

## 解説



### ソフトウェア製品生産の人的側面†

東 基 衛††

#### 1. はじめに

近年、いろいろな分野で人間の能力開発に、多大の関心が寄せられるようになってきている。人間の能力開発のための学習法や、産業における行動科学などがその例であり、これまでいろいろな方面から研究が行われ、成果が見られてきている。

しかしソフトウェア製品生産のための研究の分野では、多くの努力はソフトウェアの構造や、言語、方法論、ツールなどに注がれているのが現状である。人的側面に関する研究は、いくつかその例が見られるといえども、その数は少ない。

このような状況は、工業生産の分野で 1900 年から 1920 年代に、F.W. テイラー、F.B. ギルブレス、H. フォード一世などが、科学的管理法<sup>1)</sup>や、動作時間研究、オートメーション化などに努力を集中していたのを想起させる。これらはたしかに多くの成果をあげることができた。

しかし、今回我々は、その後 1927 年から 1932 年に E. メイヨー<sup>2)</sup>らによって行われたホーソン実験以来、人間の労働意欲が、正しい作業方法と同等以上に、生産性に大きな影響を与えるということが注目されるようになり、多くの研究がなされてきたことを知っている。この人間の労働意欲に関する研究は行動科学といわれているが、ソフトウェア技術者を対象にした行動科学の研究はまだ少ない。

ソフトウェア開発に関する工学的アプローチは、Pascal のような優れた言語や、UNIX のもとでのプログラマーズワークベンチのような優れたツールを生み出すなど、多くの成果を収め、前進した。しかし、成果の中の多くはプログラムの製造工程以降を支援するものである。勿論、ISDOS や SREM のように要

求定義を支援する言語と解析ツールとして有名なものがあるが、これらはドキュメンテーションと解析を支援するものであり、人間にとって替るわけではない。

この状況はハードウェアの生産についても同様である。トランスファーマシンや工業用ロボットの開発により、製造工程のオートメーション化は進歩したが、設計以前が自動化されたわけではない。勿論、デザインオートメーションというものは存在している。しかし、これは IC のレイアウト設計のような特定の分野に限られている。その他の分野では、たとえば建築の設計に見られるように、構造計算と製図を支援するというような範囲に留まっているといえよう。

要求を定義する過程（これはシステム分析、要求分析、あるいは製品計画などが該当する）および、その要求をどのように実現するかをきめる設計過程は、本質的には人間の創造的活動である。したがって思考過程が明らかでかつ設計結果に好みの入らない場合のみ、設計の自動化が可能となる。ソフトウェアの場合でいえば、IFPS<sup>3)</sup>のように、要求定義からプログラムをゼネレートして実行し、結果を出すツールはその例と考えると良い。

このように考えてくると、どんなにソフトウェア工学が進歩しても、ソフトウェアのライフサイクルを通して見た場合、人間の役割がなくなることはあり得ない。否むしろコンピュータの応用分野の拡大と多様化によって、ますます人間の重要性が増してくるものと思われる。したがって高品質のソフトウェアを効率良く開発するための、人的側面からの研究が切望される次第である。

ここではその問題提起として、まずソフトウェア開発の作業と、人的側面からのアプローチとの関係を明らかにするための枠組みを述べ、次に現状ではどのような研究が行われているかそのサーベイを行い、最後に人的側面からの問題が多い分野として事務処理アプリケーションシステムの要求定義を一つの例としてとりあげて、人的側面からの考察を行ってみた。

† Human Factors in Software Product Engineering by Motoei AZUMA (Engineering Manager, Software Product Engineering Laboratory, Nippon Electric Co. Ltd.).

†† 日本電気 (株) ソフトウェア生産技術研究所

## 2. 人間的側面からの問題の枠組み

### 2.1 枠組みの目標

製品としてのソフトウェアを生産する目的には、いろいろのものが考えられるが、おおむね次のように考えられる。

- ・ 顧客に、要求を満たす、高品質の製品を安い価格で提供する。
- ・ 従業員に、高い収入、安定した職場、仕事を通じての生き甲斐や達成感などを提供する。
- ・ 資本家に、高い利益（配当）を提供し、また企業の安定と成長を約束する。

