任が実はソフトウェア開発にとって重要な契約類型であることに気づいたのが最近であるため、十分な聞き取りが出来ていない。したがってここでは派遣および請負についてのみ述べることにしたい。なお、よく言われる作業場所の問題は、契約類型の問題とは全く別個の問題であることに留意されたい。

派遣と請負の区別が本稿の文脈で重要なのは、ある企業の開発業務の中心がいずれなのかによって、当該企業の経営に大きな相違がもたらされるからである。確かに両者は、技術者がある開発に一定期間拘束されるという点では変わらない。にもかかわらず、売上高に占める派遣の比率を減

には、少なくともドイツおよびデンマークに関して、仕様の曖昧さが長時間労働をもたらしているという記述がある。だとすれば本当の問題は、「要求仕様が曖昧かどうか？」ではなく、「要求仕様の曖昧さという問題に対してどのように対処しているか？」であろう。

⁴同社では、現実作業すると見積もられる工数を「内部工数」と呼び、顧客に提示する、マージンを含んだ工数を「外部工数」と称している。

⁵委任契約の典型例は医師の医療行為であるとされる。医師は医療行為を、十分な配慮のもとで行うことになってはいるものの、何らかの治療効果をもたらすことを請け負っているわけではない。この点で、作業成果を約束する請負とは区別されるのである。

らし、請負の比率を増やしたいとする次のような意見は聞かれたが、その逆の意見は聞かれなかった。それには、4.6.3で述べるように、請負でこそ人材育成が可能になると言う判断がある。したがって、人材育成＝企業能力蓄積の面からは、請負の方が望ましいに違いないし、またすぐに以下で述べるような別のメリットもあるのだが、反面、リスクが高く、高い管理能力も必要である。

I社での聞き取りによれば、派遣事業では納期・品質が問われることがないから、経営者にとっては「楽」ではあるが、派遣した人月分の売り上げがあるに過ぎず、単価に上乘せされる収益も少ない。反面、請負の場合は、実際の作業に要した工数が顧客に提示した工数を下回れば、その分だけ利益が大きくなり、その意味では収益機会が小さくない。しかし反面、現実の工数が予定の数倍になってしまうと容易に赤字に転落してしまうのみならず、信用問題になるから、途中で開発をストップするわけにはいかない。そのため、赤字額が巨額に膨らむリスクがある。またそもそも顧客側としても、工程管理や品質管理の能力が不十分な企業には請負で発注しないので、管理能力の蓄積が必要である。

以上のように、確かに多くの経営者は請負事業を中心にしたいと考えているし、一見するとそれは容易なことのように見えるが、請負事業には固有のジレンマがあることも事実である。しかもそれとは別に、派遣事業中心の体質からの脱却を難しくするメカニズムも、I社での聞き取りにおいて次のように指摘された。

派遣中心の体質から脱却し、請負中心に移行し、自社の能力蓄積を推進したいと考える企業は多い。しかし、派遣中心の体質にはまり込んでしまう、次のようなメカニズムがある。好況時には請負で受注していたものの、不況時には請負案件が少なくなるから派遣の仕事も多く取る。なぜなら、不況期だからと言って技術者を遊ばせるわけにはいかないからである。しかし、次の好況時に、派遣された技術者を請負に容易に戻せない。なぜならば、技術者が客先に引きとどめられてしまい、自社に戻ってこないからである。これは優秀な技術者ほど該当する。こうして派遣中心の体質から抜けられなくなるわけである。

確かに日本企業の場合、技術者を長期雇用しているのだから、この産業での慣用語で言うところの、彼らの「稼働率」を上げる必要がある。しかしながらその試みが、派遣中心の体質からの脱却を困難にし、人材育成に注力し請負事業中心の体質へと変えていく道を閉ざしているというのが、上記メカニズムの認識である。この認識が正しいとすれば、長期雇用を行っているからといって自動的に人材育成がなされるわけではないことを意味しており、日本のソフトウェア企業にとって大きなジレンマの一つだと思われる。この点については、改めて第5節で議論する。

4.3.4 自社製品開発の場合の製品企画

開発作業の内容と総量、および投入リソースが大きく決定されてしまうという意味で、自社製品開発における製品企画は、受注管理と同じ意義を持つ。ここでは簡単にだけ見ておきたい。

自社製品を開発しているいずれの企業でも、製品企画においてもっとも優先的に考慮されている項目は「納期」である。納期がまず決まり、工数や投入人員が後で決まるという順序である。

例えばC社の場合、営業部は販売時期の計画を持っており、それは営業部の予算と関係している。その関係上、開発で遅れるわけにはいかない。同様にM社では、まず

「ターゲット」(=機能, 価格, 出荷時期) を決め, 次に開発人数を決める. そうすると自動的に開発予算が決まるという順序である. ただしこの過程でやはり原価企画が行われる. L社でも「発売日は絶対」であり, 開発期間の決定が先, 人数の決定が後であるが, 開発期間を守るために作業をあえて外注に出すこともある.

また, 製品スペックを決める会議体があり, そこで部門間の調整がなされる.

M社での聞き取りによれば, 製品のQCDは最初の「企画・プランニング」段階で大きく決まってしまうので, ここで開かれる3回の会議が重要である. 開発部門に加えて, 購買部門, 品質管理部門, 営業部門, マーケティング部門が参加する. 新製品構想は開発部門から出されるのだが, 「その訴求点では売れない」などの意見が他部門から出されることが多く, それでスペックが固められていく. E社およびL社でも, 同様の会議が行われている.

このように, 開発プロセスを大きく条件付けているのが「納期」であること, また, 納期を計画し主張するのが営業部門であることを考えると, 上記の会議体における営業部門と開発部門の間の「力関係」および具体的なやりとりの実態が, 製品開発を理解する上で重要なポイントであると思われる.

4.4 業績管理

個々のプロジェクトではどのような業績目標が立てられ, それらは全社業績目標や開発部門の業績目標とどのような連関を有しているのだろうか. こうした目標の連鎖を体系的に明らかにすることは, プロジェクトの遂行を全社の収益性に結びつけるロジックを明らかにすることにつながる. ここではまず, プロジェクト目標の例を挙げた上で, 開発部門の業績目標との関係を簡単に見ておきたい.

