経済論叢別冊　調査と研究（京都大学）第3号，1992年10月

〈アンリ・ナデル　シンポジウム〉

日本におけるソフトウェア業の賃労働関係

ルイス・アルベルト・ディ マルティノ

I　はじめに

本稿で私は，ソフトウェア産業で現在進行中の賃労働関係の制度化について，いくつかの類型をあげて論じたい。この産業は，近年きわめて高い成長を示した（表1参照）だけでなく，製造業及びサービス業にたいしてその独立性を強めてきている[1]。そのためこの産業では，ソフトウェアの生産過程の特徴に対応したかたちで，労働力の管理を容易にするための制度的枠組みが必要になっている。

具体的な問題に入る前に，まず賃労働関係に一般的な特徴とそれらに関わる制度についての復習をおこないたい。それは，後で示す具体的な説明を資本主義の社会関係の本質的な特質と関連づけて理解するためである。

本稿は，課業遂行のための労働組織，従業員の教育・訓練，賃金基準，評価体系，昇進，キャリアなどの，人事管理の典型的テーマを論じる。資本主義的企業のもとでは，一定タイプの仕事をするために雇用された従業員の労働力の使用が雇用者に委ねられる。経営側はこの労働力を実際の労働に変えるのであるが，その労働の生産性をあげるために，教育・訓練をおこなう。それらは，再投資あるいは利潤部分によっておこなわれるその費用にたいして効果が最大になるようにおこなわれるであろう。

資本主義は利潤が中心的な変数であるような社会関係である。企業の目的について高い利潤分配ではなく長期の成長が選択されると論じられることもあるが，企業の成長も利潤の再投資による資本の増大がなければ可能ではない。利潤をあげるためには剰余（資本主義のもとでは剰余価値）を生み出すことが必要であるが，その源泉は労働（資本主義のもとでは賃労働）である。労働の生み出す価値は，経営のやり方とそれに対する労働者の態度によって変わる（その意味で労働力は可変資本である）。経営者も労働者も意識と意志をもった人間であり，人事管理は経済的のみならず政治的，イデオロギー的側面をもつ。それを論じるためには，一種学際的な視点が必要であろう。制度の重要性はこれと関連している。制度は一般にいろいろなグループのあいだの利害関係を安定させる役割をもつが，企業の労使関係にかかわる制度は，対立する利害関係から生まれる不安定性が生産単位としての企業の存立を脅かさないようにする。人事（労務）管理はこの制度の一部である。

技術革新によって労働生産性が上昇している場合でも，上記のようなことが妥当する。ここで述べておきたいことは，第一に，人事管理は技術発展と無関係ではないことである。労働者が新生産物や新生産方法の構想に参加しない

表1　情報サービス産業の成長

（単位：億円）

	企業数	従業員数	売上高
1985	2,556	162,010	15,618
1986	2,808	198,522	19,159
1987	3,692	241,187	22,993
1988	5,627	333,587	32,973
1989	5,587	377,113	43,514
1990	7,042	458,462	58,727

資料：情報サービス産業白書　1992年版

1）　今野・佐藤(1990) 1章3節及び北谷利之(1990)，参照。

テーラー主義からいわゆる「日本式管理」（技術的進歩を可能にするアイデアを経営に提供するような積極的態度を労働者に奨励する）に進む中で，人事管理担当者は，技術進歩がおこる組織の境界を決定する制度をも含めて，剰余を増加させるための最善の方法をさがしもとめているのである。第二には，研究・開発もまた，人事管理の体系にしたがうということである。人事管理が労働力を現実の労働に変換し資本蓄積を最大化することを目的としているということが正しいならば，研究・開発部門での技術者の活動をコントロールする人事管理は，社会制度と技術進歩が密接に関連した領域に位置する問題である。

この報告でとりあげるソフトウェアの生産には，人事管理と技術革新にかかわる次のような特徴をもっている：1）生産性の向上にとって，ソフトウェア・エンジニアの知識と技能のもつ意義は，プロセス・イノヴェーションやプログラムの部分的再利用[2]におとらず大きい[3]。2）ソフトウェアの需要は日本のソフトウェア産業の供給能力を上回るが，その根拠は人員不足と生産性の上昇が緩やかなことである。これは，ソフトウェア産業の生産性上昇が，大部分，システム・エンジニアやプログラマーの教育・訓練や仕事経験に依存するという1)の特徴と関連している。3）ソフトウェア産業の生産物の大半は，民間企業や政府機関でおこなわれる情報処理のための専用ソフトである。迅速で信頼性のある情報システムを構築することは，民間産業がその生産性を上昇させるためのファンダメンタルな手段になっている。