そしてこれらを通じて最終的には、社会の文化の向上と幸福の追求に寄与することになる。そのために行われなければならないことは多いが、

- ・ ソフトウェア要求の適格な把握と定義
- ・ 要員のレベルの向上とモチベーション
- ・ 計画の質的向上
- ・ 単なる個人の和ではなく、チームとしての効率向上

・ 自動化ツールの開発と利用の促進などは重要である。これらはいずれも人間的側面からのアプローチを必要とする。第5項の自動化ツールは一見関係なさそうに見えるが、ここにおいても良いマンマシンインタフェースや、人間の思考過程を助けるツールなど人間的側面からのアプローチは不可欠である。

この枠組みの目標は、ソフトウェア製品生産における人間的側面に科学または工学の眼を向け、科学的方法あるいは工学的方法によって問題を解決してゆくために、どのような問題があるかを明らかにすることにある。

枠組みは3つの軸から成っている。ソフトウェア・ライフサイクルの軸、問題領域の軸およびアプローチの軸である。

### 2.2 ソフトウェア・ライフサイクル

ソフトウェアのライフサイクルにより、つまりソフトウェア開発の工程によって、人間的側面の問題も異なってくる。システム分析者、プログラム設計者など、従事する人の職種や、最適なチームの構成などが異なってくるからである。

ライフサイクルをどのような段階に分けるかについては諸説があり、開発するソフトウェアの種類や規模によっても異なる。人間的側面、特に分業に注目した

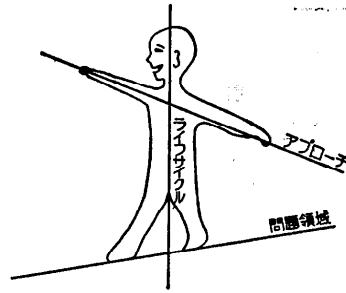


図-1 人間的側面の枠組み

ライフサイクル上の枠組みと問題の例は次のように考えられる。

- ・ 問題定式化とシステム要求定義
- [例] エンドユーザと DP 部門の意志の疎通
- ・ システム設計とソフトウェア要求定義
- [例] システム分析者、システム設計者およびプログラマのチーム活動
- ・ ソフトウェア設計
- [例] 設計方法論と設計者の適性および訓練
- ・ ソフトウェア製造およびテスト
- [例] 設計と製造の分離と製造担当者のモラル
- ・ システムテストおよび移行
- [例] テストの効率とマンマシンインタフェース
- ・ 運用、評価、および保守
- [例] 一人の保守担当者の保守し得るプログラム規模の限界

### 2.3 問題領域

人間的側面を問題領域から見た場合、技術問題と管理問題に大別される。

技術問題は、いわゆる狭義のソフトウェア工学の問題領域そのものである。たとえばプログラミングスタイルにおける、プログラムの理解しやすさの定量評価尺度や、ツールのマンマシンインタフェースにおける操作性の問題は重要である。

ここでは、本特集のテーマから、管理上の問題にしばって見ることにする。

管理の枠組みとして有名なのは、R.N. アンソニのフレームワーク (framework)<sup>4)</sup>である。アンソニは、実際の作業（オペレーション）の上に3つのレベル、つまり下から順にオペレーショナルコントロール、マネジメントコントロールおよび戦略意思決定の各レベルを設定した(図-2)。これは大変有効な枠組みであるが、問題の位置づけを更に明瞭にするために、2つの軸を追加したい。

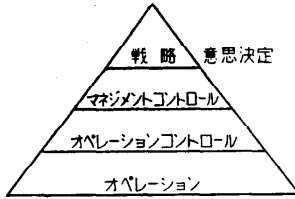


図-2 管理のフレームワーク

1つは管理の時点である。これはシステムライフサイクルとして把握される。しかし、ここでは、開発プロジェクトに重点を置き、プロジェクト開発前の不断の管理、プロジェクトの開始、プロジェクトの進行中およびプロジェクトの終了として把握する。

ほかの1つは管理の対象である。経営工学では管理すべき資源として4M (Man, Money, Material, Machine) が挙げられる。ソフトウェア開発プロジェクトの場合これらに加え、資源としてプログラムとデータが管理されなければならない。このほか管理対象としてはスケジュール、品質、技術が挙げられる。

