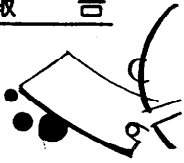


報告

パネル討論会



ソフトウェア技術者の養成†

パネリスト

有山正孝¹⁾, 三浦大亮²⁾
米田英一³⁾, 司会 高橋延匡⁴⁾

I. はじめに

ソフトウェア製品をどう作成し、どう管理してゆくかという問題は、ソフトウェアの設計、製作に携わる部門の共通の悩みであり、地道に、一步ずつでも前進させねばならない。本シンポジウムで「ソフトウェア技術者の養成」というパネル討論が計画された意図は、従来の工学分野にくらべてソフトウェア製品の開発では、製品の質が技術者の能力に依存する割合が高い、という認識に基づく。

しかしながら、日本の現状においては、ソフトウェア技術者が十分に育つ土壌にも問題があり、現実には「ソフトウェア製品」は主として若年労働者にその生産が委ねられている。一方管理者層は必ずしもソフトウェア製品の開発経験を持っているとは限らず、その面からも十分な育成プログラム(方針と計画)を持たない場合が多い。また、日本の経営風土が専門職よりも管理職指向が強いことから、技術者の長期的育成の面でも、各種の矛盾を内包している。

本パネル討論は、上記の共通の問題意識を基礎に、有山正孝(電気通信大学)、三浦大亮(東レ(株))、米田英一(日電東芝情報システム(株))、高橋延匡(東京農工大学)のパネリストによって行われた。なお、パネル討論では、時間の制限もあるので、事前に各パネリストの問題意識と各種の提案の論旨をシンポジウム予稿集にまとめた。本報告では、II章に、同上の論旨に加筆・訂正を加えたものをまとめた。なお、その論旨は、個性あふれるものであったため、できるだけ、加筆・訂正を少なくし、パネル討論の熱っぽい雰囲気をこわさぬようにした。また、パネル討論自身は、問題の本質を浮かび出させるため、歯に衣を着せずに本音の討

論に終始した。

第III章には、パネル討論での討論の内容をまとめてある。したがって、第II章の各パネリストの論旨を十分理解したうえで、第III章を読むものと仮定している。

ここで、パネリストの略歴を紹介しておく。

自己紹介



高橋 延匡(正会員)

1957年早稲田大学第一理工学部数学科卒業。理博。日立製作所中央研究所にて計算機システムとくにオペレーティングシステムの研究開発に従事。その間、HITAC 5020 モニタ(東大大型計算機センタ納)、HITAC 5020 TSS(世界初の仮想記憶方式のTSS)を開発。1977年4月、東京農工大学工学部数理情報工学科(教授)に着任後、引きつづき計算機システムを中心とした設計技法および方式論、OSの方式論、マンマシンシステム、日本語情報処理等を研究。



米田 英一(正会員)

昭和10年生。昭和34年東京大学数学科卒業。東京芝浦電気(株)へ入社。17年間、ユーザ部門において、数値解析を中心とする技術計算プログラミング、計算機室運営管理、全社的な計算機設備計画等の業務に従事した後、昭和51年第一電子計算機事業部へ移る。昭和53年4月日電東芝情報システム(株)へ出向、ソフトウェア技術本部次長。長年EDP部門にいたので、EDP部門の企業における地位、悩み、更にはSE/プログラマの知的レベルについては熟知しているつもり。その意味で、ソフトウェアの生産性向上も、有償化も、EDP部門に働く人々の地位の向上が鍵を握っていると感じている。また、高度に数学的な手法のみをもってよとする考え方には、個人的にはともかく、企業

† 日時 昭和54年12月13日、15:10~17:10

場所 日本都市センター

1) 電気通信大学、2) 東レ(株)、3) 日電東芝情報システム(株)
4) 東京農工大学

人としては賛成できない。

なお、55年1月より東芝電子計算機事業部副技師長。



有山 正孝 (正会員)

昭和4年生。昭和28年東京大学理学部物理学学科卒業。同大学院、工学部助手、理学部助手を経て、昭和36年より電気通信大学に勤務。現在同大学教授、計算機科学科に所属。初期には分子物理学を専攻し原子・分子の電子構造に関する理論的研究に従事、手回し計算器・微分解析機等から始めてその時代としては大型の数値計算を行い、計算機の重要性を痛感する。昭和33年頃より本格的に計算機について学ぶ機会を得て、以来主として計算機に関心を持つようになる。電通大においては30年代末よりささやかな計算機教育に従事してきたが、昭和45年電子計算機学科(現計算機科学科)が設置されてからはその整備・発展に専念している。現在の最大の関心事の一つはいわゆる情報処理教育のあるべ

き姿とその内容に関する事項である。本会情報処理専門教育研究委員会委員。



三浦 大亮 (正会員)

昭和9年生。32年に東京大学工学部応用物理学科を卒業。同学科森口研究室助手。34年に東レへ入社。現在に至る。コンピュータを積極的に使うようになったのは、31年から、ETL Mark II, HIPAC Iなどが最もなつかしい。その後、電子協に設置された国産試験機も次々と使わせてもらった。東レ入社後は、各種応用分野の開拓と、コンピュータ運用管理の諸業務を行ってきた。その間、USSC-90のALGOLコンパイラなどを作って楽しんだこともあるが、広範な実務に携わってきた。東レが日本語情報処理関連の機器を開発、製造・販売するようになって、50年から情報関連の新事業企画を担当している。

II. 問題の提起とその論旨

1. ソフトウェア技術者の教育について

高橋 延匡

1. はじめに

ソフトウェア技術者の養成は世界的な問題である。特に日本ではエネルギー資源の乏しい現状からして、情報産業、特に計算機に関連した情報産業は国家的見

大学教育	教育内容	入門コース	数学・物理学に基礎を置いた情報工学の専門教育	演習・実験(OJTに対応)	関連分野の技術・知識の習得(一般教養も含む)	その他
学部・学科を問わない 計算機の入門教育		FORTRANでプログラムが書ければ計算機を自由に使える気になさせる。(計算機のことにはわからない)				多人数教育 多人数の演習
情報工学科の学生教育			教育内容が多すぎる。多人数教育がしづらい分野である。カリキュラムのコンセンサスが難しい。	プログラムの添削が難しい。また、添削の労力がかかる。プログラム・リストだけでは不十分である。設備のアップデートが大変である。	問題意識の無い学生にはつまらない。	買取りの計算機を長期に亘り使用する。TSS用の良いシステムが国産機にはきわめて少ない。スタッフ、予算も少ない。

企業の教育	教育内容	新入社員教育	組織的教育	OJT	管理者教育自己啓発	その他
あるソフトウェアハウスの教育 (情報工学科の卒業生の比率は低い)		数週間から数か月の導入教育。固有のOSやシステムの教育。十分な教育ができない。		顧客へ外向き、そこで現実の問題を解くことが一番の教育になっている。	基本的には個人の問題であり、自分で勉強する以外にはない。ソフトウェア技術者こそ、社会とのインタフェースをあずかっており、関連分野の知識習得が大切にもかかわらず、自分の教に閉じこもり勝ちである。	古い世代は独力勉強組。組織的教育の必要性を感じていない。企業により差が大きすぎる。
一般企業での教育		導入している計算機メーカーのユーザ教育に依存する度合いが大きい。	その企業の主たる事業内容に関連する教育が中心。計算機のバージョンアップに追われる。	ジョブ・ローテーションがきかない。ソフトウェア技術者は視野が狭い。		特定メーカーの発想文化(言語、用語)から抜けられなくなる傾向にある。
ある計算機メーカーの教育		自社の計算機システムの学習が中心。	研究講座、技術講座等の社内教育中心。組織的教育へのおとがれと、忙しい人は受講できなくなる。	システム・プログラム(OS、コンパイラ)が中心となる。応用への関心がうすくなる傾向がある。		管理職への昇進がはやく、新人中心のソフトウェア開発になり勝ちである。

→ 大学卒3~5年 →

図-1 ソフトウェア技術者の各フェーズでの教育上の問題点

地からも将来の重要な産業のひとつに成長することが期待される。事実、現実の社会では計算機の利用はかなり進んでおり、国鉄の座席予約システムや銀行のオンライン・システムの一部など、世界のトップレベルのシステムを構築している。また、この分野に関連を持つ人口は増加の一途をたどっている。

ところが、このソフトウェアを作成するための設計技術（問題の定義づけ、問題の解析技術、設計仕様の作成技術、プログラミング技術、など）の科学技術上の水準は十分なものと言えない。さらに、通常の産業における設備投資に対応するものは「教育された人材」そのものであるからして、個人個人の能力向上、および全体としてのレベル・アップが重要な問題である。勿論、「企業は人なり」の格言がある通り、すべての業種に当てはまることではあるが、ソフトウェアに関しては、ほかの産業以上に「人」に依存する割合が高いにもかかわらず、教育の面では、むしろ既存の産業の方が熱心⁷⁾ではないかと思われる。

2. 問題の背景

ソフトウェア技術者の養成は如何にあるべきかという主題には数多くの側面が存在する。たとえば、

(1) ソフトウェア技術者のライフサイクルの問題

オペレータ、コーダ、プログラマ、SE という系列は本当に妥当か。再教育にはどうしたら良いか。

(2) メーカーにおけるソフトウェア技術者の養成と大学教育とのミス・マッチングの問題

- 情報工学科の教育は果して役に立っているのか。

- 大学の教育に何を期待しているのか。

(3) 大学の教育現場の問題

- カリキュラムはこれで良いのか。

- 演習、実験に手間と時間がかかるが良い解決策はあるか。

- 役に立つ（今すぐ役に立つ）ことよりは役に立たぬ（将来役に立つ）ことを教えるべきだとの意見も強いが、将来役に立つこととは何か。

(4) ソフトウェア・ハウスの長期的な経営問題

- 質の高さで商売できにくい環境にあるという説⁷⁾があるが本当か。

- 経営母体が弱く、教育とか研究に金と時間をかけられないところが多いというが将来どうするか。

学生諸君へ

— 大学で創造力を養おう —

大学時代とは、受験勉強という「大学へ入学すること」を目的とした勉強から解放されて、自己を確立する時期である。この時期をなによりゆきで過ごさず、目的意識を持って、創造力豊かな人間になろう。