この3点をまとめると，ソフトウェア産業の生産性を向上させる労務管理は全体経済の生産性上昇にとっても倍加した効果をもたらすが，そのなかで教育・訓練が中心的な役割を果たすということになる。しかし，生産性向上は資本蓄積の源泉ではあるが，かならずしもシステム・エンジニアやプログラマーの労働条件の向上を意味するわけではない。本報告の目的は，この矛盾的な社会関係のいくつかの具体的局面と，矛盾の暫時的解決を表現する制度的枠組みを紹介することである。第2節では，ソフトウェア業の労働過程と従業員のあいだの分業を論じる。第3節では，この産業の賃労働関係の三つの類型を紹介する。それらはこの産業についておこなわれたいくつかのケース・スタディ[4]に基づいている。第4節では，賃労働関係の総合的な特徴を説明する。第5節では結論として，今後の人事管理政策の方向を概観する。

II　労働過程と分業

ソフトウェアの開発は次のような段階にわけられる：1）コンサルテーション　注文会社と相談しながら情報処理をおこなう業務を決定し，それに適したコンピューターを決定する。2）システム分析　情報処理の対象となる業務のながれを分析する。3）システムの概念設計　業務のながれを理解し，そのながれが最適のものであるかどうか確認したうえで，コンピューター処理ができるように業務手順の設計をおこなう。4）システムの詳細設計　システムをいくつかのユニットに分け，個々のユニットごとに業務手順を細かく分析してシステムに組み込む。5）プログラムの設計　詳細設計されたシステムをコンピューターにもわかるように書き直す。これは使用されるコンピューターの操作のルールに関係している。6）コーディング　コンピューターの言語を使用してシステムを記述する。7）テスト　業務の情報処理が実際に注文どおりにできるかどうかを，最初にユニット・テスト，次にいくつかのユニットをまとめたテスト，最後にシステムのテストにより確認する。

以上述べた労働過程の段階は，作業の標準化

2）プログラムの部分的再利用について，Cusumano（1991）参照。

3）生産性の向上のためのエンジニアの知識と技能の主な役割について，北谷利之（1990）参照。

4）東京大学社会科学研究所（1989），雇用職業総合研究所（1987），今野・佐藤（1990），北谷（1990），筆者がA社（東京）で行なった聞き取り調査（1990）。

（合理化）を可能にする程度と従業員に求められる能力とによって2つのグループに分けられる。

段階1から段階4までは上流工程である。ここで求められているものは，a） ソフトウェアとハードウェアにかんする技術的知識，b） システムの分析および設計の知識，c） 情報処理の対象となる業務のながれの知識，である。とくに最後の知識については，ユーザーと関係を緊密に保つことが重要である。生産されるソフトウェアを使用する会社の部門の多様さだけでなく，同一の部門でも会社ごとに作業編成の仕方（たとえば在庫管理の仕方や賃金構成）は異なっている。そのため，ユーザーと何度も会って業務が実際に遂行される具体的なながれを把握することが必要となる。これはコンサルテーション段階の役割であるが，システムを分析するさいに何度も繰り返す必要がでてくる。ソフトウェア・エンジニアの側に理解不足がありうるだけでなく，使用者の側が教えた手順を変更することもあるだろう。

他方，下流工程に含まれるのは，詳細分析，プログラミング，コーディング，ユニット・テストである。ここで働く従業員に求められるのは，上流でつくられたシステムを理解すること，そして，それをコンピューターが読める言語に翻訳することである。ここでの生産性の上昇にとっては，同種の課題，同一の言語での労働経験が重要である[5]。

上流工程と下流工程は異なったタイプの労働者によっておこなわれている[6]。上流で働くのはシステム・エンジニアであり，下流で働くのはプログラマーである。従業員のこの2グループは，教育を受けた水準とは必ずしも結びついていない。ソフト会社は，実際には，労働力不足のためもあり，まったくの未経験者を雇いいれて初歩から訓練をおこなうこともしている[7]。

システム・エンジニアとプログラマーの労働組織には，次の二つの形態がある。a） 大部分の会社では，プロジェクト・チーム制度がとられている。（東京大学社会科学研究所調査では71.5％）これは特定のソフトウェアの開発をはじめから終わりまで担当するために組織され，任務が終われば解散するチームであり，そのなかでシステム・エンジニアとプログラマーは共存する。b） 大企業で優勢なのは前者とはことなる組織形態で，上流工程と下流工程が別個の部（あるいは課）になっている。（同前調査によれば24.5％の会社）システム・エンジニアとプログラマーは別々の部課で働くが，それぞれの部課内部ではまたプロジェクト・チームが組まれるのが普通である。