これらを考慮した管理領域の枠組みは図-3のようになる。この枠組みから人的側面に関するものを抜き出すと、次のような問題領域が考えられる。

- ・ 計画と意思決定
- [例] プロジェクトマネージャの意思決定過程と、意思決定支援システム
- ・ 人事管理
- [例] キャリヤパス (仕事の経験を通じて人材の育成を図るための望ましい経路)、プログラマの停年と高令者対策
- ・ 勤務管理
- [例] フレックスタイムと生産性
- ・ 教育訓練
- [例] ソフトウェア技術者の教育カリキュラム
- ・ チーム編成
- [例] 日本の雇傭環境における分業形態
- ・ 要員の適性と配置
- [例] ソフトウェア工学の発達と要員適性の把握方法
- ・ 見積りと標準工数
- [例] 要員のパフォーマンス標準
- ・ オリエンテーション
- [例] プロジェクトの開始とオリエンテーション項目
- ・ モチベーション
- [例] ソフトウェア技術者の欲求レベルとモチベーション

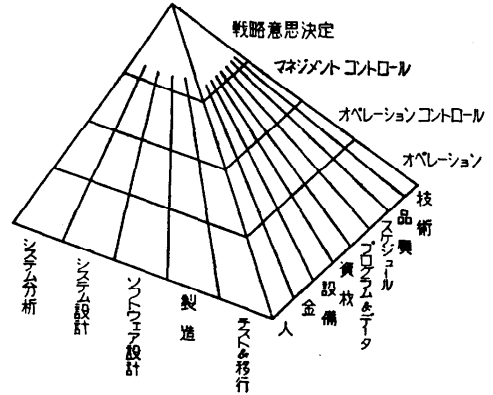


図-3 管理領域のフレームワーク

- ・ コントロール
- [例] 創造活動の進捗率の定量的把握とコントロールの方法
- ・ コミュニケーション
- [例] ソフトウェアプロジェクト内の情報伝達の経路、方法と最適な頻度
- 2.4 人的側面からのアプローチ**
- 人的側面といってもいろいろな方面からのアプローチが存在する。これらは単独で行われることも多い。しかし、人間という多面性をもつ対象を扱うので今後1つの問題に対して、各アプローチを複合した、学際的なアプローチが必要となるであろう。主なアプローチは次のようなものである。
- ・ 社会科学
- [例] 社会構造がプロジェクトチーム構造に及ぼす影響
- ・ 大脳生理学
- [例] 脳のメカニズムと効果的学習法
- ・ 医学
- [例] ソフトウェア技術者の職業病
- ・ 産業心理学
- [例] 変化への抵抗とシステム移行、プログラム作成時の心理状態とバグ
- ・ 行動科学
- [例] ソフトウェア技術者の欲求レベルと動機付け
- ・ 創造工学
- [例] システム分析者の創造性開発
- ・ 人間工学
- [例] 単純なミスによるバグを起し難いプログラムの入力方法
- ・ 環境工学

[例] プログラマの作業環境と生産性

・ 管理工学

[例] 頭脳集団におけるリーダーシップのあり方

### 3. 人間的側面からの研究のサーベイ

#### 3.1 工数見積りと生産性

工数見積りの問題は古くて新しい問題である。1人のプログラマが1日に行う作業量は、バラツキが大きく、最大と最小との差は15倍とも30倍ともいわれている。したがってどのように工数を見積るかは、大きな問題である。特にソフトウェア製品の場合には、見積りの誤りは安すぎる受注や納期の大幅な遅延の原因となり、会社の営業成績に影響を与える。

F.P. ブルックス Jr. は、見積りは単純に人月の積算ではできないことを“ソフトウェア開発の神話”で指摘している<sup>9)</sup>。

B. エスタリングは、ひとつのプロジェクトに参加する人が増加する程、コミュニケーションに多くの時間がとられ、1人あたりの作業効率が悪くなることに注目して、モデルを作り、費用と時間とマンパワーとの関係を定量化している<sup>9)</sup>。

IBM 社では1972年にソフトウェア計測プロジェクトを開始した。C.E. ワルストンと C.P. フェリクスはこの結果を、プログラミングの測定と見積りの方法として、生産性に影響を与える29の要因を中心に報告している<sup>7)</sup>。