(1) 大学とは、物事の考え方を学ぶところである。

社会に出ると、物事を考え自分で解決してゆく能力が不可欠である。正しい物の見方、考え方、問題解決のための判断力などの基本となる、自分なりの考え方、哲学を身につけることが大切である。

(2) 大学とは、学問の方法論を学ぶところである。

大学の教育と高等学校の教育で一番ちがうのは、大学の教育では「学問の方法論」を身につけることを目的としている。教えてもらうのではなく自分で学びとるところである。

(3) 大学とは、標準となる物差しを作ることを学ぶところである。

科学や工学の基礎は物をはかるところから始まる。はかるためには、物差しが必要であり、これは物理的な意味での物差しに限らない。研究対象に関する物差しを削り出し、対象をはかる眼力（測定能力）を養ってほしい。このために、特に基礎的、基本的な学力をきちんと身につけることが必要である。

(4) 大学とは、自分の考え、主張を的確に表現する能力を養うところである。

情報処理の分野では、自然言語（国語や英語）でその仕様（スペック）を記述する能力が重要である。すなわちその仕様が、第三者に誤りなく伝わるように、わかりやすく正確に記述されていることが、共同作業をしていく場合に、最も大切なことである。

これらは実験のレポートを記述することを通して学んでいくのが良い。

(5) 大学とは、人格形成の場である。

仕事は全人格の表現である。工学部であるから、社会へ出て工学的な設計にたずさわることが多からう。その時、設計者の全人格的な幅で、製品の良し悪しが決まってしまう。何事であれ、一芸に秀でた人の人生経験などを取り入れ、活かすように心がけてほしい。

以上の趣旨にそって、当学科の**数理情報工学実験**は創造力をいかにつけるかの観点から計画されている。すなわち、

(1) その人なりの発見の喜び、物を完成させるための泥くさい期間と完成の喜びを味わうこと。

(2) 楽をしては何もできないことを実践を通して知ること。

(3) 実験の目的は、「○○を学習する」ではなく、「○○を究明する、また○○を製作する」であると認識すること。

(4) 物を完成させなければ終りでないといわること。

(5) 人に理解してもらえるようなレポートが書けるよう積極的に工夫すること。

実験を通して、学生諸君は、大いに創造力を養ってほしい。

図-2 学生へのオリエンテーションの学科の考え方

- 下請け企業的経営体質のため良い人が集めにくい、どうしたらよいか。

(5) 日本固有の問題

- 縦社会、ジェネラリスト指向の処遇体系になっているが改革は可能か。

- 「知識」、「ノウハウ」に金を払う習慣を確立するにはどうすべきか。

3. ソフトウェア技術者の教育に関する各フェーズでの問題点

ソフトウェア技術者の教育を学校教育から企業内教育まで拡張して考えれば上記のような問題を指摘することができる(図-1, あくまでも例であり, 特定の所を意味しない).

4. 東京農工大学 数理情報工学科の例

ここでは, 情報工学系の学科の一例として, 筆者が所属している数理情報工学科を採り上げ, 教育の現状を紹介し, 情報工学科の教育の一実態を理解してもらうことを意図した. 勿論, 実態は, 学科の成り立ちの歴史的な背景(電気系, 物理系, 数学系など)によって, カリキュラムの内容や演習・実験の内容は千差万別^{5), 6)}であるので, 以下の例は, 逆に特異例かも知れないことを, お断りしておく.

東京農工大学工学部数理情報工学科は昭和51年4月に設立された講座数4, 学生定員1学年40人の学科である.

教授4名とも大学外で研究した経歴を持つことから, 学部教育は基本的な教育を重視し, 創造力のある人間を育てたいとの強い願望^{1), 2), 4)}を持ち, 教官を含め学科全体で精力的に教育を行うべく努力している.

図-2は新入学生のオリエンテーション資料の抜粋である. ここに学科の方針を明示している.

付属設備としては, 学科専用の計算機として, TSS方式のACOS-600を導入した. カードレス, ペーパーレスを志向し, 積極的に計算機を活用したいからで, バッチ処理重点のシステムは情報工学科の研究教育用には不向きであるとの判断³⁾による. このことは, 米国やヨーロッパの主なComputer Science学科を見ても自明である.

また, 特異な設備として, 計算機技術展示室⁹⁾があり, 100種類をこえる計算機械や回路部品が収集・展示・活用されている.

4.1 実験と演習の重要性

情報工学の分野の教育で大切なことは, 他人が使うシステムを設計できる能力を持つ技術者を養成すること, である. 技術を身につけるということは全身で覚

実験・演習テーマ一覧

		卒業論文		
4				
3	ソフトウェアの製作	<ul style="list-style-type: none"> ・シミュレータ ・数式処理 ・データベース 	<ul style="list-style-type: none"> ・デジタルタイマ ・IC検査器 ・プログラムパルス発生器 ・文字フォント表示装置 ・デジタル演算装置 ・任意波形発生装置 	まとまった装置の製作実験
3	現象の解析	<ul style="list-style-type: none"> ・簡単な言語処理システム ・計算機の性能評価 <ul style="list-style-type: none"> ・テクニカルライティング ・調査研究 ・KJ法とブレインストーミング ・ミニコンによるシステム制御 	<ul style="list-style-type: none"> ・電源回路 ・アナログ演算回路 ・D/A変換回路 ・磁気ヒステリシスの測定 ・検出器の応用 	簡単な製作実験
2	基礎的理解	<ul style="list-style-type: none"> ・数値計算における誤差 ・電位分布(実験と数値計算) ・整列(ソーティング: 手作業とプログラミング) ・ハッシュ法 ・スタック技法 ・共振回路の応答(実験とシミュレーション) ・確率統計現象 ・論理回路 ・手回し計算器による計算 ・EDSACの機械語 	<ul style="list-style-type: none"> ・波形の合成, 分解 ・トランジスタの応答時間 ・TTLの電気的特性 ・光演算 ・伝送線 ・直流サーボモータ ・機械振動 ・パルストランス ・マルチバイブレータ ・トランジスタの増幅作用 ・機械的カウンタ ・熱と半導体 ・スイッチングレギュレータ 	基礎的理解
2				
前				
		ソフトウェア系 ハードウェア系		

図-3 東京農工大学数理情報工学科の1980年度実験・演習テーマ

える必要がある. そのため実験と演習に重点を置いている(図-3参照).

当学科の特徴とするところは, 実験と演習に時間をかけているのは当然としても, ソフトウェア系の実験とハードウェア系の実験を2本柱とするが, 1つの実験で, 理論(解析解)とコンピュータによるシミュレーション, さらに物理実験としてハードウェア的にも納得させるという型のものをふやすべく努力している点をまずあげることができる.

全身で覚えるフィロソフィを実現するために, 「手回し計算器(タイガー計算器)による計算²⁾」なども桁落ちによる誤差を認識させるのに役立つと確信して

いる。この種類のものとして、整列（ソーティング）を手作業とプログラミングによって実施するなど含まれている。

プログラミングに関しては、2年の演習で、まず、EDSAC^{10)~13)}によるプログラミングを徹底的に行う。できるだけ簡単な計算機で基礎的な演習をすることは、計算機を用いて工学的に“計算機でできること”の直感を得るのに大切なことだと確信している。ただし、教育に便利だからと言って、この世の中に存在しないペーパーマシンでの教育は、現実の計算機の泥臭いところを覆い隠してしまい演習の精神や効果を半減する危険が存在する。ケンブリッジ大学の Wilkes 教授も同上のことを指摘¹⁰⁾している。当学科では、EDSACの TSS によるシミュレータ¹²⁾と、ミニコン的にTTLで組み上げたEDSA C 7620¹³⁾を用いて演習を行っている。

そのほか、3年前期に K-J 法とブレンストロミング、結果を2組でデバイトさせるなど、考え方をまとめさせる訓練を行っている。

最後に、レポートを非常に重視しており、学科専用の原稿用紙を採用した結果、文章の査読、修正も容易になり、学生の誤字脱字が少なくなっている。また、レポートの再提出を行わせ、いい加減なものは何回でも繰り返し再提出をさせるように心掛けている。

カリキュラムに関しては説明を省略するが図-4を参考にのせる。

5. おわりに

ソフトウェア技術者の育成は本当に難しいことではあるが、結論的に言えば、どのように育ちたいのかはまったく個人の問題であり、どんなシステムを導入しても解決できるものではない。したがって、ここでは、筆者等が現実に実行できる当学科の学生に対して実施していることを採りあげた。情性での教育はつづみたいと思っている。

最後に、当学科の教育方針に関しては、学科の教職員全員で長期に亘る討論を繰り返した結果⁴⁾であり、各位に感謝の意を表したい。

参考文献

- 1) 高橋延匡：大学で独想力を養おう，bit, Vol. 10, No. 6, pp. 666-669 (1978).
- 2) 西村恕彦：虎よ虎よ——数理情報工学科の教育——，数学セミナー(1979-8).
- 3) 中森真理雄他：情報工学科の計算機導入に関する一評価方式の提案，昭和 53 年度電子通信学会

数理情報工学のプロフェッショナルへ

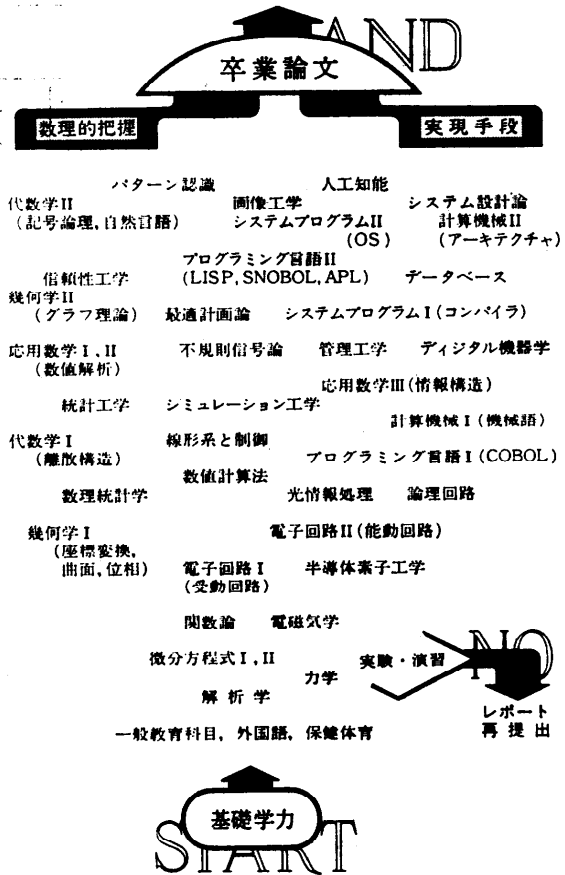


図-4 東京農工大学数理情報工学科の1979年度カリキュラム

総合全国大会論文集, 1386(1978).