以下ではこの産業の賃労働関係を3つのタイプにわけて論じるが，それらのタイプ間の相違には，上述のような労働組織のほかに，企業規模，賃金制度，従業員の階層づけ，従業員の学歴，訓練制度，従業員のキャリア，といった要素が関係している。

Ⅲ　賃労働関係の3タイプ

《タイプ1》

最初にシステム・エンジニアとプログラマーが同一プロジェクト・チームに共存するタイプをとりあげよう。これはソフトウェア業ではもっともよくみられるタイプであるが，とくに上流も下流も担当する小規模会社ではほとんどがそうである。（表3）

深刻な労働力不足を示す労働市場のもとでは，これら小規模会社は大学の卒業者を雇うことができない[8]。大卒者は良好な労働条件を提供できる大企業にいくからである。小企業は高校および専門学校の卒業者を雇用し，はじめはプログラマーとして働かせる。

5） システム・エンジニアとプログラマーに求められている知識・技能について，雇用職業総合研究所（1987）第一部参照。

6） 上流と下流の分業の歴史について山ノ内（1989）第三節参照。

7） リクルート社発行の技術社向け求人雑誌「Tech-Bing」1991年の各号でのソフトウェア関連企業の募集要項をみよ。

8） タイプ1の典型であるA社での調査。

表2　種類別ソフトウェア売上高

（単位：百万円）

	専用ソフトウェア	ソフトウェアプロダクト
1989	2,159,326（85.9%）	353,209（14.1%）
1990	2,905,840（84.0%）	552,107（16.0%）

資料：同表1
専用ソウトウェア：民間企業か政府機関のための注文生産
ソフトウェアプロダクト：一般市場に向けての大量生産

表3　プログラマーとシステムエンジニアの労働組織

（単位：%）

	プロジェクトチーム	部あるいは課
合計	71.5	24.5
従業員数によって		
9以下	56.8	22.1
10−29	79.0	19.4
30−49	78.8	20.4
50−99	76.3	22.3
100以上	58.2	38.8

資料：東京大学社会科学研究所（1989）

表4　規模による会社設立年

（単位：%）

従業員数	合計	1969以前	1970−1974	1975−1979	1980以降
9以下	100	1.5	4.5	14.2	79.1
10−29	100	4.3	8.8	22.9	64.0
30−49	100	4.7	16.8	25.3	52.0
50−99	100	8.5	27.1	28.7	35.1
100以上	100	26.2	30.7	17.8	25.3