ISO9001を取得した企業は, プロジェクト遂行に関する計画文書を作成することが義務付けられている. この種の文書には, 当該プロジェクトの「目標」ないし「優先項目」を記入する欄がある. この欄に記入されるのが, プロジェクトの業績目標に他ならない. 先行研究は, 開発部門とプロジェクトとは重視される業績目標が異なっており, 前者では「納期」「品質」が, 後者では「売り上げ」「損益」が重視されていることを明らかにしている(日本労働研究機構[12]). プロジェクトレベルの業績目標を明確に調査できたのはB社およびD社の2社に過ぎないが, いずれも上記先行研究を裏付けるように, プロジェクトが財務的成果に直接の責任を負っているとは必ずしも言えず, 複数の非財務的目標が用いられていることが分かる.

B社では「プロジェクト計画書」を作成している. その中の「プロジェクト目標」という項目で, 例えば「見積もった工数は10人月だが, それよりも少ない8人月でやろう」などの具体的目標が立てられる. プロジェクト目標としては他に「品質に注力」「納期に注力」「戦略的プロジェクト」⁶などがある. またD社でも, プロジェクトごとに「開発計画」という文書を作成する. この文書で特筆すべきことは, プロジェクトの優先項目を記入する欄があることである. 例えば「納期優先」「予算優先」「品質優先」「部員養成優先」などである. 具体的には「予算の95%で完成させる」などの形で記入される.

⁶4.3.1 項を参照のこと.

開発部門の業績目標については、明確な調査が出来ていない。しかし F 社および M 社の例から分かることは、確かに財務指標が含まれているとはいうものの、非財務的な目標もまた軽視されているわけではないということである。

F 社の事業部の業績目標は、売り上げ、作業効率、作業時間、労働時間、その他である。また、事業部の中に作られている「グループ」⁷の目標は、例えば「新技術をマスターする」「リーダーを*名育てる」などである。また M 社の場合、開発部門ではプラン通り（=コスト、機能、スケジュール）に作るというのが部門目標である。したがって予算評価もなされるが、「新技術開発」や「新たなソリューション展開」などの評価項目もある。事業部制だった頃は予算評価のウエイトが大きかったが、現在はトータルな評価となっており、利益ばかりが求められているわけではない。ただし M 社では、財務数値による目標はプロジェクトごとにあり、開発コスト、販売予測、その差額としての粗利益が業績目標となっている。

この 2 社の例から判断する限り、「プロジェクトの目標 = 非財務指標」「開発部門の目標 = 財務指標」というようには、きれいに分かれているとは言えない。開発部門が非財務指標に関する責任を負う場合や、プロジェクトが財務指標に関する責任を負う場合が存在する。既述の通り、機能重視型プロジェクト組織をとっているため、理屈上、個々のプロジェクト目標は部門目標が割り振られたものとなっているはずである。したがって、このことは不思議ではないし、プロジェクトごとに目標が異なっているのもまた自然なことであろう。

開発部門の業績管理とプロジェクトレベルの業績管理の関係についても、十分な調査ができていない。ただし、J 社では、収益を開発部門全体では管理しているが、プロジェクトごとには管理していないという。だがこうした「井勘定」的な管理を脱して、プロジェクトにまで細かな利益管理を及ぼす必要性が、同社において認識されている。他方 K 社では、プロジェクトごとの業績管理が行われているが、その決裁権がプロジェクトの売り上げ規模によって異なる。すなわち規模が大きいプロジェクトとなるほど、課、部、事業部、取締役会というように、決裁権を持つ階層が上がってゆく。しかも両社は、管理指標を売上高から利益に変更しつつある点で共通している。総じて、業績管理を厳格化する方向に向かっていることは、日本労働研究機構 [12] の調査結果とも符合する傾向である。

J 社では、開発部門の業績目標は売上高である。プロジェクトが予定通り進捗するとの仮定の下では、部門のコストは把握できるので、ある利益額をあげるために必要な売上高は自動的に算出されるわけである。ただし今後は、プロジェクトごとに利益を管理する細かい管理方式に移行したいという。K 社では、プロジェクトごと、あるいは製品ごとの利益管理を行うが、それによって当該プロジェクトないし製品の事業としての評価を単純に行っているわけではない。例えば、ポテンシャルのある重要顧客向けの新規案件があれば、長期的視点から、単独プロジェクトは赤字であっても実施される。また、2006 年までは売上高が管理指標であったが、それ以降は利益額となっている。

すると次なる問題は、そうして厳格化された目標を達成可能とするような、実効性ある手段は何かということだろう。

⁷4.3.1 項を参照のこと。

4.5 開発管理

プロジェクトマネジメントの書籍が対象としているのは、主にこの領域である。まず、プロジェクトマネジメントの権限と責任について見る。次に、4.3.2, 4.3.4 および 4.4 で述べた通り、「納期」が重要な目標である以上、進捗管理を確実にすることは重要なことのように思えるが、実際のところ、無体物であるソフトウェアにおける進捗管理はどのようになされるのかを検討する。

4.5.1 権限と責任

プロジェクト管理の権限は、L社およびM社を除いてはプロジェクトマネジャー（以下PM）⁸が持つ。M社にはPMに類する職位・職務は存在せず、開発部門の課長がプロジェクト管理を行う。また、詳しくは後の「人事考課」の項目で述べるが、プロジェクト成果はPMの査定に含まれる。その意味ではむしろ、PMはプロジェクト成果に対する責任を負っているが、必ずしも全責任を負っているとは言えない。それは第1に、PMが失敗を恐れて萎縮してしまうことが、PMの育成にとってマイナスだからであり、第2に、のちに4.6.2で述べるように、異なる難易度を持ったプロジェクトが会社によって各PMに割り当てられる以上、個々のプロジェクト成果に対して一律に責任を持たされるのは不公平であるとの認識があるためである。

典型例であるB社のケースを見てみると、進捗管理、予算管理、工数管理はプロジェクトリーダー（PL）が行い、上司であるグループマネジャー（GM）は、「進捗予定に対して現実の進捗はどうなっているか？」「大きな問題が生じていないか？」というような統括を随時行うだけである。しかし、大きな問題が発生した場合には、GMも問題解決を助ける。このように、権限をPLに与えているが、成果に対する大きな責任はGMが負う形になっている。もちろんPLも責任をとるが、GMの責任の方が重い。全責任をPLに負わせると失敗を恐れ萎縮し、ステップアップしなくなってしまう。その方がマイナスだという認識があるためである。