- 4) 東京農工大学工学部数理情報工学科の紹介, 1979年度版.
- 5) 情報工学の学問体系のあり方に関する研究, 昭和 53 年度文部省科学研究費総合研究(A)・第二年度研究報告, 昭和 54-3.
- 6) CAI に関する調査研究報告書, 協同システム開発(株)(1980).
- 7) ソフトウェア技術移転に関する調査研究報告書, 財団法人 機械振興協会経済研究所, 協同システム開発(株)(1980-2).
- 8) 宮沢喜一郎：カリキュラムの調査 日米の情報科学教育, bit, Vol. 9, No. 5 (1972).
- 9) 西村恕彦：昔の計算機たち, 情報処理学会第21回プログラミング・シンポジウム報告集(1980).
- 10) Wilkes M. V. et al.: The Preparation of Programs for an Electronic Digital Computer,

Addison-wesley Company, INC.

- 11) 高橋延匡：EDSACによるプログラミング教育の方針，情報処理学会第21回全国大会論文集，pp. 1221-1222 (1980).
- 12) 清水敬子，高橋延匡：EDSACによるプログラミング教育の実際，情報処理学会第21回全国大会論文集，pp. 1223-1224 (1980).
- 13) 井上 賢，阿刀田央一：教育用EDSACハードウェア・シミュレータ，情報処理学会第21回全国大会論文集，pp. 1225-1226 (1980).
- 14) 中森真理雄，高田正之：数値情報工学科の教育における手回し計算器の活用，情報処理学会第21回全国大会論文集，pp. 1247-1248 (1980).
- 15) 野瀬 隆，鶴沢繁行：計算機械教育への1シミュレータの利用，情報処理学会第21回全国大会論文集，pp. 1251-1252 (1980).

2. ソフトウェア技術者の教育について

米田 英一

1. はじめに

与えられたテーマは「ソフトウェア技術者の高度化と再教育」であり，ソフトウェア製品を作る技術者の再教育が問題となっているようである。しかし，筆者はソフトウェア技術者の高度化のために最も重要なことは，ソフトウェア作りの仕事に従事している人々の社会的地位の向上であると考えているので，以下では企業の情報システム等，商品とはなりにくいソフトウェアを作る人々を対象として考える。また，再教育にはこだわらないことにする。

なお，ここに述べることは，あくまで筆者の個人的見解であり，NTIS ないしは東芝の公式見解とは全く無関係のものであることをお断りしておく。

2. ソフトウェア技術者の教育の必要性，目的，対象

そもそも，教育，少なくとも技術者教育，専門家教育には2つの側面がある。1つは社会のニーズに応えるという面である。“ソフトウェアからソフトウェア製品へ”という社会，時代の移り変りに伴って，高度のソフトウェア技術者，特に計算機メーカーやソフトウェアハウスにおいてソフトウェア製品を作るソフトウェア技術者，を育成し再教育せねばならないというのがこれである。もう一方の面は，その教育を受けた者が，将来幸福な人生を送り得るかという，教育を受ける側の視点から見た面である。もっと俗な言い方をすれば，偉くなれるか，社会で高く評価されるか？ということである。

この第二の観点から考えるとき，日本的風土の下で

はかなりの問題があることが分かる。特に，一般ユーザの EDP 部門の SE やプログラマを考えると，専門家を作ることが本当に必要だろうかという疑問も出て来る可能性がある。例えば次のような見方がある。

(A) 分散処理の時代，エンドユーザ中心の時代になれば，EDP 部門の仕事は少なくなり，高度の技術者は不要になるという(俗)説。

(B) EDP 部門は，一般企業の中では，所詮陽の当たらない部門である。一生 EDP 部門にいななければならないような「専門家」を作ることには教育界は加担すべきではない。現に EDP 業務に従事している人々の再教育は必要だが，新たに「不幸な」人々を作り出すべきではないという説。

(C) コンピュータサイエンスには詳しいだけで，社会科学，人文科学，自然科学，ほかの工学諸分野のことを知らない者は，EDP 部門の中では使いにくいという説。

これらの説がどこまで当てているかは別として，情報工学系に学んだ者が必ずしも計算機メーカーへゆきながらないという話もあるので，(B) や (C) の問題は避けて通れない問題である。

三浦大亮氏は情報処理に携わる人々を4つの職種に分類することを提唱しておられる(II, 4参照)。日本の一般企業の EDP 部門を考えた場合，三浦氏の考えがどこまで受入れられるか分からないが，技術者教育の対象は誰か？という点では，三浦氏の提案は甚だ明解である。すなわち，ここで考えている教育の対象はソフトウェアエンジニアとプロフェッショナルプログラマである。

3. 高度の教育とは何か

高度の教育というと，とかく数学的色彩を帯びたものが中心になり勝ちである。技術の進歩が激しい分野では，一生役に立つのは，Scientific Discipline, Mental Tool であることに異論はない。この意味で，いわゆるコンピュータサイエンスが情報工学系で教える学科の中心になることは当然であろう。

しかし，製品，実用品としてのソフトウェアを作る技術者を育成するためには，コンピュータサイエンスだけでは不十分であろう。情報工学科の目的は何人もの Dijkstra, Knuth, Wirth…… を作ることではあるまい。第一，大学に入って来る素材としての学生の質を考えると，こんなことは不可能であるし，一方，Dijkstra には 10^4 年近くに近しい工数をかけて作るバンキングシステムは到底作れまい。

ソフトウェア製品を作ることのむずかしさ、特にその開発プロジェクトチームのリーダーをつとめることのむずかしさ、は一方において高度の知的レベル、知識経験、ノウハウを要求されながら、他方においてチームメンバー全員を掌握し管理してゆくことを要求されるところにある。

人間の脳の右半球と左半球の役目に違いがあることはよく知られている。Mintzberg²¹は「プランニングは左半球で、マネジメントは右半球でそれぞれ行われる。一人の人間ではいずれかの半球が優位であるから、優秀な企画家であると同時にすぐれたデザインメーカーであることはむずかしい。……」と述べている。ソフトウェア製品の開発チームのリーダーには、まさにこの左右両半球の働きが要求されるところが辛いところである。角田忠信氏²²によれば、日本人の脳は欧米人のそれと異なるところがあるそうだが、左右両半球の役目に差があることには変りがない。伝統ある電気工学科、機械工学科、……など工学部、各学科の卒業生を見ると右半球優位の人も多いようである。情報工学系の学科に左半球優位の者ばかりが入って来ないような工夫が必要である。ソフトウェア技術者に要求される能力のうち、管理力、統率力といったものがかなりの程度まで天性のものであるとするならば、このような工夫、配慮は益々重要になる。また、これの能力がある程度訓練可能であるならば、「高度な教育」の中には、そうした実習、訓練も必要である。いずれにしても、情報工学系に入ってくる学生の性格についても、教える学科の内容についても、脳の左右両半球がバランスした形になるように期待したい。

4. 社会環境

ところで、管理力、統率力など組織のリーダーにとって不可欠の条件であるこれら能力に関して、今後の日本を考えるときわめて悲観的であるように思う。核家族化、ひとりっ子の数の増大、小学生時代からの塾通いと受験勉強や遊び場所の不足から来るガキ大将の不在、……という現在の（もっと正確に言えば、ここ十数年の）子供達のおかれた環境を考えると、リーダーの人間が将来不足することは避けられないように思う。

これに加えて、戦後日本の悪平等主義、「低きに合わせる」というものの考え方と、これに伴う能力の全般的低下、映像文化の時代における活字離れとこれに伴う文章力表現力の低下がある。大学生の全般的な学力低下については、先日新聞で大内力教授がなげいておられたが、文章力表現力の低下はソフトウェア技術

者の育成という見地からは、由々しき事態である。

マイコンの普及がこれに輪をかけるということも考えられる。なるほど、マイコンの普及により、コンピュータレギは少なくなり、一億総プログラマ論を唱える立場の人から見るならば、マイコンの普及は喜ばしいものであろう。筆者自身もマイコン、電卓の類はうまく使えば絶好の数学教育用ツールになり得ると考えている。しかしこれは両刃の剣である。しっかりした指導理念を伴わないまま、マイコンによる“toy program”作りの名人ばかりが増えることは、ソフトウェア製品を作る技術者の育成という立場からは、必ずしも歓迎すべき事柄ではない。戸川教授のお話では「コンピュータ関係の学科の学生は考えることよりもプログラムを作ることの方が好きな者が多い」ということであるが、マイコンの普及はこれに拍車をかけることになろう。