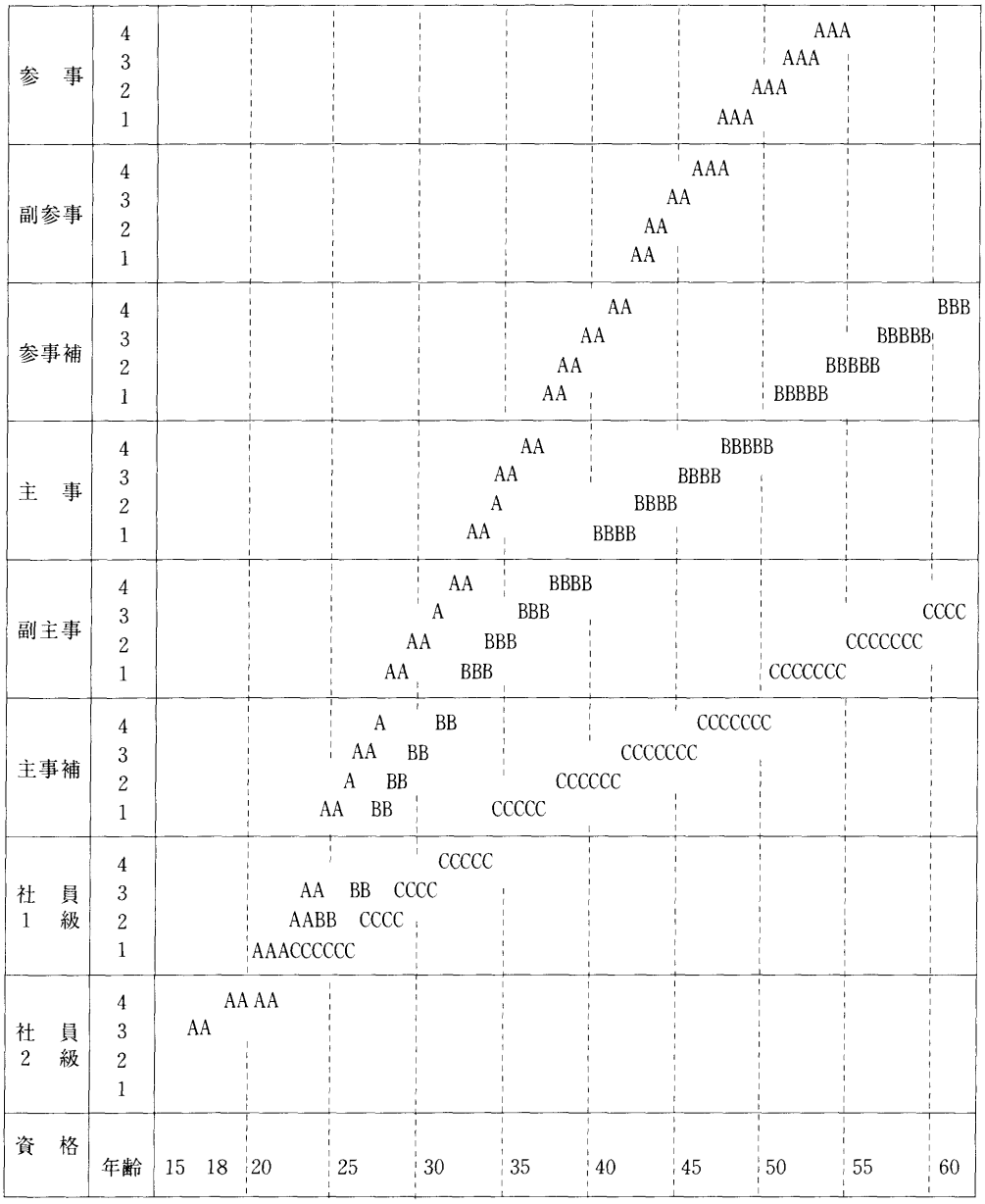
資料：表3と同様

この産業の会社の大部分は小規模会社であり，その80%以上が石油危機のあとに設立されている。これらの新しい小会社では，創業者（会社設立時のメンバー）自身が，経営だけでなく，システム分析やユーザーとの折衝などの上流工程を担当することが多い。

新規に雇用された従業員は下流工程で働くが，将来システム・エンジニアになるチャンスは原則的には誰にも開かれている。経営者は，彼／彼女らのうちの一部（10〜20%）が数年後にはシステム・エンジニアの仕事ができるようになることを期待している。会社の成長はシステム・エンジニアの数によって制約されているが，これらの会社は外部労働市場からシステム・エンジニアを獲得できないのである。

賃金は基本的に年功によって決められているが，勤続年数が多くなると評価（査定）の影響が大きくなるが，これは会社によって差異がある。10年くらいの勤続年数になり賃金格差が広がると，職階と賃金の逆転がおこるケースもでてくる。（表5）その頃には誰がシステム・エンジニアあるいは課長になることができ，誰が退職（実際は35−40歳ころ）までプログラマーにとどまるのかはほとんど決まっている。入社したばかりの従業員のあいだに，性別をのぞいては，差別はみられず，男子従業員は機会均等の社内労働市場で競争する。これはいわゆる日本的経営のモデルに近い。昇進によって得られ

表5　昇格モデル



＊年齢は4月1日を基準とする。
＊記号A，B，Cは，上期・下期のランクを示す。
　A……最も昇格の早い社員
　B……標準的な社員
　C……最も昇格の遅い社員

る職位は将来のキャリア発展の機会を約束してはいるが，実際の仕事の内容には必ずしも関連していない。しかし，職位は賃金水準とは密接に関連している。

しかしながら，このようなタイプの管理は様々な困難に直面している。新規学卒者が不足しているため，このタイプの管理を採用している会社では，新しく雇用される人員の半分近くが他の業種からの転職者で占められている。日本的経営における労使妥協の一つは，従業員のキャリアの諸段階において本人と妻子の生活を保障することであり，賃金の決定にあたっては勤続年数だけでなく，年齢と家族状況を考慮にいれなければならない。その結果，学校を卒業してすぐ入社する（18－20歳）従業員の賃金は絶対的にも相対的にも低い。彼らはプログラマーの経験を積むにしたがって生産性を上げるが，賃金の上昇はそれほどではないので，22歳から35歳までの期間は，生産性と賃金は会社の利益にとってもっとも望ましい比率になる。しかし，経験をもたない比較的高年齢の人を採用する場合には，低い生産性にもかかわらず高い賃金を支払わねばならない。特に仕事の習得期間中はそうである。こうしたケースの解決策は，彼らに一定年齢の給与のうちの最低ランクを適用することである。（表5参照）給与査定を加味した賃金の柔軟性がこの場合，外部労働市場の影響から日本的経営（内部労働市場）を守る手段となっているのである。