4.6.1で後述するように、M社も含めて、PMは機能部門の長ないしマネジャーが兼ねている。しかし、例外はE社とL社である。プロジェクト管理を専門に担当する部署を持つ企業（L社）がある反面、プロジェクトと開発部門は全く独立であり、お互いに責任関係を持たないケースもある（E社）。延岡 [11] に倣っていえば、前者は「マトリックス組織」で「軽量級PM」であり、後者は「プロジェクト組織」であると言える。L社の場合、製品の技術的構成が複雑であって、複数の機能部門が開発に関与しているため、部門間調整作業が繁雑であると予想される。そのため、調整と進捗管理に専門特化する部門を設定する方が効率的だと判断しているものと考えられる。またE社の場合、画期的な新製品開発を促すべく、機能部門からのプロジェクトの独立性を高めたものと考えられる。

L社においては、プロジェクト管理は、構想設計部という部署の仕事に過ぎない。製品仕様の最終的な承認権限は各部の部長にある。構想設計部からプロジェクトリーダー（PL）が出されて、PLのもとに、回路設計（システム開発部）、機構設計（機構設計部）、ソフト設計（ソフトウェア開発部）の各リーダーが付き、各々の下につく技術者をまとめる。構想設計部の仕事は、製品仕様を確定させ、会議での承認を得ること、またその過程で文書を残すこと、および、開発の統括・管理・調整である。そのう

⁸企業によっては「プロジェクトリーダー」などの名称で呼ばれるが、ここでは便宜上「プロジェクトマネジャー」に統一して用いる。

ち PL の仕事としては、開発の統括・管理・調整である。特に部門間の調整は PL が担う（例：日程変更）。

E 社の場合、開発部門はプロジェクト成果に関する責任を持たない。そもそも、課とプロジェクトとは、予算も別立てである。このことは、後述 4.6.1 で述べるように、E 社では PM のメンバー人選権限が大きいことと適合的であると思われる。

4.5.2 進捗管理

ソフトウェア開発の進捗管理には、客観的に進捗を把握することの難しさがあり、究極的には技術者の自己申告に依存する。なぜなら、例えばプログラムの行数のような数量的指標での進捗は、プログラムの機能・品質を伴っていない場合もあるから、必ずしも開発作業の進捗を意味しないからである。したがって、進捗管理はプロジェクトマネジャーの能力に大きく依存する。

B 社での進捗管理は、PL が「決められたポイント」(= マイルストーン) で達成度を確認することが中心である。そこでは「予定に対して現実の達成度はどうか？」をチェックしている。ただし、各メンバーの進捗度チェック自体は毎日、チーム内のミーティングで行っている。ただしあくまで、技術者本人の自己申告に依存している。その際、各技術者がどういう性格の人間なのかという特性を掴んでおく必要がある。例えば、進捗の遅れを隠すような性格なのかどうか等が重要。その上で彼らの申告を判断する。

D 社での聞き取りによれば、「行数で管理しても意味がない。品質を問わなければ、行数はどんどん膨らませることが出来るから」とのことである。マネジャーごとに方法があるが、回答者(= マネジャー) は技術者に対してヒアリングをすることにしており、自分のプログラムを言葉で説明させている。そうした方法をとっているのは、次のような理由からである。理解しにくい説明がなされた場合、設計が整理されていない証拠だと判断される。設計が整理されていなければ、作成されるプログラムは低品質になってしまうから、結局は作り直しが発生してしまい、進捗遅れが生じてしまう可能性が高い。

上記のようなことも一因かも知れないが、進捗管理を含むプロジェクト管理技法は、品質や開発効率を改善する上で、限定的な役割しか果たさないという認識がなされている。むしろ先述のように、見積もりを確実に行うことの重要性が強調されているのである。また、こうした管理技法をそのままの形で導入することは困難だとされる。

H 社では、一人ごとの生産高、原価率を毎日オンラインでチェックしている。各人の進捗は日報入力で分かるし、コストは給料から求められるので、理屈自体は簡単である。しかしあくまでこれは、プロジェクトが赤字化する前にメスを入れるための手法に過ぎない。予算超過の理由として「プロジェクト管理がよくなかったからだ」という見解があるが、これは逃げ口上でしかなく、実際には仕様の詰めと正確な工数の確保が重要だとの認識をしている。言い換えれば、プロジェクト管理は問題発見のシグナルであるとの考えである。B 社でも全く同様のシステムで利益管理をしているが、見積もり工数と開発進捗状況とを比較して、順調かどうかを確認し、現状で工数を使いすぎているかどうかを、PL も GM もチェックするのである。

また D 社での聞き取りによれば、教科書的なプロジェクト管理手法は確かに合理的であるが、無理やりにそれらを当てはめるとかえって無駄ができ、納期に間に合わな

いことがあると認識されている。納期や予算の制約がますます厳しいし、慢性的に人員も足りない。このような制約条件を考えると、教科書通りにプロジェクト管理を適用することは現実的ではないということである。

以上のように、進捗管理の困難さがある上、それを含むプロジェクト管理技法にも、その効果および実施に限界があるために、ソフトウェア開発のパフォーマンスにとって、進捗管理を含むプロジェクト管理技法は一般に考えられるほど重要な意味を持たないと認識されている。それゆえに、H社の聞き取りで挙げられているように、受注管理が極めて重要な意味を持つことを、再度強調しておきたい。同時に、4.4で既述のように、プロジェクトへの業績管理が厳格化する傾向にある割には、プロジェクトには所望の業績を実現するための手段が余りないのではないかと考えられる。この点は第5節で簡単に議論する。

4.6 人事管理

明らかにソフトウェア開発は(知的)労働集約的なプロセスであるから、人事管理は中核的な重要性を持っている。今野・佐藤 [2] によれば、人事管理は本来「雇用管理」「就業条件管理」および「報酬管理」の広範な領域をカバーするものだが、本稿では不十分さを承知で、企業競争力を考える上でとりわけ重要と思われる「人員の配置」「人事考課」および「人材育成」だけに焦点を絞って扱う。