また、R.W. ウルバートンは、費用と見積りの関係に注目し、プロジェクトで発生する工数が、工程別にどのように推移するかをモデル化し、そのピークをならす方策を論じている<sup>9)</sup>。

このほか、プログラマの生産性標準に関しての E. クライスラーの報告や、プログラマの生産性向上の戦略に関する A. イエイツの報告も注目される<sup>9), 10)</sup>。

#### 3.2 モチベーションと行動科学

ソフトウェアの生産に従事する人達の行動は、ハードウェアの生産に従事する人とは異なるものと考えられる。むしろハードウェアの設計者や、研究者に近い行動をとるのではないだろうか。

研究者の行動を大がかりに調べてまとめたものとして、D.C. ベルツ他の“創造の行動科学”<sup>11)</sup>を注目したい。ベルツ等はこの中で、科学者を大学および研究機関と企業に属する者とに分け、自由性、コミュニケーション、多様性、献身、動機づけ、満足度、類似性、創造性、年齢、年齢と雰囲気、調整度、集団などをと

りあげ、業績との関係を分析している。たとえば、内的要因（自己のアイデア）により仕事をしている人は監督者のみから刺激を受けている人に比べて、高い業績を上げていること、また同僚と接触の機会を多く持つ人は、持たない人に比べて業績が高いことが、指摘されている。

また北カロライナ大学の C.K. ウードルフは、プロジェクトマネージャがメンバに影響力を行使するための要因として、個人のニーズと、仕事の満足度を取りあげ、それぞれの DP 分野における調査結果を引用して論じている<sup>12)</sup>。

このほか、プログラマは何に興味を持つかを調査した P. バーンズの報告も興味深い<sup>13)</sup>。

#### 3.3 チーム編成

チーム編成に関する文献は比較的多い。最も大きな影響を与えたものとして、F.T. ベーカーのチーフプログラマチームに関する論文が挙げられる<sup>14)</sup>。チーフプログラマチームについては、このほか前記“ソフトウェア開発の神話”など多くの文献がある。

G. シューブ他は、チーフプログラマチームのダイナミクスと題し、インフォーマルコミュニケーション、生産性、モチベーションなどかなり詳細な報告を寄せている<sup>15)</sup>。しかし、日本ではチーフプログラマチームの実施例の報告は少なく、日本的な労働環境下では、その効果を疑問視する向きも多い。

B.W. アンガーは、チームの効率をきめる要因として、

- (1) グループのメンバが注ぐ努力の水準
- (2) メンバがその仕事に持ち込む知識とスキル
- (3) 仕事遂行上、そのグループのために用いられ

た、パフォーマンス戦略の適切さを取り上げている<sup>16)</sup>

W.J. フロスト、C.M. ボーアンのチーム開発に関する論文も注目される。彼等は、“我々の組織は、過去200年以上もカトリック教会やシーザのやり方に学ぶという誤った仮定の上で動いてきた。”という引用を用いて、チーム開発の重要性を強調し、マネジリアルグリッド(Managerial Grid)に基づく開発方法を述べている<sup>17)</sup>。

このほか、K.M. パートルが、“共働 EDP チームの建設”(Building Synergistic EDP Teams)と題して共同作業が円滑にできるチームを作るのは容易でなく、したがって常に新しいチームを作るのではなく、良いチームを管理ベースで保持しつづけることが重要

であることを主張しているのが注目される<sup>18)</sup>。

### 3.4 その他

ソフトウェア開発におけるユーザの役割に関しては D. スキーン<sup>19)</sup>, V.G. ガラヤルデ<sup>20)</sup>, S.E. カバーデール<sup>21)</sup>等がある。

創造性開発に関しては、ソフトウェア工学の分野ではないが、KJ 法<sup>22)</sup>,<sup>23)</sup>, NM 法<sup>24)</sup>, TTHS 法<sup>25)</sup>などのように、日本に勝れた成果が多い。また F. フェスタの学際的なアプローチによる学習プロセスのモデル<sup>26)</sup>ケストラ<sup>27)</sup>や市川<sup>28)</sup>の創造工学も興味深い。

このほかに日本での興味深い報告として、SRA の三田守久等のセンシティブィティトレーニング (Sensitivity Training) の実施結果に関するものがあり、今後が期待される<sup>29)</sup>。