いま一つの問題は，従業員のキャリアである。プログラマーの大部分は，退職するまで上流工程の仕事をする経験をもたずに，ずっとプログラマーの仕事をしている。人手不足のためプログラマーは仕事におわれ，システム・エンジニアになるために，Off－JT を含む特別な訓練を受ける時間がない。先に述べたように，上流と下流で求められる能力が異なるので，ソフトウェアとハードウェアの技術的訓練だけでは不十分で，ユーザーとの関係のもち方も含んで会社の内部と外部の双方での Off－JT が必要になる[9]。こうした訓練コースへの参加は費用が高くつくという理由で，プログラマーをそれに参加させないとすれば，それはプログラマーのキャリアにとってきわめて不利にはたらくであろう。退職までプログラマーの仕事だけしか担当しないならば，係長レベルまでいけない工場労働者と変わらない。

実はこれは経営者も苦しんでいる問題である。というのは，このタイプの会社は外部労働市場でシステム・エンジニアを獲得することができないので，会社の中期的な成長（資本蓄積）は，会社内部での人材育成に依存しているからである。しかし，先の問題とも関連することだが，良好な労働条件を提供できないため，このタイプの会社に入社する従業員は，その教育・能力・意志・目的意識などの点で，システム・エンジニアに育てにくい人が多い。

現在のところソフトウェア業の小企業はまだ若く，こうした問題に深刻に悩んでいるわけではない。とくに設立時のメンバーが上流工程を自ら担当できる場合はそうである。それができなくなったさいにしばしば採用される解決法は，仕事の一部を上流だけ担当している，より小規模な企業に下請けに出すことである。こうした零細企業は，多くの場合，大企業の情報処理部に勤めていたシステム・エンジニアによって設立されている。

より基本的な解決は，好成績をあげて職位の進級の早いプログラマーに，20代後半の頃に特別な訓練の機会を与えることである。もしこの訓練がプロジェクト・リーダーによって行なわれるとすれば，そのコストは極小化されるであろう。プログラマーの技能の社内評価の最近の書式では，システム設計の能力の評価を含むものもある。どの段階のプログラマーが選抜されて訓練に参加させられるかは，会社ごとに違いがある。

下請け，および，プログラマーの選抜・訓練は，しかしながら，経営にとっては問題解決になるにせよ，プログラマーにとってはそうでは

9）システム・エンジニアとプログラマーの教育・訓練について，雇用職業総合研究所(1987)，23～29ページ参照。

ない。なお大部分のプログラマーは，これまではほぼ40歳前後であった退職時まで同じ仕事を続けなければならないからである。

《タイプ２》

ここでは，プログラマーとシステム・エンジニアが同じプロジェクト・チームのなかで働くが，会社が大企業である場合について論じる。情報サービス産業では従業員100人以上の会社は会社総数の13％であり，また500人以上は1.3％にすぎない。ソフトウェア開発会社だけをとれば，その平均規模は情報サービス産業全体のそれよりなお小さくなる。ここでは，従業員100人以上の会社は大企業と考えてよい。

もっとも大きな相違点は，大企業は大学卒業者を雇用できることである。大学卒業者は，はじめからシステム・エンジニアあるいは事務管理職になるためのトラックに入る。高校および専門学校卒業者はプログラマーのキャリアにとどまり，彼らの最高の職位は一般的にいってプログラマーの監督者である。しかし，大変少ないが，小規模なプロジェクトのリーダーになるものもいないわけではない。

大企業では，小企業に比べて，キャリアの分離はより明確でまたその開始段階も早い[10]。将来システム・エンジニアになる従業員も下流の仕事の仕方を知る必要があるので，入社した時には，皆プログラマーの仕事をするが，彼らはその後，システム・エンジニアとなるために必要な技能を，主としてOff－JTによる訓練によって身につける。

システム・エンジニアには，管理職にいたるコースと技術の専門化をはかるコースの二つのキャリア・パスがある。管理職と専門職の職位体系は異なるものの，対応しあう職位についての相違はその名称だけである。どちらのコースを選ぼうとも（選ばされるとしても），職位の昇格においても賃金においても大きな差異が生まれないようになっている。

専門職制度は，システム・エンジニアが技術的知識を深めるために設けられたとされるが，実際には管理職ポストの不足のために管理職になれないシステム・エンジニアが多いことと関連している。この不足は，経営組織のフラット化が進行したためでもある。また，40代はシステム・エンジニアの能力が陳腐化する年代といわれるが，事実，この年代のシステム・エンジニアがOff－JTの対象者に選ばれることはあまり多くない。専門職の職位の高さは賃金の高さに連動するが，これは，勤続年数と過去の累積的評価の結果であって，実際の仕事との関係は薄い。再訓練がもはや収益をもたらさなくなった従業員については，経営者は他の部門や関連会社に出すことを検討している。高年齢のシステム・エンジニアの賃金と生産性の比率は，会社の収益性にとって負担となるからである。