以上要するに、現代は脳の右半球、左半球いずれについても問題のある時代である。教育を云々する場合も、こういう認識のもとで考えねばならない。

5. 技術の確立、技術の確立を目指す体制の必要性

技術者に対する教育、再教育を高度化するためには、まず技術そのものを高度化せねばならない。これは当り前のことであるが、ソフトウェアの場合は色々問題が存在する。

一般の工業製品の場合には、これを作る（開発する、設計する）技術者の卵を教育するのは工学部の役目である。学生は、基礎的な理論、技術、物の考え方……を身につけて企業に就職し、それぞれの企業独自の 방법으로訓練されて一人前の技術者になってゆく。電機製品を例にとると、これを作る技術は、電気工学、機械工学から IE までの広汎な領域に亘っており、技術者の分業体制もしっかり確立している。大学にもこれらに対応する工学部各学科があるほか、二、三の大学には生産技術研究所も存在し、モノを作る技術が研究されている。また背後にある理論としては、電磁気学、力学、回路理論、自動制御理論……と確固たるものがある。

翻ってソフトウェア作りを考えると、この技術分野自体が若いこともあって、ハードウェアとは異なるところが多い。少なくとも、ハードウェアの場合のような、基礎理論—応用理論—実験—製造技術—生産技術……という全体的な流れに欠けている。情報工学系の学科は各大学にできて、ソフトウェアの生産技術研究所はどこにもない。どこかひとつの国立大学にソフ

トウェア生産技術研究所を設立するよう、文部省、あるいはお門違いかも知れないが通産省に対して働きかけることは、ソフトウェア技術に携わる者、特にその中でも大学人の責任ではなからうか。

もちろん、トヨタ自動車(株)のカンパシシステムの例にも見られる如く、真に効果的な生産技術は民間企業から生れるものであろうから、大学にのみ期待するのは間違っているが、日本の大学のアウトプットがあまりにも少ないことは否定できない。もしこれが制度に起因するものであるならば、その制度を改めさせるようなキャンペーンがまず必要であろう。

6. 大学教育とカリキュラム

ソフトウェア技術者に要求される、知識、経験、ノウハウのすべてを大学で教えることはできないから、大学教育で扱うべきテーマは自ずから絞られる。

大学進学率が40%を越えてしまった現代において、普通のアカデミズムが通用するか否かは大いに問題のあるところだが、大学というところは、物の考え方、Disciplineを身につけさせる場所であると思うので、教育もその方向でお願いしたい。ACMのカリキュラム68の改訂版カリキュラム78については、不勉強でよく見ていないが、数学的色彩が薄くなったという批判もあるようである。3節で述べたことと矛盾するようだが、数学的色彩が薄くなったことはやや問題のように思う。管理力、統率力という要素を重視することは必ずしも数学的なものを重視することとは矛盾しないはずである。

カリキュラムの内容そのものについて、これ以上云云することは筆者の能くするところではないので、ここでは普段から気になっていることを羅列することにした。

(1) 大学で本格的なモノ作りをさせるべきか否か
プログラム作りは、どの大学でも、プログラミング入門ということで、1人で作れるものを作ることになっているようだが、ミシガン大学でやったISDOSのようなソフトウェア製品を作ることもやらせてはよいのではなからうか。国立大学の場合、制度上の問題もあるようだが、どうせプログラム作りをやらせるなら、ここまで徹しないと本物ではないように思う。

(2) ミニチュアプログラム作りの害

プログラム作りには麻薬に似たところがある。特に若いときはそうである。プログラム作りへの過度の熱中は正しいことであるかどうか考える必要がある。少なくとも toy program を数多く作るのは問題であ

る。

(3) 技術文書の作成訓練

大学でぜひやってほしいのは、技術文書の書き方の訓練である。レポートの書き方、仕様書の書き方等、教師が学生の書いたものに朱を入れて添削指導してやる必要があると思う。高橋先生は農工大では実行されておられるとのことだが、各大学でこれが実行されれば、非常に有効である。

(4) ケーススタディの重視

アメリカのビジネススクールでは、ケーススタディをきわめて重要視しているようである。ソフトウェア技術の場合も、企画やシステム設計の段階については、ケーススタディが有効であると思う。特に失敗例が役に立つ。問題は材料であるが、アメリカあたりの例を探す努力が必要であろう。

(5) 企業人による講義

一部の大学の情報工学系学科では企業人による講義が行われているが、これはもっと行われてもよからう。更に、企業と大学の間の人の交流も、もっと行われてもよからう。筆者の私見では、我が国において、大規模のソフトウェア作りについて最も深い経験を有しているのは大銀行や大鉄鋼メーカーのEDP部門の中核にいる人である。こういう人々の有する経験やノウハウを大学が活用する道を考えてもよからう。

(6) 人文科学、社会科学を無視するべからず

文科系の人の中には数学や物理が分からないことを得々として話す人も多いが、逆に理科系の人で、人文科学や社会科学に無関心であることを公言する人もいる。コンピュータサイエンスの無色透明性もあって、情報工学系学科は、自然科学にすら無関心な人を作りかねない。コンピュータ、情報システムという。人間社会にとって両刃の剣になりかねないものを扱う職業に従事する者にとって、職業人としての倫理感をもっていることはきわめて重要である。そのためには、定式化可能なもの、計量化可能なもの以外に対しても関心を抱く人間を育てるのが教育者の責任である。

7. 再教育

ソフトウェア技術者の企業内再教育をどうするかは各企業それぞれの責任において考えるべきことであるので、ここでは企業外教育を考える。

(1) 大学

現在の大学には、メーカーやユーザのソフトウェア技術者が1~2年大学へ戻って再教育を受けるためのカリキュラムも用意されているとは思えないし、再入学

するだけの魅力にも欠けているように思う。MIT やスタンフォードに匹敵するだけの魅力を有する、旧人用コースの設立を強く希望するものである。

(2) 公的機関

情報処理開発協会 (JIPDEC) から発行されている「上級 SE 育成指針」は、メーカーやソフトウェアハウスのソフトウェア技術者ではなく、一般ユーザの EDP 部門の SE を対象としたものであるが、非常に盛況山のことを書かれている。この指針に沿って実際の教育を行うのが、JIPDEC の下部機構である情報処理研修センター (IIT) の役割であろうが、IIT の現状は必ずしも満足すべきものではない。何よりも受講者が少なすぎる。一般企業の EDP 部門の教育に対するニーズがないというよりも、むしろ、数カ月間の長期派遣に対する回収が不十分であると企業側が判断しているからであろう。

(3) 国

通産省の情報処理技術者試験は一般企業の EDP 部門の SE、プログラムを対象とするものであるが、受験者は年々増加の一途をたどっている。しかし、これは資格試験でも何でも無い。2 節で述べたような観点からは、この試験を資格試験にしてゆく必要がある。

一方、通産省は昭和 49 年以来、国産メーカーに補助金を出している。また、ソフトウェアハウスも JSD そのほかの形で補助金の恩恵に浴している。しかし、この補助金の一部を、思い切って、ソフトウェア技術者の再教育に強制的に使わせるという方法も考慮されてよい。例えば、補助金の一部で、各メーカーから数人ずつ MIT やスタンフォードへ留学させるというやり方である。速効性はないが、長い目で見たとき、開発費の一部を補助金で負担するという現行方式より効果があるかも知れない。

8. 国や大学に望むこと

すでに述べたことも合わせて、国や大学に望むことをまとめると次のようになる。

(A) ソフトウェア技術者の教育の問題は、一方において教育の問題であると同時に、他方において、職業の問題、技術振興の問題、更には産業の問題である。したがって、文部省、労働省、科学技術庁、通産省… という各省庁が一体となって、政策立案を行う必要がある。(これは何もソフトウェア技術者の場合だけの問題ではない。) 現実には、各省庁間の壁が厚いことは周知の事実である。また、高級官僚は席の暖まる暇もなくポストからポストへ移り歩くので、長期的政策

の立案は不可能というのが現実の姿である。それゆえ、我々情報処理の仕事に携わる者が、もっと国に対して働きかけを行う必要がある。

(B) 文部省だけを相手にすればよいレベルの教育制度の改善、例えば国立大学でソフトウェア製品を作ってもよいというような制度の導入、については大学の先生方のキャンペーンをぜひお願いしたい。

(C) ソフトウェア技術を高度化するためには、何といっても、優れた人材がソフトウェア作りの仕事に従事する必要がある。優れた人材を集めるには、まず、ソフトウェア作りという職業が世人から高く評価されるようにならねばならない。これでは卵とニワトリの関係になってしまうが、少なくとも、大学でコンピュータ関係に携わっておられる先生方が世人の尊敬を集めることが必要である。ごく一部の先生方に見られるようなメーカーいじめをしていたのでは、自分達の教え子達がメーカーに行きたがらなくなるというのも理の当然である。正価の 8 割引で計算機を購入したというような話も同断である。

(D) 日本の大学でも、もっとオリジナルな理論、オリジナルなソフトウェア製品 (またはその試作品) というアウトプットが出るようにしてほしい。偉大な学者が 1 人でも出れば、その分野に優秀な若人が集まることは、湯川博士のノーベル賞受賞後の素粒子論人口を見れば明らかである。日本の大学の奮起を祈りたい。

(E) 大学全体の計算センタ、あるいは教育センタには大型のコンピュータが入るようになって久しいが、研究用のコンピュータ、各種端末機器等をもっと充実させるべきであろう。「あの設備では日本の大学にソフトウェア技術は期待できない」といっている某大企業のシステム部門の部長もいることを、文部省あたりにもっと認識させる必要がある。

(F) メーカーのソフトウェア技術者や SE、あるいは一般ユーザの EDP 部門の SE などが 1~2 年位大学へ戻って勉強する気になるような、魅力的な「再教育」用コースの設定をぜひお願いしたい。大切なことは、このコースが本人、会社側双方にとって魅力的であることである。

9. おわりに

以上、普段考えていることを随筆風に書きつらねてみたが、再三繰り返すように、技術の高度化のためには、何よりもまず、よい人材をその技術分野に集めることが大切である。そのためには、これも再三繰り返