4.6.1 プロジェクトへの配置

誰がどのプロジェクトのマネジャー、もしくはメンバーとして配置されるのかは、プロジェクトの成果にとってはもちろん、組織の能力蓄積、技術者個人のキャリアパス、能力形成にとっても重要な問題である。部門横断的なプロジェクトが組まれる機器開発の場合とは異なり、調査を行ったソフトウェア開発の場合、先述の通り、開発部門の中でプロジェクトが組まれる。

プロジェクトマネジャー (PM) は、候補者の中から各人のコア技術を基準に選定されてなるパターンと、開発部門の部長、課長、ないし部門内グループの長がそのまま PM になるパターンがある⁹。

例えば前者の例である A 社の場合、各人の持つコア技術によって PM が決められる。また、任されるプロジェクト規模は、経験によって決められる。また E 社でも、PM は幹部会議での合議によって決められる。この際に重視されるのは、各人の専門分野である。

それに対して、後者の例である B 社の場合は、開発部門内に開発分野ごと(アプリケーションソフト/ファームウェア/デバイスドライバ/評価、等)に分けられたチームが複数ずつあり、ある案件の開発分野や、その時点での各チームの繁忙などによって、どのチームに当該案件を割り振るかが決められる。この場合、チームがそのままプロジェクトとなり、チームの長がそのままプロジェクトリーダーとなる。F 社においても、開発部門内に複数ある「グループ」¹⁰を率いるチーフリーダー、もしくはグループ内でチーフリーダーの下に複数いるリーダーが PM をする。C 社および M 社では、部長および課長が一手に、複数のプロジェクトの PM を引き受ける。

⁹ただし、4.5.1 で述べたように、機器メーカーである L 社では、プロジェクト管理は構想設計部という部署の職務であるに過ぎないため、PM は構想設計部員から選ばれることになる。

¹⁰F 社の組織については、4.6.3 項を参照。

また、4.6.3項で詳述するように、PMの選定においては人材育成という側面も考慮されている。

メンバーは、技術者の能力や経験、繁閑を考慮して選ばれるが、複数のプロジェクトを掛け持ちする場合は通例である。メンバー選定には、PMが全面的な権限を持つ場合が多い。

部門横断的にプロジェクトが編成されるL社では、各部（ソフトウェア開発部、システム開発部など）からメンバーが集められる。得意分野が重視されるが、各部の部長が、プロジェクトメンバーを誰にするかという構想を持っている。そのため、PMを出す構想設計部はメンバー選定に主導権を発揮しない。

逆にE社は、PMは開発部門の意向から独立して、メンバー選定権限が最も大きい事例だと思われる。メンバーは、技術者の能力（適不適）でPMが決めるのだが、その人が多忙かどうかを度外視してPMが希望を出し、その後に調整を行うという形で選ばれる。このようにPMの人選権限が大きいことは、開発部門からのプロジェクトの独立性が大きいという、上記4.5.1で述べたE社の特徴と適合的であろう。

E社、L社以外では、メンバー選定をPMが完全に行う場合と（例えばB社、C社）、PMの希望を上司と調整して決定する場合とがある（例えばF社）。しかしいずれの場合でも、A社、E社、L社以外では、PMは開発部門の部長ないし課長クラスがそのままなので、開発部門での人材育成方針などとの整合性を取りやすいであろう。

以上のように、E社のケースを除けば、メンバーの人選に関しては開発部門の意向が強く反映される仕組みであることが分かる。したがって、プロジェクト制とは言っても、機能重視型プロジェクト組織であることが再び確認される。そのため少なくとも組織構造上は、戸塚・中村・梅澤[9]の理解とは異なり、人材の長期育成と適合的であると思われる。もちろん、現実に人材育成が難しく進んでいるかどうかは別問題であるし、また、ソフトウェア開発に短期的な柔軟性が求められていることは変わらぬ事実なので、長期的視野を要する人材育成と矛盾する局面が皆無になるとは考えにくい。この点については、第5節で議論する。

4.6.2 人事考課

業績がどの程度、どのような形で人事考課の対象になっているのかは、今日の文脈から考えて重要な問題である。また、その方法が技術者のクラスによってどのように異なっているのかも明らかにする必要がある。残念ながら人事考課については、体系的な聞き取りが出来ていないものの、管見する限り、概してPMクラスだとプロジェクト目標の達成度が人事考課で問われるものの、個々のプロジェクトの成果、とりわけ財務面の業績を厳密に評価することに対しては、躊躇も見られる。それは第1に、あるプロジェクトが開発した製品・サービスの収益は、当該プロジェクトのみの責任に帰することが難しいためである。第2に、例えば難易度が高い案件、あるいは新人メンバーを含む案件＝「外れ案件」をあてがわれたPMにとって、不公平な評価になってしまうと認識されているからである。これに対して、一般の技術者に対する考課は、成果に比してプロセス、例えば形成した能力のウェイトが大きい。また、「目標管理制度」が多くの企業で運用されている。

例えばB社では、プロジェクトリーダーは「プロジェクト計画書」に書かれた「プロジェクト目標」が達成できているかどうか評価され、その意味で、考課におけるプロジェクト成果の反映度は大きい。しかし、若手になるほど「プロセス重視」であり、プロジェクト目標達成度が評価されるウェイトは小さい。反面、目標管理制度で設定さ

れた「技術力向上」などの目標の達成度のウェイトが大きい。同様に M 社でも、一般技術者の考課では、プロジェクト成果はほとんどリンクしていない。せいぜい「スケジュール遵守」「品質」が、＜日頃の素行＞的な項目として入っている程度である。この背後には、エンジニアは成果との対応が弱いから、成果主義は適用しにくいとの意向がある。このように、ベテランと若手で、評価項目に対して異なったウェイトが設定されているのは、多くの企業で共通している。

ただし、PM の考課においても、プロジェクト成果の反映度合いが抑制されているという側面もある。

例えば M 社では、開発部長は予算達成で考課される比率が大きいですが、課長 = PM に関しては、プロジェクトの失敗は、余程のことがなければ考課に反映されない。その理由の一つは、確かにプロジェクト目標には粗利益が掲げられてはいるものの、粗利益が低いのはプロジェクトのみの責任ではないという考え方があるためである。したがって課長の評価は、売り上げ・利益が 50%、「とりくみ」が 50%という比率である。