管理職の場合には，フラット組織が普及している。たとえば，小さな部をとると，部長と関連プロジェクトのリーダー，そしてプロジェクト・メンバーがいるが，プロジェクトのリーダーとメンバーには変化があるので，永続的な管理職ポストは１つだけである。より大きな部の場合には，副部長がいるが，次長，課長，係長といった中間管理職のポストは一般に廃止されている。大きなプロジェクトであれば，その中にいくつかのグループがあり，それぞれごとにグループ・リーダーがいる。フラット組織の普及とともに，決定権の部長への集中が進んでいる。

《タイプ３》

大企業によって採用される第二の組織形態は，生産過程の上流に分けられた二つの部課からなるものである。この場合には，プログラマーとシステム・エンジニアは別の部（あるいは課）で働く。二つの部課のあいだで人員の移動がおこりえないわけではないが，このタイプは，〈タイプ１，２〉においてもすでにあらわれていた労働力の二極分解を制度化するものである。プログラマーのキャリアは，下流を担当してい

10）大企業のキャリアの分離について，今野・佐藤（1990）6章2節参照。

る部課の内部にとどまる。プログラマーのキャリアは明確で，〈タイプ1，2〉のように実現しない期待で悩まされることは少ないであろう。

この労働組織はソフトウェア産業における技術革新の傾向を示している。プログラミングの自動化と合理化が進行し，プログラマーに求められる能力は減退する傾向がある。他方，システム・エンジニアに求められる能力は，プログラムされる業務手順の複雑化と，コンピューターの技術革新の結果，より高度なものになっている。システム・エンジニアが自分の仕事のやり方を再検討することは重要な意味をもつ。とくにプログラムの部分的な再利用は，生産性を上昇させるための主な方法であり，それをどれほど徹底できるかはシステム・エンジニアの能力にかかっている。こうした発展傾向に対応して，大企業のなかで選択される傾向が見られるのがシステム・エンジニアリングとプログラミングを別個の部課にわける組織形態なのである。システム・エンジニアとプログラマーの間の分業は構想と実行の分離に等しくなり，賃労働関係はネオ・テーラー主義に近づいている。

Ⅳ　技術革新と賃労働関係

労働過程の上流と下流での要求される能力の差異の増大は，労働力の二極分化を正当化するであろうか？　いいかえれば，技術革新と分業（労働組織）の発展方向と労働者のキャリア・パターンの間には一義的な関係があるだろうか？　一見したところでは，技術革新を独立変数として，そこから労働者に要求される能力の差異の拡大とそれに対応した労働者の二極分化を結果として導くことができるかのようである。しかしこの産業においても，労働者間の分業とその個人的キャリアは，労働組織，ローテーション，訓練の仕方，ポストの階等などに依存している。第Ⅰ節でみたように，労働力を現実の労働にかえることが労務管理の基本目的である。この目的は労使双方の利害をふくんでいるので，技術システムの範囲を超えて，経済的・社会的・政治的次元をももっている。技術システムの定着とその革新も，こうした多層にわたる労使関係の問題の解決に依存している。様々な解決が可能であるが，そのうち実現された解決は労使妥協とよぶことができる。日本的経営はその一つである。ここでは大企業の正規従業員に社内でのキャリアを保証し，そのみかえりに従業員は生産性向上のために協力している。生産の現場から流れる情報のフローが，全体としての労働者の日常の労働経験のなかから生まれる知識を利用した増分的な技術革新を可能にしているのである。

それではなぜ，ソフトウェア産業の大企業ではこうしたやり方が採用されないのだろうか？

プログラマーからの情報のフローによる生産向上の可能性はネグリジブルである。生産工程の上流では，生産性の向上はシステム・エンジニアを適切に訓練することで達成される。そのため，OJT と Off－JT をどのように実施するかが経営にとっての基本問題となる[11]。下流の工程では，経験を長く積むこと，とくに同一言語への習熟が作業の速度と生産性の増大をもたらす。したがってシステム・エンジニアにするためのプログラマーの訓練は生産性の減退をもたらしかねない。