すように、ソフトウェア関連の職業が社会から高く評価されるようにすることが重要である。これなくしてソフトウェア技術の明日はないといってよい。大学の先生方、我々民間企業（メーカーおよびユーザ）に属する者が、この点について相携えて進んでゆきたいものである。

参考文献

- 1) Mintzberg, M.: Planning on the left side and managing on the right, Harvard Business Review, Vol. 54, No. 4, July-August, 1976, pp. 48-58.
- 2) 角田忠信：日本人の脳，大修館書店（1978）。

3. ソフトウェア技術者の教育

—大学の立場から—

有山 正孝

1. 大学は何をしてきたか、何をしているか

わが国の大学でコンピュータ教育が始められたのは1950年代なかばのことである。始めは自然発生的なもので、少数の教育にすぎなかったが、徐々に拡大して本格的にカリキュラムに取り入れられるようになった。1960年代の後半、本格的な情報化時代を迎えるに至って情報処理技術者に対する需要は急増し、educated manpowerの計画的な供給が必要となった。この時期に文部省に設置された「情報処理教育に関する会議」は1972年までに「情報処理教育振興の基本構想」と題する最終報告と、これに付随する一連の報告書を発表した。その中に盛られた提言はその後のわが国におけるコンピュータ教育のあり方に少なからず影響を与えてきた。

この時期から、大学における“情報処理教育”は、“一般的情報処理教育”と“専門的情報処理教育”の2本の柱に沿って展開される。前者はおおむねユーザ教育であり、後者は“専門家”の教育を目指すものである。

一般的情報処理教育は“学生一般に対し情報処理の可能性と限界を認識させ、情報化時代に人間が主体性をもって未来を切り開いていくことができるようにする”ためのものとされ、あらゆる学部・学科の学生を対象とし、本来は教養科目としての性格もあわせ持つべきものであるが、現実にはそれぞれの専門分野において不可欠の手法の1つとなったコンピュータ利用の入門教育が行われているものとみられる。したがって

その内容は、学部・学科によって相違もあるが、主に“プログラミング入門”——FORTRAN または COBOL、まれにはアセンブリ言語によるプログラミングの初歩にあわせて数値計算法の手ほどきもする——、“コンピュータ概論”——計算機のアーキテクチャ・システムプログラム・プログラミング言語等に関する概説、ブール代数・論理回路・デジタル技術の初歩など——というような内容が、独立の講義・演習として、あるいはほかの授業科目や実験の中に織りこまれて、広く浅く行われているのが実状のようである。

一方、専門的情報処理教育は大学・研究機関やメーカー等においてハードウェア・ソフトウェアの研究・開発・製作に携わる専門家を育成するためのものである。早くからその重要性を認めてこの分野の教育に力を注ぎ、優れた人材を送り出してきた学科も少なくないが、1970年以降は特に国立大学の理工系学部重点的に“情報関係”の専門学科および大学院が設置され、その数はそれぞれ25ないし30に達している。これらの学科・専攻の教育内容はそれぞれの成立の経緯や学内環境により相当の相違があるが、一方ではACMのCurriculum 68やIEEEのCOSINE委員会のカリキュラム勧告案等の強い影響を受けてきたように見受けられる。

ともかくこのようにして、コンピュータの急激な普及に伴う人的資源の需要に対応して大学が相当の人数を供給し、かつ広範囲にわたる潜在的エンド・ユーザの準備教育にも一定の成果を上げてきたことは評価されてよいだろう。

しかしその現状に問題がないとは言えない。特にここで関係があるのはソフトウェア技術者の“専門家”としての水準または固有の能力にかかわる問題である。

ソフトウェア技術者においてはこれまで“専門家”と“素人”の区別が判然としない状態、あるいはその差があっても認識・評価されない状態が続いてきたのではあるまいか。その原因の1つは、これまでは現実にFORTRANでプログラムを書ける程度の“素人”をソフトウェア技術者として投入せざるを得なかったこと、あるいは投入が可能であったことであろう。つまりそれで間に合わせることが可能であったわけで、言い換えれば“専門家”と“素人”との能力の差が平均的にそれまで大きくなかった、あるいは大きくないと思われてきたのである。

自動車は誰でも少し練習すれば運転できる。しかし

一級のレーサーになることや自動車を作ることは、誰にでも容易にできるわけではない。別の比喻を持ち出すなら、理工系の学生は誰でも微積分や線形代数を一通りは使いこなせるはずである。だからといって彼等を数学者と呼ぶだろうか？

この違いは、技術なり学問体系なりの深さにかかわる問題であろう。以前のソフトウェア技術は単純素朴なものであった。だから言語を知っていれば、そして多少の訓練をすれば、それで一応は間に合った。しかし今や事態は急速に変化しつつあることを認識すべきである。

誤解のないように断っておくが、“専門家”が排他的にソフトウェアの生産を独占することを奨励するつもりではない。素人が専門家をしのぐ良いソフトウェアを作ることはあり得るし、あっても差支えはない。ただソフトウェア技術の“専門家”があるとするれば、それは一般のユーザとは異なる責任を持つものであり、かつプロとして素人の追隨を許さぬ技術と能力を持つものでなくてはならないことを、また大学で専門的ソフトウェア技術者を養成していると言うからにはそれ相応の教育をせねばならぬことを、強調したいのである。これはまた、ソフトウェア製品の質を向上するためにも、必要なことの1つであろう。

もう1つの問題として再教育が挙げられる。言うまでもなくコンピュータの分野では技術の進歩が特に早いことや企業内での EDP 要員の人事管理上の問題等から、中堅技術者に対する再教育・継続教育が重要視されるようになってきている。これに対する大学側の対応としては、従来からある制度、すなわち研究生・委託生あるいは正規の大学院学生としての受け入れ、または公開講座という形式によるものであるが、これでは今後の需要——あればの話だが——をみたすには十分のように思われる。

2. 大学は何ができるか、何をなすべきか

大学に要求される社会的機能の1つは高度の専門家を育成することである。しかし大学はそのほかにも多くを要求され、また内部にも問題を抱えている。その中でここに関係のあるものを二、三あげておこう。

第一は、いわゆる大学の大量化に伴う学生の能力・志向の多様化である。

第二は、学問・技術の急速な進歩による専門知識の増大と、それを4年の年限の中で詰め込もうとするための教育過多・過密カリキュラムの問題である。

第三に、現在の大学教育ではいわゆる一般教育を重

要な柱の1つとする建前となっている。これは理念上は人間教育のはずであるが、現実にはその一部——理工系では数学・自然科学——が専門の基礎とみなされている。戦後の学制改革に伴って誕生したこの一般教育は、実はそれから30年を経た今日、なお成功を収めているとはいえ、邪魔物扱いさえされている。

さてこのような大学教育問題に敢えて触れたのは、それが大学のアウトプットの1つである卒業生に対する社会の要請と関係があるためである。ソフトウェア技術の分野に限定すれば、折に触れて我々の耳に入る企業側の要望には次の2つのタイプがある。その1つは、“大学では基礎をしっかりと養ってくればよい”、“あれもこれも教えるよりは人間教育をしっかりとしてほしい”というものである。ただしこれらは、自社の望むような教育を自分でする余裕と実力のある企業の声のようである。もう1つは、多くの知識・技能を持っていること、たとえばアセンブリ言語で大きなプログラムを書いたり、マイクロコンピュータを扱い、インタフェースを設計・製作できる技能を持ち、今日からでも役に立つ即戦力を求める声である。

この一見相反する要求と前記の諸問題の間に立って大学はどのような教育をすればよいか、2つの要求を同時に満足させることができればよいが、それはかなり難しい注文である。日進月歩の技術をいろいろ教え込もうとすれば学生の消化不良はますます重くなるであろうし、カリキュラムの面では基礎教育を一層圧迫することになる。そうして教え込んだ知識もおそらくたちまち陳腐化するであろう。大学の修業年限を延長することも一般論としては考慮に価するが、そのような大きな改革は早急に実現する可能性が小さい。結局、考えられることは取捨選択である。長い目で見れば、大学の学段階では最も基礎的なことを精選して教育し、基本的な訓練を徹底し、物の考え方・物事を見る目を養い、卒業後も学問・技術の進歩に追従できる能力、必要に応じて必要な手段は自ら見出し創り出せる能力を身につけさせることが大切であろう。しかしそのためにはまずソフトウェア技術にとって何が基礎であり基本であるかをはっきりさせる必要がある。ソフトウェア技術が単なる技法の集積、ノウハウの集積にとどまらず、一層標準化され体系化され、理論的に基礎づけされることは、良い技術者を育てるためにも必要である。

社会人の再教育に対しては大学はどのように寄与できるであろうか。これまでのところ、大学はこの種の

事業には不熱心であったし、社会の側にも強い、組織的な要求があったとは思えない。しかし近年再教育・生涯教育に対する関心がたかまるに及んで、この方向への努力も始められたようである。もっともこの場合にも、大学のできることは基礎的な面の教育であろう。

4. 品質管理とリーダー教育

三浦 大亮

一般ユーザやソフトウェアハウスの多くの若い技術者と接触してみて、その実状と彼等自身の反省・意見から、高度化と再教育のポイントとして、特に次の2つを指摘したい。

- 品質管理教育と品質管理システムの開発・定着化への動機づけ
- グループリーダーの育成とグループ管理技術の向上

個別の技術も重要である。これについては、技術開発や技術教育という形で、直接論じられよう。上記の2点を掲げたのは、ソフトウェアの生産がシステムとして効率的に働くために、重要な役割を果たすと信じられるからである。