PM の考課で個々のプロジェクト成果を評価すると、PM の不満を生むという事情も存在する。

F 社および J 社の聞き取りによれば、たとえリスクが大きいチャレンジングな案件であっても、技術蓄積目的、あるいは顧客と取引関係を構築する目的で、赤字覚悟でも戦略的に受注することがある。したがっていわば「当たり物件」と「外れ物件」があるのだが、「外れ案件」までプロジェクト成果を評価することは、それを担当させられた PM の不満を生むという。したがって、プロジェクト成果を厳密に反映させるわけにはいかないのである。また、OJT のために新人を入れると、かなり余分にチェックなどの工数が必要になるから、そのプロジェクトの効率は落ちるのが普通である。そのため、新人を受け入れざるを得ない PM は不満を持ちやすいという (H 社)。こうした事情もまた、プロジェクト成果を厳密に考課に反映させることを難しくしていると思われる。

4.6.3 人材育成

これまで繰り返し言及してきたような、人材育成とプロジェクト制は相容れないという認識もあり、多様な案件に対して短期的に確実な成果を求めるソフトウェア開発の中で、長期的な視野からの人材育成がどのように行われているのかというのは、極めて重要な問題である。

人材育成の方法は、OJT と座学によるものがある。OJT に取り組んでいない企業は、少なくとも調査対象企業にはなかった。しかし後述のように、OJT を体系的に行おうとすれば無視できないコストがかかる。したがって、ある企業が「OJT を行っている」と称していても、当該企業がどの程度まで体系的に実施できているかどうかは、また別問題である。

H 社での聞き取りによれば、情報系の場合、ソフトウェアが用いられる文脈での業務知識が必要なのだが、これは顧客との接触から学ぶしかない。反面、座学によるプログラミング言語の社内教育は比較的簡単である。ただし、I 社のような組み込み系開発の場合、プログラミング技術の高度化が速いので、社内教育では対応しきれない。そのため、教育予算を設定し、外部の講座に社員を派遣する。しかし、日常業務と講座の時間ががち合ってしまうなど、外部講座受講にも固有の困難がある。

実際にはOJTは、アドホックではなくかなりシステマティックに行われている。しかしその分、以下のようにかなりのコストがかかると認識されている。

派遣業務も行っているG社の場合、OJTに適した案件が常にあるとは限らない。その場合、あえて社内にダミーの案件をつくる。例えば、自社用の次世代情報システムプロジェクトを作るのである。ただしOJTは請負のみならず派遣でも行っており、例えば客がとにかく技術者人数を求めている場合に、「代金はいりません」と交渉をして、新人を教育用にあてがうこともする。

また、D社での聞き取りによれば、「納期やコストを優先するのなら、経験者だけで行えばいいので話は簡単」である。なぜなら、若手をプロジェクトに入れると、トレーナーを付けねばならないなど、コストがかかるからである。ただし、プロジェクト規模が大きければ、そのコストも何とか吸収できる。また、プロジェクト数が全体として多ければ、ある一定数のプロジェクトが若手育成優先であってもコストを吸収できる。さらに、リスクが小さいプロジェクトならば若手育成をしやすい。同様にJ社でも、新人教育を含むプロジェクトはテストなどで工数がかさむが、プロジェクト総数が多ければコストは吸収できるとする。その意味で、OJTのコストを吸収するためにも、プロジェクトに関するいわば「規模の経済」「範囲の経済」が重要だという認識がなされている。

OJTは新人のみを対象としているのではない。多くの企業で、PMにあえて能力以上のプロジェクトを任せたり、あるいは「PM 一歩前」の若手技術者に小さいプロジェクトを任せている。こうした場合、任せた上司が彼らの仕事をフォローする必要がある。

開発部門の部長や課長には人材育成の責任があり、それが人事考課の評価項目にもなっている。しかも、A社、E社、L社以外では、彼らがそのまま部下を率いてPMとなるのだから、事実上、プロジェクトにおける実際の人材育成にも責任を持つことになるわけである。したがって、確かにプロジェクト制のもとでOJTを体系的に行うコストは大きいものの、戸塚・中村・梅澤 [9] に言われる「矛盾」は相当程度、現実的に解消されていると見てよいのではないだろうか。

なお、OJTを効果的に行うためには、派遣ではなく請負で受注を行う必要がある。派遣の案件の場合、作業管理は顧客によって行われるので、OJTを組織化する上で自由度が低いためである。また、派遣案件が中心になってしまうと、そもそも技術者の定着率自体も低くなってしまいうという問題も指摘されている。

G社では、事業が派遣中心であるとOJTが効果的に出来ず、また社員の評価もしにくいいため、派遣の比率を低めて請負の比率を上げるのが理想だと考えている。反対に請負の場合、顧客が出す条件は「納期、システム名称、仕様」であり、どういうグループで作業するのかは、先方と関係なく決められるので、OJTの組織化においても当社の自由度が大きいのである。I社の場合も、派遣中心だと人材育成が困難という認識は同様である。I社はかつては派遣事業が中心であったが、あまりに離職率が高いことを問題視し、請負事業を中心に行うように改めてきた。

またF社のように、現実的には派遣の仕事がゼロになることはないものの、極力、技術者が技術を吸収し、会社に技術が残る派遣案件のみを行うというように、派遣の受注に際しても案件の選別を行っている事例も存在する。

5 討論

以上に叙述した経営管理実践が日本企業の競争力を規定している。本節では、そのうちでもとりわけ重要だと考えられるいくつかのポイントを指摘し、その含意を議論する。

5.1 見積もりの重要性

上記 4.3.2 で明らかにしたように、見積もり工程が開発プロセス中で最も重要だと認識されている。また 4.5.2 で述べたように、進捗管理を含むプロジェクト管理は、開発プロセスで限定的な役割しか果たしていないとされる。言い換えれば、「問題を検知しそれに対処する能力」よりも「問題を生じさせない能力」の方が重要だと認識されていると言うことを強調したい。こうした認識は、プロジェクト途上で発生した問題に対して有効に対処する方法が限られているという事態を反映しているものと考えられるが、もしそれが正しければ、小池 [6] がつとに強調する「問題への対処と変化への対応」としての「知的熟練」の意義は、ソフトウェア開発プロジェクトの、少なくとも下流工程になるほど、大きくはないということを意味しよう。