社外の労働市場でシステム・エンジニアの採用が不可能であれば，経営者はプログラマーの訓練の費用を負担する可能性があり，現在のソフトウェア業では小企業はそうした状況にあることが多い。大企業ではシステム・エンジニアや大学卒業者を採用できるので，その結果，労働力の二極分化が激しくなるのである。このように，生産性上昇と技術革新，労働市場の状況，企業規模，システム・エンジニアとプログラマーのキャリア分化，これらは密接に関連しあっている。

従業員のキャリア・パスとそれに関連した訓練は，労使間の交渉の対象となりうる。ソフトウェア産業では労働条件の交渉はどのようにお

11）システム・エンジニアとプログラマーの雇用システム，キャリア・パスと彼らの教育・訓練との関係は，山ノ内（1989）第4節参照。

こなわれているだろうか？　労働組合がある会社は会社総数の５％にすぎない。（大企業ではその割合はもっと高い。）しかし，60％の会社には，社員会，親睦会などの従業員組織があり，労働条件にかかわる問題がある程度とりあげられる。（会社によっては形式的な話し合いがおこなわれる場合もある。）しかしこれらの組織の主要目的は，スポーツ，旅行など余暇時間を共同に利用するところにある。こうした組織（社員会）の81％（会社数）が，社長を含めた全従業員をメンバーにしている[12]。

労働組合および社員会の存在は従業員の労働条件に影響を及ぼしているだろうか？　東京大学社会科学研究所の調査の結果（回帰分析）では，労働時間と賃金についての影響は認められないが，Off－JT の機会については影響が認められている[13]。しかし，労働組合や社員会が存在する割合は，企業規模につれて高くなっているのであり，また，大企業ではたらくシステム・エンジニアが Off－JT を受ける頻度のもっとも高いグループであることも考慮しなければならない。大企業は訓練と福祉制度が小企業よりも良好なために，相対的に希少な大学卒業者を雇用できているのである。企業の規模と Off－JT の頻度の直接的な相関関係は，労働者の交渉力の成果ではなく，〈タイプ３〉でみたような労働力を二分化する方針の結果である。経営側はその剰余の一部を割いて，その技能向上が生産性向上の主要源泉となるシステム・エンジニアの訓練費用にあてる。他方，小規模な企業においては，システム・エンジニアは創立メンバーや経営者でもあることが多く，システム・エンジニアになるための訓練が受けられるプログラマーは一部にとどまっている。

以上みたように，労働組合や社員会の存在は労働条件に影響力をもっていない。したがって，従業員のキャリアは，経営側によって内部および外部の労働市場と関連づけられて設計されていると結論してよいだろう。このことは労働者の二分化がより普及することを必ずしも意味しない。二極分化の戦略は，プログラマーの動機づけの低下という問題に直面しているからである。日本的経営が長期間にわたる協力的な労使関係を生み出すことができるのは，社内労働市場と社外労働市場がはっきりと分かれ，従業員に会社の目的と自分の目的を結びつけた意識をもたせることができるからである。ところが，プログラマーのキャリアは魅力に欠けるので，社内と社外の労働市場の境界は弱くなる。事実，プログラマーの退職の割合はかなり高く[14]，その原因は，体力・能力の限界ではなく，将来の見通しの無さ，および，その他の労働条件についての不満である。

Ⅴ　プログラマーとシステム・エンジニアに対する今後の経営政策

先に紹介した３つのタイプのなかでは，〈タイプ１〉から〈タイプ２，３〉に進むにしたがって労働力の二極分化が増大する。とくに〈タイプ３〉の場合には，仕事内容の点でも要求される能力の点でも，プログラマーにとってのキャリアが存在していない。この場合には，今後の経営政策として次の２つの極端なケースを考えることができるだろう：1）　年功序列にわずかに評価を加味した職位と仕事内容・能力の分離を維持する。2）　プログラマーを非正規労働者にかえ，正規従業員に保証する年功的給与と福利厚生負担からのがれる。実際には，経営者はこの２つの極端な政策のあいだで適当な組合せを採用する可能性が大きいだろう。これからの傾向としては，非正規従業員として雇用されるプログラマーが増加することは確実であろう。

しかしこの政策にも，次のような限界がある：1）　プログラマーの供給の安定性を確保する必要がある。労働力不足状態の外部労働市場のもとでは，非正規労働者の増加は困難である。逆に，産業予備軍の増加はプログラマーの雇用

12）　東京大学社会科学研究所の調査（1989）による。
13）　同上。
14）　北谷（1990）11ページ参照。

契約の柔軟化を助けることになる。2） 会社によってその程度はさまざまであるが，日本的経営の維持は，労使関係の安定性のための基本的な枠組みになっている。といっても，日本的経営による安定したキャリアの享受者をシステム・エンジニアだけに限定する方向に徐々に進むことは不可能ではない。とくに，会社レベルおよび産業レベルでみるかぎり，こうした傾向に対抗できる制度がないからである。