1. ソフトウェア技術者

情報処理システムの開発に携わっている技術者全体を、情報処理技術者とおおまかに呼ぶことにする。この技術者は、次のように区分して理解し育成・管理すべきである。

A. 情報分析技術者 (Information Analyst)

企業活動に必要な情報のニーズと流れを分析し明確化する。

現在この種の活動ができる技術者はほとんど居ない。しかし今後の主役はこれが演ずることになる。

B. システム設計技術者 (System Designer)

情報のニーズと流れ(処理)を、利用可能なハード/ソフトを駆使してシステムとして最適な構成をする。

従来のいわゆるSEをまともなレベルにしたもの。現在までのSEは、技術的にも企業実務でも中途半端である。技術中心の専門職務スタッフとして育てていく必要がある。

C. ソフトウェア技術者 (Software Engineer)

ソフトウェア(プログラムの集合、実用に供するのに必要なすべての文書を含む)を、所要の基本機能を

適切に果たすことができるように、最適に設計する。またその生産の指導・推進をする。

処理のアルゴリズム、プロセスなどの開発・選択も適正に行える。ソフトウェア技術の向上が最も望まれるのはこの分野である。

D. プログラマ (Programmer)

ソフトウェアの生産工程であり、その実施運用者として、基礎的な役割を果たす。

安定的に一定以上の品質のプログラムが生産できるように育成・管理しなければならない。ソフトウェアの基本的な設計は、ソフトウェア技術者によって行われる。

A, B, Cは、上下関係にあるのではなく、情報処理システム開発の3本の柱として鼎立する機能と見なすべきであろう。DはB, Cの下位である。

それぞれの技術者が十分に認知され、相当の地位を確保するためには、素人と画然と区分される技術的实力を持っていないといけない。またその技術の有用性が何らかの方法で証明されていることが必要である。

そして、これを保証する1つの手段として、技術者の国家的な資格制度が施行されることを望む。現行の情報処理技術者認定試験制度も、ある程度の意義を持っているが、弁護士、医師、公認会計士などのような実施資格の有無を決めるようなものでなければならない。

さて、本論の主テーマである高度化と再教育の対象となる技術者はどの辺か。現在最も有効な対象は、およそ次の条件にあてはまる層である。先の区分では、DのトップクラスをCに育てることである。

① ソフトウェア開発を主業務とする数年以上の経験がある。

② 今後、直接的な指導監督なしでも、相当のソフトウェア開発ができるようになることが期待されている。

③ 今後、数人のプログラムのグループリーダーとして、相当の成果をあげることが期待されている。

④ 本人も今後、ソフトウェア技術者に成長し、その職務を継続する意欲がある。

⑤ 周囲から見て、そこそこの適性があると認められる。

俗に言い伝えられているプログラマ30歳停年説は、この②～⑤のどれかの条件で著しく欠けている者にあてはまるのだろう。

このような対象は、まあまあ製品となるソフトウェアを作るのに、最も重要なポジションを占めている。また、年齢としては 20 代後半から 30 歳くらいまでで、教育効率も良い。先の区分 A, B および C の高レベルの技術者は絶対的に不足しているが、そう簡単には教育できないし、またそのレベルに近い人達は自ら研鑽することができるだろう。ただ、中間管理者層、経営者層の教育は、依然として課題として残ろう。

2. 品質管理 (QC) について

ここで既成の品質管理手法を、ソフトウェア生産の現場に直接適用することを意図しているわけではない。先進的コンピュータメカでは、もう長い間ソフトウェア生産に QC を利用しているといわれている。しかし、それがハードウェア生産の場合のように成功したという話は聞いていない。

ハードウェア生産で世界に冠たる日本の QC だが、ソフトウェア生産に適した QC に未だ育っていないからであろう。おそらく、個々のメカの特定の組織によって推進される閉鎖的な QC では、良い結果は得られないだろう。日本の QC の普及期から行われている開放的かつ業際的な研鑽を経るべきである。そのための実際のアプローチの一つとしても、以下の主張をしたい。

(1) QC の必要性

品質によるフィードバックの無いシステムは、生産システムではない。またこのような所で使用される技術は、結局は独善となり適切な進歩はなく、形骸化することになる。

SE になろうという中堅の技術者が、「生産性を向上させるため、SP を採用し Walk Through の実施をした。その結果は良好だった。」というような論文を書いていることがよくある。よくよくその話を聞いてみると、何がどのように良好であったのか、少しも定量的な裏づけが無いことが分かる。

このような論文が出てくるところに居る SE またはその卵達(彼等の指導者や管理者もだろうが)は、おそらくモノを生産するとは何か、を知らないのだろう。一つの側面として、彼等の生産現場にはフィードバックが無いのではないか。そういう所には創意も工夫も動機も生じない。フィードバックが情報処理システム開発部門ではひどく欠けていること、またそれが技術者達の大きな不満の一つであることは、米国の行動科学者達の調査研究報告でも明らかにされている^{9),10)}。

第二には、品質意識の無い所で良い製品が作られる

はずがないということである。良い製品とはどういうものか、明確でなければいけない。そして、それが測定可能である必要がある。

良いプログラムとか、優れたソフトウェアとかの一般論は、当面無用である。多様なソフトウェアには、それぞれの品質目標やその評価基準が決められるべきである。これを科学的に決めていく努力と工夫を、実務的にすすめていかなければいけない。

(2) QC 教育のポイント

ソフトウェア生産に役立つ QC は未だ存在していない。QC 教育の主眼は、その基礎作りに役立つということを、まず目指さなければならない。彼等とともにソフトウェア向きの QC を現実の場に確立すること。QC と言って、すぐに多数のデータを採り、沢山の計算をさせ、平均値や標準偏差による管理図を書くことなどから始めるのは不適當である。

品質とは何か、ということを実践的に検討し明らかにすることがまず大切だろう。一般には、① 製品の機能、② コスト、③ 納期、④ メンテナンスコストである。これらを総合して、客のニーズや経営のポリシーによって、品質目標が決められる。現状では、このことが不足しすぎている。

管理については、QC のバックボーンになっている Plan-Do-Check-Action の管理サークルの現実化が重要。

次に重要なのは、自己啓発のための基本的動機づけである。QC も管理者や一部の管理スタッフの道具となったとき、発展はしなくなる。QC が日本的経営風土の中でうまく培養され、世界をリードするまでになったのは、現場の担当者およびグループ自身の技術として定着したことが原因である。これがまた自己発展的に成果を積み上げていく大きな力となっている。統計の方法などの基礎的技法を植えつくと同時に、実施運営の方法も、QC サークル運動に学ぶべきである。

管理を外からしなくても、レベル以上の生産を安定的にしてくれるソフトウェア技術者あるいはそのグループを育てるのがポイントである。当初からソフトの QC は完成しないが、このような基礎作りの過程で、新しい QC が生まれると信ずる。

3. グループリーダーについて

ほとんどのソフトウェアは、一人だけで完成できる状況ではなくなった。そこですぐに登場してくるのが、プロジェクトチームというものである。管理方法もリ

一ダシツプも無いチームは、鳥合の衆と言う。大きなプロジェクトのリーダーが仮に優れていたとしても、末端の細分化されたグループのまとめはできない。さらに、現在までのところ自然にグループやチームリーダーが育っていくような職場環境にない所が多い。もし良いリーダーが育ったとすれば、全くの偶然かその個人の能力がきわ立って良かったと言わざるを得ない。

特に問題なのは、人間のスキルや、意志疎通のコミュニケーションスキルについては、訓練されるよりは一層遠ざかるような傾向を助長するのが、プログラミング作業の大きな特質だ、という点である。

(1) 人間的側面

QC と同様、ソフトウェア工学なるものが単なる技法として存在しても、現場に浸透し有効に活用されているかという点には、大きな疑問がある。個人のソフトウェア生産性が正しく把握・評価されず、さらにそれに見合った処遇ができていない状況では、自然に技術向上や、新しい技術の積極的活用が行われるとは期待できない。どちらかと言えば、科学的評価法の利用は、個人差を露わに示すことになり、かえって管理上の障害を生み出すかもしれない。

この予感が、個人も管理者も、個人の技術向上について不熱心にさせている。これらを打開しようとするならば、何らかのインセンティブあるいは動機づけが必要なのである。第一線のグループリーダーがこの面でもかなりの寄与ができることは、前にもあげた米科学者の研究で示されている^{9),10)}。

ポイントは、日本的経営風土の基盤である運命共同体的小集団主義の發揮、活用である。QC サークルの活動とあいまって、現場レベルでの成果を極限まで高めることができるのではないか。勿論コンピュータ周辺にまつわる特有の人間の諸問題⁹⁾の解決あるいは取扱い技術の開発が、強く望まれる。

新QCの開発と定着化と同様に、小集団管理の方法も、第一線グループリーダーの育成と、結果としての彼らの協力により完成していこう。

(2) 計数的側面

管理の手段として完成するためには、科学的な道具が無ければならない。QC が一つの指導原理になるが、ソフトウェア開発固有の技法の確立も不可欠である。

これらは、管理者や管理スタッフの責任であり任務であるが、末端のリーダーの参画があって実効が上げられるという点が日本的経営風土の良い点でもある。おそらく、これからは彼等の主導的役割に期待したい。

課題は、① 開発コスト見積、② 開発工数見積、③ スケジューリング、④ 資源配分、⑤ 進捗度把握、⑥ 生産性把握など。

参 考 文 献

- 1) 職業的プログラムの育成シンポジウム報告集、プログラミングシンポジウム委員会(1976-7)。
- 2) 三浦：ユーザが望むソフトウェア工学、ソフトウェア工学研究資料 10、情報処理学会。
- 3) 三浦：ソフトウェア技術者の地位確立、bit、1979年10月号(巻頭言)。
- 4) 三浦：ソフトウェア製品日本の経営品質管理、情報処理、Vol. 21, No. 4 (巻頭言)。
- 5) よいプログラムを作るにはシンポジウム報告集、プログラミングシンポジウム委員会(1979-7)。
- 6) ソフトウェア技術移転に関する調査報告書、(財)機振協・協同システム開発(株)(1980-2)。
- 7) CAIに関する調査研究会報告書、協同システム開発(株)(1980)。
- 8) 上級情報処理技術者育成指針——各論第2部、日本情報処理開発協会(1979-5)。
- 9) Couger, J. D. and Zawacki, R. A.: What motivates DP professionals?, DATAMATION (Sep. 1978)。
- 10) Fitzenz, J.: Who is the DP professionals? DATAMATION (Sep. 1978)。
- 11) Couger, J. D. and Zawacki, R. A.: Some thing's very wrong with DP operators jobs DATAMATION (Mar. 1979)。