このことは、工数・予算超過と言った問題を解決する有効な手段を、各プロジェクトおよび PM が十分に持ち合わせていないことを意味すると思われる。そのことは確かに、個々のプロジェクトについて業績評価を行わないという、4.4 および 4.6.2 で見られた慣行と整合的ではある。なぜならば、所望の業績を得るために有効な手段が不十分な条件下で業績評価を行えば、PM が不公平を感じることは避けられないと考えられるためである。

5.2 ウォーターフォール方式という「技術選択」

見積もり工程が重要だというのは、4.2 で述べたように、調査対象企業を含む日本企業の多くが「ウォーターフォール方式」と呼ばれるソフトウェア開発プロセスを選択・採用している¹¹ことと関係があるだろう。ウォーターフォール方式は、まずは要求仕様を確定した上で、システム設計、コーディング (= プログラミング)、テストの各工程を順次、逆戻りがないように進めていく。したがって、要求仕様を確定した後の工程は、その仕様を徐々に詳細化・具現化していく工程であるから、要求仕様を確定する見積もり工程でのミスが致命的であることは自明であろう。

日本企業が選択したウォーターフォール方式は、決して唯一の開発方式ではなく、複数存在するソフトウェア開発方式の「一つ」でしかないことに留意する必要がある。例えばクスmano [5] によれば、米国企業で一般的になっているという「同期安定化プロセス」という開発方式は、そもそも要求仕様を事前に確定させるという考えを拒否しているという点で、仕様変更に対して脆弱なウォーターフォール方式と対照的な開発方式であり、開発途上でも頻繁にユーザによる評価を行い、そこで生じる仕様変更を当然のものとして許容するように構造化された開発プロセスを有しているという¹²。

¹¹ 実際、『2005年版組み込みソフトウェア産業実態調査』（経済産業省）によると、組み込みソフトウェア開発において日本企業では35%超のプロジェクトでウォーターフォール方式が採用されているのに対して、米国企業では5%程度で採用されているに過ぎない。またクスmano [5] によると、日本企業では仕様書を明確化するウォーターフォール方式が主流であるが、米国企業では、詳細な設計書をほとんど作成しない開発方式が一般的になっているという。

¹² だからといって、あらゆる面において「同期安定化プロセス」の方が優れているということの意味しない。クスmano [5] およびマクブリン [14] によれば、極めて高い品質が求められる組み込みソフトウェアの場合、ウォーターフォール方式は適切な開発方式だという見解も有力である。だが反面、組み込みソフトウェアであっても携帯電話向けソフトのように頻繁な仕様変更にさらされる分野の場合、仕様変更に対して脆弱なウォーターフォール方式のマイナス面が大きいのではないだろうか。したがって、開発方式における one best way は存在しないと見るべきであろう。

日本企業の技術 (= 開発方式) 選択, すなわちウォーターフォール方式という技術選択は, 日本のソフトウェア産業を特徴付ける, 外注依存度の高さと整合的である。定義上, 要求仕様が決められないと請負契約にすることは不可能だが, 要求仕様確定を最優先することは, まさにウォーターフォール方式の最大の特徴に他ならないからである。また, 上流工程 (= 要求仕様を確定する要件定義工程のみ, もしくはシステム設計工程まで) を内製し, 下流工程を外注するという内外製の慣行も, 上流工程こそがクリティカルであるウォーターフォール方式と適合的である。加えてウォーターフォール方式は, SE とプログラマの間に区別が存在し¹³, また, SE の方がプログラマ (=PG) よりも「付加価値」が高いとされ, 「PG < SE」というキャリアパスが一般的である人事管理慣行を形作っていると考えられる¹⁴。

ウォーターフォール方式の下で, 全社の業績管理強化に対応してプロジェクトへの業績管理を厳格化することには, 一定の限界があるものと考えられる。なぜならば, 実際にプロジェクトが問題解決手段を持っていて初めて, そうした業績管理強化は実効性を持つはずだが, ウォーターフォール型の原理を維持した場合, 所望の業績を達成するための手段がプロジェクトに十分に与えられているとは考えにくいからである。そうだとすれば, 全社の業績管理強化に対する対応策としては, 開発プロセスの改善・規格化や, 見積もり方法の規格化と言った, 迂遠に見える組織能力の体系的構築が主要な方法となると考えられるのである。事実, 本稿では触れなかったが, 調査対象企業の多くでは, 品質管理やプロジェクト監査を専門とした部門を設置し, 開発プロセスの規格化に注力している。このことから, 業績管理強化に対する対応策は, 開発プロセスの選択, ひいてはプロジェクト組織形態の選択によって異なってくると考えられるだろう。

5.3 プロジェクト組織形態の選択

4.5 および 4.6 における記述から分かるように, プロジェクトは機能部門内で組織され, プロジェクトマネジャーを機能部門長ないしはマネジャーが兼ねるのが通例であった。また, 人選についても機能部門の意向が反映されていた。したがって, 調査対象企業の組織形態は「機能重視型」(延岡 [11]) プロジェクト組織であると言えるだろう。この組織形態は, プロジェクトマネジャーの権限が強く, プロジェクトの機能部門からの独立性が強い組織形態とは明らかに異なっている。例えばトヨタにおけるいわゆる「重量級 PM」のような, マトリックス組織でプロジェクトマネジャーの権限が強い組織形態とも異なるし, またクスマノ [5] が「同期安定化プロセス」について述べるような, 個々のプロジェクト¹⁵が(ある限度内での)機能変更の権限を持ち, したがって PM が進捗管理以上の権限を有している組織形態とも異なる。

調査対象企業が機能重視型プロジェクト組織を選択し続けてきた理由としては, PM に権限を多く与える必要がない, 換言すれば PM が裁量を振るう余地が少ないウォーターフォール方式が選択されてきたということがあるだろう。つまり, 開発方式がウォーターフォール方式の場合, この組織形態を選択することは合理的だと思われる。これは既述の論点なので繰り返さないが, ウォーターフォール型の場合, 上流工程, 例えば要件定義を機能部門で担い, 下流工程だけをプロジェクトで行う, と言った形の分業をしやすいことは明白であろう。

¹³ 「アメリカでは, システムエンジニアはプログラマと同義であり, アメリカのプログラマは, 日本で言われるプログラマだけでなくシステムエンジニアの仕事も兼ねているのが一般的である」(<http://ja.wikipedia.org/wiki/システムエンジニア>)とされているのが事実だとすれば, 国際的に見て, この区別は必ずしも自明のこととは言えない。