次に，労働者の二極分化を避けてプログラマーとシステム・エンジニアが一つのキャリアに統合される場合を考えるために，システム・エンジニアがどの程度社内労働市場で育っているかをみてみよう。〈タイプ1〉で論じた小企業は，こうした統合されたキャリアにもっとも近いタイプである。しかし実際には，経営者はプログラマーのせいぜい1割から2割が上流工程を担当できるシステム・エンジニアになることを期待しているにすぎない。労使のあいだに明文化された合意がないために，プログラマーは入社した時点では，彼らの昇進とキャリアの可能性がわからない。過去の事例を参考にしようとしても，これらの会社は設立後の年数が少ないので先輩の経験に頼ることもできない。

〈タイプ1〉の会社でも，労働力の二極分化の外部化への傾向をみてとることができる。労働過程の上流はより小さい規模の企業に委託されることがあるが，こうした零細企業が最近ふえている。製造業あるいはサービス業大企業の情報処理部門で数年はたらいたあと，独立したシステム・エンジニアが生産工程の上流の委託を受けていることが多い。〈タイプ1〉の企業は，システム・エンジニアを外部から採用することもできず，かといって短期に社内でシステム・エンジニアを育成することもできないからである。このように労働市場の発展と上流工程の委託の増加には密接な関係がある。経営側の戦略としては，上流工程の委託をいっそう進めるか，それとも，社内でのシステム・エンジニア育成策を強化するかの選択が問題になる。

結論を急げば，第一に，〈タイプ3〉に近い場合には，労働市場における厳しい労働力不足のために労働契約の柔軟化は回避されるが，プログラマーとシステム・エンジニアの二分化は継続する可能性が高い。なぜなら，昇進および給与水準と現実に遂行する労働内容の関連はある程度分離できるからである。第二に，〈タイプ1〉に近い場合には，労働市場の状況は，上流工程の委託増加の原因になっている。実際，親会社との資本関係をまったくもたずに設立された会社の割合は73％であり，とくに従業員9人以下の零細経営では90％にも達している[3]。

失業率の増加がおこれば，〈タイプ3〉のような企業での労働契約の柔軟化が容易になるだろう。また，〈タイプ1〉のような企業でシステム・エンジニアの雇用がそのために可能になれば，賃労働関係は〈タイプ2〉あるいは〈タイプ3〉の方向に移動する可能性がある。

一般的な労働力不足が緩和されれば，プログラマーの供給は十分になるであろうが，その場合でもシステム・エンジニアの不足は近い将来に克服しうる見込みはほとんどない。こうした状況は，〈タイプ3〉では（おそらく〈タイプ2〉でも）プログラマーの労働契約の柔軟化をもたらすであろう。〈タイプ1〉の場合には，プログラマーの供給が十分になれば，大部分がOff－JTである訓練にあてる時間を増やすことが可能になり，システム・エンジニアへのキャリアが容易になり，内部労働市場が強化される可能性が高い。しかし，上流委託の方向への発展も可能である。いずれにせよ，労働市場の状況に対応して経営側は，中長期における費用と利益の関係を基準として，賃労働にたいする戦略をたてているのである。

参考文献

今野浩一郎・佐藤博樹，ソフトウェア産業と経営，東洋経済新報社，東京，1990年3月。

上条史彦，情報処理，日経産業シリーズ，日本経済新聞社，東京，1987年7月。

剣持一巳，マイコン革命と労働の未来，第11章：ソフトウェア産業とソフトウェア労働

者，日本評論社，東京，1983年１月。
雇用職業総合研究所，情報処理技術者の能力開発とキャリア形成―ソフトウェア技術者を中心に―，東京，1987年３月。
情報サービス産業協会・編，情報サービス産業白書，東京，1992年４月。
辻　淳二，情報サービス産業界，産業界シリーズ，教育社新書，東京，1990年４月。
東京大学社会科学研究所，情報サービス産業の経営と労働，東京，1989年３月。
石井修二・奥林康司　編著，ＭＥ技術革新下の労働，中央経済社，東京，1989年11月。
―第５章　山ノ内敏隆，ＭＥ技術革新と教育訓練―情報処理技術者の事例を中心に―
労働調査協議会，労働調査，東京，1990年３月号。
―北谷利之，各種調査からみた情報サービス産業の課題―人材育成を中心として―
―八幡成実，情報サービス産業における人材の確保と育成
CUSUMANO, Michael A., Japan's software factories, Oxford University Press, New York－Oxford, 1991.