III. 討 論

本章では、主としてパネラ各氏の発表に対し、会場からの質疑応答、およびパネラ間の質疑応答に関してまとめる。ただし、編集上、同じ種類の質問もあり、論旨をまとめるうえで必ずしも当日の時間の経過通りにはなっていないことを、お断りしておく。

(1) 日本の企業におけるソフトウェア技術者の環境問題について

江村(日本IBM)：プロのソフトウェア技術者が必要であり、その教育も必要であることは私も賛成であるが、逆に、そのようなソフトウェア技術者を受け入れるような環境が現在の日本の企業体のなかにあるのだろうか、という疑問を持つ。教育をいくらやっても環境としてそういう体質ができていなかったら、教育の結果などはすぐに消えてしまう恐れがある。その理由として、通産省の情報処理技術者試験の合格者 777 人を追跡調査したデータをあげることができる。これによれば、技術者として当分やれると答えた人は 4 人に 1 人であった。残りの 4 人のうち 3 人はソフトウェ

ア技術者として機能しきれるということに対して、非常に環境に不安を持っている、という結果が得られている。もうひとつは 1978 年に調査されたデータで、227 人を対象として、ソフトウェア技術者を専門家とみなした専門職制度が実施されているかを調査した結果、227 件のうち 4 件だけが完全に実施していると答え、あとは完全には実施されていない、と回答している。このような現状では環境面での改善促進をはかっていかないと、教育だけでは解決しないのではないか。

三浦(東レ): 資格制度が環境面の改善を促進するのに役立つのではないか。ただ、法律で資格を決めても技術力の実体を伴わないと全くナンセンスである。技術が役に立つという証明を何らかの形でやっていただきたい。

米田(NTIS): 三浦氏の意見にまったく賛成である。資格制度が役に立つことの証明が重要だと思う。あえて言えば、三浦氏のインフォメーションアナリストに関しては特定の企業では非常に素晴らしい人も存在するが、試験でひっかかる人も出よう(ペーパーテストで計れる能力の限界については疑問があるということ)。また、先程も述べたが、ソフトウェアに携わる人間の社会的評価が変わらない限り、いくら教育しても駄目だと思う。

(2) ソフトウェア技術者のキャリアパスについて

江村: 三浦氏の提案では情報処理技術者を 4 つのカテゴリに分けたが、そのなかでのキャリアパスについてどのように考えているか。

三浦: 一般的にいうと、数多くいる平均的なプログラマがしかるべき適性がある訓練を受ければ上級プログラマになれる。また、それを狙うべきだと思う。上級プログラマもそれなりの努力をすれば、ソフトウェアエンジニアになることを期待している。問題はインフォメーションアナリストとシステムデザイナーだが、インフォメーションアナリストは本質的にはコンピュータないしは情報処理に関するこまかい技術は必ずしも必要とはしないかも知れない。これは、もっと違った——もし企業であれば、企業の機能や組織のベテランであって——別途養成されるべき人間だと考えている。システムデザイナーもソフトウェアからだんだん登って行って自然になれるものではないように思われる。

もし、これらを技術者として専門職というなら、イ

ンフォメーションアナリスト系統の技術者と、システムデザイナー系統の技術者と、ソフトウェアエンジニア系統の技術者の三つの柱があると考えている。また、本当のシステムエンジニアはこの三つを兼ね備えた人だと考えている。だから、ソフトウェア業界で優れたコンサルタントをしようと思うと、この三つを兼ね備えていないとだめだ。教育も、この三つでは、しかたが異なると思っている。

(3) 資格制度は役立っているか、必要と感じているか

多田(日本科学技術情報センタ): 現在、資格制度には科学技術庁の技術士制度があって、情報処理技術者として、数学応用、電算機応用、情報管理に分かれている。これがコンサルタントの国家的資格制度である。現在そういう制度があるが、それがどのように企業で尊重されているかわからない。ソフトウェア技術者の発展に資格がどのように企業のなかで評価されているかうかがいたい。

三浦: 実は私も 10 年位前に情報処理部門の電算機応用で技術士の資格を取ったが、私の会社ではその種の試験を受けて資格を取ることを奨励している。ただ、それを取ったところで処遇の面で良いことはない。現在はひとつの勉強の目的として利用している。合格すれば受験料や登録料などが会社から出て、落ちると個人が負担する。これは通産省の情報処理技術者試験でも同様である。

これを有効に利用するにはどうしたら良いかという、技術士にならなければ情報処理関係で金を取ってコンサルティングをやってはいけない。というように——公認会計士や弁護士の資格のように——したらよいだろう。そうしないとほとんど無意味である。しかし、立法化は、そういう技術を持っている人が持っていない人に比べて、明らかに有意差があるという証明をしないと社会的に認められない。また、そのような制度を有効なものにするためには、資格を持っていることによって、社会的に何らかのメリットがあるという制度にしてもらわないといけないと思う。

米田: 最近、IBM をはじめ各メーカーがエンドユーザ言語を PR している。これはある意味で両刃の剣であると思う。コンピュータを「誰でも」、「どこでも」、「いつでも」使えることは、大変結構なことであるが、これがソフトウェア技術者不要論に短絡するようでは困る。医者や弁護士の分野ではプロフェッショナルな

世界がはっきりしており、たとえば、「この薬を飲めば医者にかからなくても」などという医者は一人もいない、一方、ソフトウェア技術者の場合は、「専門のSEやプログラマなしでも」などという自己否定的なキャッチフレーズを唱えなければならないケースもある。「誰でも」、「いつでも」といえるためには、その背後で高度の専門ソフトウェア技術者が色々仕事をしているのだということをもっと認識してもらう必要があるのではないか。

極論するならば、ソフトウェア技術者の資格制度が社会的に評価されるためには、何かイベントが起きて、原因を調べたら素人がやったからだということが判明し、「やはり素人ではだめだ」ということが世の中に受け入れられることが必要かも知れない。もちろん、こういう事件は起きては困るので、できればどこかほかの国でそういうケースがあるとよい。

高橋(東京農工大): 資格制度の問題は社会の熟成度と関連がある。医者とか弁護士というのは、基本的にはメンテナンス産業であって、メンテナンス産業になるとそのような資格が物を言う。昨日来、計算機関連の仕事では60%が保守であるという意見がだいぶ出ており、そろそろそのような時代になったのかと思う。しかし、一方において、工学の分野では非常に進歩が激しいから、知識はすぐに古くなる。だから、技術者試験として従来のような知識重点の資格制度ではなく、経験的なファクタを重んずるような資格制度を考えていかないとかえって害になるのではないか。

(4) アマとプロと教育と

高橋: 国鉄のみどりの窓口や鉄鋼の大プラントシステムの設計は高度に技術集約されたものである。上記のシステムは、部品点数の多いハードウェアの組立が大変であるのと同じように部品点数の多いプログラムを作っており、大変なテクノロジーのいる仕事だと思う。ここに居る3人共、経験豊富な方々なので、プログラムは誰でも簡単に作れると思っておられるようだが、実際、学生を教育してみても、そう簡単に良いプログラムができるわけではない。勿論、計算機言語の文法を理解するのは簡単だがいいプログラム、役に立つプログラムを作るのには何年も経験がいることであると思う。

したがって、一般的に、大学教育というのは基礎的な教育で、基本的なこと、考え方とか方法をどうやって学ぶかに重点を置くべきだと思っている。細かい技

術というものは自分達がお手元にはあって短期間に修得できる、そのような能力をつけたい、というのが大学の教育だと思っている。

有山(電通大): 計算機を道具として使っている研究者がいるが、そのような人達は大きなプログラムを書いている。彼等は、自分達はかなりプログラムのベテランだと思っている。ただ、そういう人達が作っているプログラムは仲間うちで使うものであって、人様にお金を取って売りつけようという気はない。そのようなものをどう考えていいかということがある。ひとつの問題はそういうプログラムで能率の悪いものが作られていると、大学などの計算センターで非常にたくさんの無駄な時間を費やしてしまうのではないかと思う。その意味では、一般的情報処理教育、ユーザ教育のなかできちんとした教育をやる必要があるのではないか。

(5) 情報工学科系の卒業生への期待(パネリスト間の討論)

高橋: (ソフトウェア技術者の社会的評価が高くないということから) 米田氏の意見で、「従来の工学部から社長が何人も生まれているが、情報工学科は生むであろうか」という問題提起であるが、このような変化の激しい時代には、現在の大企業が20年後、30年後にも大企業として存在していることすらあやしいと思う。日本がタテ社会であるとしても、分散化の傾向は広がっており、新しい工業は新しい土壌に生まれる。社長が生まれるかどうかは情報処理に携わる人のバイタリティなどで決まることであって、逆にその位の意欲がないとだめだと思う。

また、業務知識が要求されるということだが、これは情報工学科卒の教育では不十分であるということだが、これは一面の真理ではある。しかし、たとえば事務システムなどで、情報工学科卒業の学生が1週間も解析してもわからないような事務システムをとっていること自身が、その会社が老朽化している前提だと考えれば、そのような事を驚くことはないと思う。

私は日立流の教育を受けたが、そこではT定規論というものを教育された。つまり、若い頃はT定規の縦のものをいかに深く掘るか、深く掘ると回りの土地がだんだんくずれてくるからだんだん広がる。若い頃は狭くても良いから深くやる。深くなったら自然に横がわかるようになる。また、横がわからないような浅い井戸を掘ってはいけないといわれた。