¹⁴ 日本のソフトウェア企業においては, SE への育成が重視され, コーディングのみを行う PG はその通過点に過ぎないと考えられている側面がある。言い換えれば PG は軽視されてきたわけであるが, このことを反省・問題視する見解が経営者側からさえ出てきている。例えば, 「システム開発は楽しい」NTT データ 浜口友一社長に聞く」(<http://www.sankei.co.jp/keizai/sangyo/070323/sng070323000.htm>) を参照。

¹⁵ クスマノが念頭に置いているのは, マイクロソフト社におけるアプリケーションソフトの開発である。そうした開発案件は, 複数の小さなプロジェクトに分割され, 各グループが一定の自律性を持って開発できるように組織されるという。

また、人材育成を社内で行わなくてはならない日本の雇用慣行の下では、機能重視型プロジェクト組織という選択は合理的であると考えられる。この組織形態では、機能部門の長・マネージャーがPMだから、人材育成と適合的な人員配置が容易となるためである。

5.4 受注と雇用・取引慣行の関係：短期収益指向の抑制

案件ごとの収益は見積もりで大きく決まってしまうということは既に述べた。だとすると、企業全体の収益は様々な案件の組み合わせ方（＝ポートフォリオ）によって決まるはずである。したがって、受注方針は収益性にとって重要な意味を持つが、この方針は以下のように雇用・取引慣行に大きく規定されて、短期的な収益性を最大化する受注行動は抑制されざるを得ないと考えられるのである。

4.3 での記述を踏まえると、企業が受注に際して、「収益」以外に考慮している項目は、「人材育成」「稼働率」および「取引関係構築」である。第1に、人材育成の必要から、プロジェクトの「規模の経済」「範囲の経済」を実現しうるような受注が促される。第2に、長期雇用技術者の稼働率を維持する必要から、受注数量の確保が促され、とりわけ不況期には不利な「派遣」の案件までを受注するように促される。また第3に、新規顧客と長期継続的取引関係を結ぶために¹⁶、たとえ不利な条件であっても「戦略的受注」がなされうる。これら3つの要因は雇用慣行、取引慣行に根ざしたものであり、いずれも収益の短期的最大化をはかるような受注を抑制せざるを得ないことは明白であろう。加えて、「人材育成」と「稼働率」とが相容れない場合もある。なぜならば、例えば稼働率を最優先項目とした場合、派遣の案件を受注することが最も安易な方法であるが、しばしば派遣案件では人材育成が困難なためである。

こうした事情を考えるならば、確かに近年多くのソフトウェア企業では、業績目標において収益性を今後一層重視したいとしているが（日本労働研究機構 [12]）、長期雇用慣行や取引慣行との両立は容易ではない課題だろう。また、開発部門の長・マネージャーが営業も兼ねて行うケースでは、この課題の解決はひとえに彼らにかかっているため、彼らに十分な権限を与え、人事考課と整合性をとることが早晩、現実的課題となるだろう。

6 まとめ

繰り返しになるが、筆者の当初の問題意識は、国際分業が進む中で日本のソフトウェア産業が占める位置を問うということであった。その課題に答える上でまず、日本のソフトウェア産業の競争力を左右する仕組みを明らかにすることが必要だが、そのためには、経営管理を体系的に明らかにする必要があると考え、そこから競争力を規定する仕組みを抽出する必要があるだろう。本稿はそのための準備作業であり、第4節で経営管理を体系的に叙述し、そのうち重要と考えられる特徴について第5節で検討した。そこで検討したのは(1) 見積もりの重要性の意味、(2) その背後にある開発方式選択の意味とロジック、(3) その選択に連動したプロジェクト組織形態選択の意味とロジック、および(4) 受注における短期収益志向の抑制、という事項である。いずれも日本の雇用・取引慣行と、一定の適合性があることが論じられたのである。雇用・取引慣行のような制度的要因、技術選択、およびソフトウェア企業の経営管理の関係について解明するためには、本格的な国際比

¹⁶よく知られるように、日本企業において、長期継続的取引を行う企業どうしは「基本取引契約」を締結した上で、本来なら個々の取引に対して締結されるべき「個別取引契約」は省かれることが多い。したがって、顧客と基本取引契約を締結することが、長期継続的取引関係の構築にとって決定的に重要であり、そのためにはたとえ不利な条件の開発案件であっても受注せざるを得ないのである。

較と、一層詳細な聞き取り調査が必要だろうが、日本における制度的要因が技術選択、ひいては経営管理にどのような連関を持っているのか、そのロジックの一端は明らかにできただろうと考えている。

本稿は本格的な研究の準備作業でしかないため、新たに発見された課題は多い。ここでは、めばしい課題のみを4点だけ挙げておきたい。

第1に、工数・予算の超過などの「問題への対処と変化への対応」(小池 [6]) が実際にどのように行われているかということである。確かにウォーターフォール型開発の原理に忠実であれば、事後的な対処・対応によって問題を解消するのではなく、本文で強調した通り、見積もり段階で問題の芽を事前に摘んでおくことこそが重要である。事前の問題解決が強調されるのは、1990年代末以降の半導体産業と全く同じであり、技術・製品の複雑さへの対応策としては一定の合理性がある (Tokumaru [28])。しかし現実には、仕様変更および仕様の曖昧さに起因する作業手戻りがしばしば発生し、工数・予算超過をもたらす。こうした問題に対しては、本稿で触れた「見積もり能力の向上」しか対応策がないのだろうか。あるいは現実には、別様の対応策が採られているのだろうか。そうだとすれば、それは誰によって担われ、どのような能力を要請するものなのだろうか。例えばプロジェクトレベルにおいて可能な対応策はあるのだろうか。頻発する仕様変更に対する対応能力は、ソフトウェア企業の競争力を大きく左右する重要要因だと考えられるから、この解明、とりわけ国際比較に基づく解明は重要な課題である。

第2に、上記の課題と関連するが、開発プロセスの技術選択についてである。第5節では、ウォーターフォール型という選択が現行の雇用・取引慣行と適合的であることを論じ、またこの技術選択が日本企業のソフトウェア開発にとってきわめて重要な意味を持っていることを述べたが、なぜ現実にはウォーターフォール型が選択されつづけているのだろうか。あるいは、他の開発プロセスを選択した場合に、どのような組織的変更が必要となるのだろうか。例えばクスマノの言う「同期安定化プロセス」に類する開発プロセスを採用する場合、PMの権限を拡大せざるを得ない。そのことは必然的に人事管理をはじめとする経営管理全般に影響を及ぼすだろう¹⁷。こうした技術選択のロジックを、とりわけ国際比較により解明することは、開発プロセスと雇用・取引慣行の関係を一層深く理解する上で必要だと思われる。

第3に、採用している開発モデルと、業績管理強化に対する対応方法との関係についてである。既に第5節で述べたように、業績管理の強化に対してどのように対応するかは、開発モデルに依存すると考えられる。したがって、ウォーターフォール型を選択している多くの日本企業では、業績管理強化に対してある特定の傾向を持った対応方法をしているはずであるし、諸外国企業とは異なる対応を行っているのではないかと考えられる。このことを国際比較によって解明したい。

第4に、人材育成方針と外注の関係についてである。日本企業は自社で人材育成を行わなくてはならないが、そのことは外注範囲を決定する上で制約条件となるだろう。言うまでもなく、過度な外注依存は自社の人材育成にとってマイナスとなりうるからである。どのような工程を外注するのか、またそれを決定する基準はどのようなものなのかを明らかにする必要がある。また、派遣契約で外注するのか、あるいは請負契約で外注するのかの切り分けは、受注企業にとって大きな相違を生む選択である。その選択基準についても明らかにしたいと考える。むしろこれも上と同様に、国際比較によって取り組むべき課題であろう。

¹⁷ 妹尾 [8] は、非ウォーターフォール方式を採用した企業の事例を検討し、そこでのPMの役割が質的に変容していることを述べている。創造性・即興性が重視されるそこでのリーダーシップを、彼は「状況論的リーダーシップ」と呼ぶ。しかしそうした技術変化の背景では、経営管理の体系的変化が付随していると予想される。その局面に光を当てる必要があるだろう。

参考文献

- [1] 今野浩一郎・佐藤博樹, 1990, 『ソフトウェア産業と経営』 東洋経済新報社 .
- [2] 今野浩一郎・佐藤博樹, 2002, 『人事管理入門』 日本経済新聞社 .
- [3] 今野浩一郎編, 2003, 『個と組織の成果主義』 中央経済社 .
- [4] 石田光男, 2003, 『仕事の社会科学』 ミネルヴァ書房 .
- [5] マイケル A. クスマノ, 2004, 『ソフトウェア企業の競争戦略』 ダイヤモンド社 .
- [6] 小池和男, 1999, 『仕事の経済学』 東洋経済新報社 .
- [7] 佐藤厚編, 2007, 『業績管理の変容と人事管理』 ミネルヴァ書房 .
- [8] 妹尾大, 2001, ソフトウェア開発の新潮流：状況論的リーダーシップの胎動, 『組織科学』 35(2): 65-80 .
- [9] 戸塚秀夫・中村圭介・梅澤隆, 1990, 『日本のソフトウェア産業：経営と技術者』 東京大学出版会 .
- [10] 西村克己, 2000, 『よくわかるプロジェクトマネジメント』 日本実業出版社 .
- [11] 延岡健太郎, 2002, 『製品開発の知識』 日本経済新聞社 .
- [12] 日本労働研究機構, 2000, 『調査研究報告書 No. 134：情報産業の人的資源管理と労働市場』 日本労働研究機構 .
- [13] 古谷真介, 2007, 『パソコン・ブーム下におけるパッケージ・ソフトウェア開発の作業組織研究』 東京大学社会科学研究所研究シリーズ No. 23 .
- [14] ピート・マクブリー, 2002, 『ソフトウェア職人気質』 ピアソン・エデュケーション .
- [15] 李美多・高橋信弘, 2006, 日本の情報サービス産業における海外へのアウトソーシングの進展：中国・インドへのオフショア開発の現状とその影響, 『経営研究』(大阪市立大学) 57(3): 79-97 .
- [16] Arora, A. and Gambardella, A. eds., 2005, *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland, and Israel*. Oxford University Press.
- [17] Baba, Y., Takai, S. and Mizuta, Y., 1995, The Japanese software industry: the 'hub-structure' approach, *Research Policy* 24: 473-486.
- [18] Barley, S.R. and Kunda, G., 2001, Bringing work back in, *Organizational Science* 12(1): 76-95.
- [19] Casper, S. and Vitols, S., 2006, Managing competencies within entrepreneurial technologies: a comparative institutional analysis of software firms in Germany and the UK, in Miozzo and Grimshaw[27].

- [20] Casper, S., 2007, *Creating Silicon Valley in Europe*. Oxford University Press.
- [21] European Trade Union Institute, 2001, *New Forms of Employment and Working Time in the Service Economy*. European Trade Union Institute.
- [22] Grimshaw, D. and Miozzo, M., 2006, Institutional effects on the IT outsourcing market: Analysing clients, suppliers and staff transfer in Germany and the UK, *Organization Studies* 27: 1229-1259.
- [23] Hobday, M., 2000, The project-based organisation: an ideal form for managing complex products and systems?, *Research Policy* 29: 871-893.
- [24] Ibert, O., 2004, Projects and firms as discordant complements: organisational learning in the Munich software ecology, *Research Policy* 33: 1529-1546.
- [25] Lam, A., 1996, Engineers, management and work organization: a comparative analysis of engineers' work roles in British and Japanese electronics firms, *Journal of Management Studies* 33(2): 183-212.
- [26] Miozzo, M. and Grimshaw, D., 2005, Modularity and innovation in knowledge-intensive business services: IT outsourcing in Germany and the UK, *Research Policy* 34(9): 1419-1439.
- [27] Miozzo, M. and Grimshaw, D. eds., 2006, *Knowledge Intensive Business Services: Organizational Forms and National Institutions*. Edward Elgar.
- [28] Tokumaru, N., 2005, The codification of technological knowledge, technological complexity, and the division of innovative labour: a case from the semiconductor industry in the 1990s, in Finch, J. and Orillard, M. eds., *Complexity and the Economy: Implications for Economic Policy*. Edward Elgar.
- [29] Whitley, R., 2006, Project-based firms: new organizational form or variations on a theme?, *Industrial and Corporate Change* 15(1): 77-99.
- [30] Yin, R.K., 1994, *Case Study Research: Design and Methods*. Sage.