この論理によれば、先程のキャリアパスでSEになれるかなれないかというの、まじめに井戸を深く掘るか掘らないかの問題だ。つまり、自分のなかに閉じこもってプログラムだけ組んで喜んでいる人達は、三浦氏のいうように、プロではないということになると思う。だから、再教育を含めて教育というのは大学だけの問題ではなく、各人の自己啓発が最も大切なのではないかと思う。

米田：私が一番心配したのは、大企業の社長になれる、なれない、という問題などではなく、情報工学科や情報科学科、などへくる学生のキャラクタがもしかして偏向しているのではないかと心配に思って書いたわけで、バイタリティがあれば問題はない。

業務知識に関して情報工学科の教育が不十分といったのは、私自身東芝で昭和44年から50年頃まで大学卒の入社試験の面接委員をやって感じたことを振り返って述べた。ある情報工学科の学生が東芝を受験したが、電算機部門には行きたくない。ユーザ部門へ行きたいと言った。そこで、ユーザ部門の業務知識に相当するまったく基本的なことを聞いたが何も知らない。ベーシック・ソフトウェア部門ならいざ知らず、ユーザ部門へ行きたいと言ってプログラムのことしか何も知らないようでは困る。たとえば、マクスウェルの方程式は何だと聞いても名前も知らないという恐しい話もあった。勿論、落としたが。

また、T定規論については、深く掘ってゆけば良いのはわかるが、日本の企業のすべてが深く掘って行った人が十分に活躍できる広さの幅を持っているかどうかは疑問だ。しかし、趣旨はまったく賛成だ。

なお、余談だが、「行政とADP」という雑誌の79年の9月号に、電電会社の中山隆生さんが実に素晴らしい随筆を書かれている。テーマは「何故若い人に嫌われるか」というもので、嫌われるのはソフトウェア部門である。もし、これに書かれているようなことが実現されれば、今日の問題のかなりの部分が解決されると思う。

(6) 情報工学系学生の適性と訓練について

東(日電)：三浦氏の30歳停年説は、使う側からすれば大変都合のよい論理だが、一生懸命情報工学や情報科学を志し、さんざんやって30歳になったら合わないといって放り出されることは使われる人間にとっては大変迷惑で、そんなことを30歳で言われるなら専門をやっている先生が何故、お前は合わないからこ

っちへ来るなよと、はじめに言ってくれないかという感じになると思う。だから、もしそういうことがあるとすれば未然に適性——合う合わないという適性がかなりあるようにみうけられる——それをどう見抜いて指導していくか。これは大学教育ではなく、高校教育問題ではないかとも思うが。

高橋：適性を大学で何とか、と言われても困るが、ただはっきりしているのは、私が企業からの求人を受ける場合でも、その企業の教育に対する姿勢、投資についての質問をするようにしている。そこで、体力だけを要求するような企業には学生を勧めない。

私は工学部の教育では、基本的に設計に携われるようなエンジニアを作りたいと思っている。あるいは、そういうところに出したいと思っている。学生が一般ユーザの立場の企業に行きたければ勿論行ってもかまわないが、適性は本人の判断の問題だと思う。

有山：大学入試の現状をみると、受験生の多くは自分の適性や将来の志望より偏差値によって大学や学科を選んでようだし、入試では適性をみていないから、我々はプログラマとして適性のない者でも受入れて教育せねばならぬ立場にある。入学後に学科を移ることは制度的に認めていない大学が多いのではないか。一方、入学したら必ず卒業させてやるという風潮が強い。

我々の学科ではかなり厳しくやっているつもりだが、それでもなるべく卒業できるよう努力せざるを得ない。だからあまり適性のない人間が情報専門学科の卒業生として出て行くことも起りがちである。ただし卒業する頃になると本人が適性のないことを自覚して、別の方面に就職することはあるようだ。

東：教育という知識を授ける論議ばかりしているが、実際やってみると、それと同等以上の比重で訓練とかしつけの問題があるのではないかと思う。そういう意味で、たとえば基とかピアノというものは、ひとつは定石を教え、実戦でセンスを養っていく、というしくみがある。また、ピアノではバイエル、チェルニと段階を追って訓練していくしくみがある。そのようなことが何かありそうな気がする。さきほどの話を用いれば、医者にはインターンという制度がある。

三浦：訓練とかしつけという事が大切であるということはまったく同感である。

高橋：電子計算機が最初に生まれて、ケンブリッジ大学のWilkes教授がEDSACでストアード・プログラム方式でプログラミングの教育をしたがその際に、

プログラミングというのは演習 (practice) なしには絶対にわからないということを強調している。そもそも情報工学科というのはプログラムのマニアができて困るという反面、逆に、そういう演習をやらないようなことではダメで、ハードウェアの実験と、それをソフトウェアでシミュレーションするとどうなる、理論ではどうなるといった三昧一体の教育が大切である。我々自身もそういう実験とか演習を創造してやらなければ、これから対応できないのではないかと思う。

(7) 情報処理技術者の資格制度は本当に必要か

東：情報処理技術者の資格は本当にいるのだろうか、という点に関して疑問を持つ。我々は医者を選択基準がないわけで、必要悪として、ある程度、確率的な確かさがあるなということ命を託するが、システムを作る場合、みどりの窓口にしても、鉄鋼の生産管理システムにしても、担当のマネジャは真剣ですから、今までの経験と実績をみてこの人なら確かだという人を選び、そのシステムの責任者にあたらせることができる。したがって、この場合の判断は資格試験の結果よりはるかに確かだと断言できると思う。

三浦：資格がなくてもうまくいくならば、それに越したことはない。しかし、一般にソフトウェア技術者のレベルが低く、非常に問題であり、不満足であるといわれているので、それを改善するひとつの方法として提案している。

ある閉じたグループのなかで、マネジャが責任をもって真剣に判断するのであれば一向にかまわないし、企業のなかでの問題として解決できる。ただ、問題はこれからは一般企業のなかで、自前でシステムを作ることはおそらくだんだんなくなるだろうと思うからである。実際に、企業のなかで不可欠な人間は、おそらく、私の分類のインフォメーションアナリストだと思う。ほかの部分は、ソフトウェアハウスに期待しなければならないだろう。というのは、そろばんの代りにコンピュータを使うのが今までのコンピュータの利用だとすると、これからは、それがオンラインシステムとか、データベースでということによって完成してしまう。したがって、今までそろばんでやっていなかったような仕事をコンピュータでやらせる時代にきていると思う。そうすると、今までみたいにやみくもに開発するのではなく、知恵の勝負となる。そうすると、ユーザがたくさんプログラマをかかえておくことが必要なほど、仕事のあることはなくなるだろう。また、停年に

なるポテンシャルの大きい人をたくさんかかえてあとで困ることになるから、そのようなプログラマはユーザでは探らなくなる。結局、必要なときにソフトウェアハウスなどがちゃんとやってくれるような時代であることをユーザは望んでいる。ただ、現在、残念ながらソフトウェアハウスは一部の企業を除いては著しくレベルが低い。だから、資格制度とか、教育とかを重視したいのは、ソフトウェアハウスのレベルを上げたからだと、というのが本心である。ユーザからみて、どういう人が実力があるかを判断するには、資格がひとつの目安となる。少なくとも、資格があれば最低限度保証できるだろう、という安心感がある。勿論、その場合には技術だけでなく倫理規定も必要である。そのような意味で資格制度が役に立つのではないかと考えている。

(8) 会場への質問

以下の2問を会場の出席者に直接賛否を挙手で問うた。

第1問：米田さんの提案にもあった、産学協同の「ソフトウェア研究所」の設立について、会場の皆さんは、どう考えるか。

結果は、賛成5%位、反対は0名、であった。この問題は会場の会員諸氏には、研究所のイメージが現実遊離とか、現実に参加するには時間的、経費的にも問題があると思ったか、とにかく判断がつかかぬというのが現実のように理解した。

第2問：三浦さんの提案の「ソフトウェア技術者の公的な資格制度を考えてみてはどうか」について。

結果は、賛成50%強位、反対5%位、どちらでもないが30%位であった。この結果からは、何等かの公的な資格制度があった方が身分保障されるという三浦氏の提案はかなり説得力を持っていたと言わなければならない。一方、反対は計算機メーカーの人達に見受けられたが、今回の参加者は、ユーザおよびソフトウェアハウスの中堅の人達が多かったことも付言しておく。

IV. おわりに

各パネリストの論旨は、II章で明示してあるので、パネル討論会の当日の各パネリストの論旨の発表部分は、省略した。当日の発表では強調すべき点、討論を活発にするための意識的、かつ刺激的な発言、的確な事例など興味のある発言があったことを付言しておく。残念ながら、そのすべてを紹介するのは紙面の都合

合で不可能である。

パネル討論に結論を出す必要はないが、各パネリストから提案された重要な提案について以下にまとめる。学会の会員諸氏にとって、「情報処理技術者の教育は如何にあるべきか」の自問自答の材料にさせていただければ幸である。

(1) ソフトウェア技術者の公的資格制度の実現は是か非か。また、実現した場合の社会へのインパクトは何か。

(2) 日本の風土にあった QC 運動に相当する、ソフトウェアに関する新しい QC 運動を起してはどうか。

(3) 情報工学系の学生は、狭いプログラミングの領域のマニアにはならず、広い視野を持つようになってほしい。また、大学教育も、toy プログラム作りだ

けではなく、チーム・プレー、リーダーシップの育成なども心掛けてほしい。

(4) 産学協同による、ソフトウェア技術研究所などを設立することを考えてみてはどうか。

(5) ソフトはケース・スタディを重視せねばならない。ビジネス・スクールと同じく、失敗の歴史、失敗の記録をつくり、再教育に活用する必要がある。

(6) 情報工学系の学科では、学問の体系化の努力が大切だ。

(7) 情報工学系の学科では、訓練（演習）とか、しつけという面も十分考えてほしい。

なお、最後に、以上の内容について、何等かの誤解が生ずるとすれば、それは、本パネル討論の司会者の高橋個人の独断と偏見による。ご寛容の程をお願いしたい。