また人間的側面の古典として、G. ワインバーグのプログラミングの心理学も見落すことはできない<sup>30)</sup>。

## 4. 要求定義における人間的側面

### 4.1 人間的側面より見た要求定義の問題

要求定義とは、図-4 のように問題を入力とし、ソフトウェア要求仕様を出力とするプロセスである。入力される問題は、非常に漠としていてコンピュータに期待する効果だけがきまっておき、コンピュータで何を行うかがきまっていない場合もある。また、何を入力し、どのようなアルゴリズムで処理し、何を出力するかまで明らかな場合もある。前者の場合には、問題は何かを明らかにする問題の定式化や、問題を解決するために何を作るべきかをきめるための要求分析が必要になる。また後者のような場合には、要求を仕様化すれば良い。

要求定義のための技術は、対象とする問題やコンピュータの応用分野によって、適する技術が異なる。またそれに対応して、人間的側面から見た問題も異なる。そこで事務処理アプリケーションシステムの要求定義における人間的側面について考えてみることにする。

事務処理アプリケーションシステムの場合、人間的側面から見た要求定義の特色は次のようである。

- 問題の構造が多くの場合不明確で、構造をどう把握するかは分析者の主観に依存する。

- 最初に提起された問題は必ずしも真の問題ではない。問題の認識が甘く、真の問題から派生した表面的なものかもしれないし、本音と建前の乖離から、偽った問題かもしれない。

- 要求分析には、関係各部門のエンドユーザを始

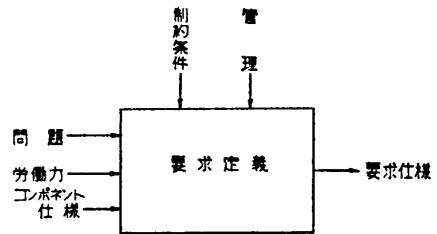


図-4 要求定義のシステムモデル

めシステム分析者、DP 部門のマネジャなど多くの分野の人が参加する。

- 要求分析のために意見を集めても、必ずしも本音が得られるとは限らない。面接者が意見を誘導してしまう恐れもある。また、被面接者が、そのシステムが自分の地位を危うくすると思っておりしかも表面上協力的に見せようと思っている場合には、真意を引き出すのには大きな困難を伴う。

- 環境の変化が激しく、要求の変化を引き起す。このため分析の期間が長いと途中で条件が変化し、最適な要求を見つけるのが難しい。

- 要求仕様に関しては、エンドユーザが最終的な決定権を持つ。したがってエンドユーザが決定しやすい仕様書や、意思決定過程が用意されねばならない。

- 要求の決定は不確実性下の意思決定問題である。したがって、意思決定には経験と直感が重要な役割を果たす。

- 意思決定は 0 か 1 かよりアナログ的な場合が多い。

これらの特色から、要求定義は、それ以後の作業段階に比べて、ツールの活用や形式化された方法論や仕様化技術の比重よりも人間的側面からのアプローチの重要性が高い。

### 4.2 要求定義の人間的側面に関する考察

以下に要求定義の人間的側面に関する考察を述べることにする。これらはこれまでの経験に基づく仮説であって、それを証明するだけのデータをもっている訳ではない。

(1) 要求定義のためのチームは、エンドユーザ主導型が良い。使用可能なハードウェアおよびソフトウェアが多様化し、システム形態も DB, DC, 分散処理など多様化した現在、最適なシステムの選択は難しくなっている。しかし、それ以前に、これらの多様化によって、アプリケーションが大規模かつ複雑になったため、それに必要な業務知識の習得は、DP 部門の要

員の能力を越すようになってきている。

(2) 要求分析チームは、意思決定者、業務専門家(エンドユーザ)、分析技術専門家(システムアナリスト)、システムリソース精通者(システムエンジニア)、およびソフトウェアエンジニアなどから成る混成チームが良い。その運営形態はチーフプログラマチームのような外科医型チームは不適當である。むしろスパイ大作戦型の専門家集団チームが良い。

(3) チームは当然の事ながら、知識、スキル、マンパワーなどが重要であるが、それらにも増して、人の和が重要である。人の和が良ければ文殊の知恵が生まれるが、悪ければ足を引っ張り合い、1+1が1以下になってしまうことも珍しくない。

(4) 最終的な要求仕様の意思決定は、一人のリーダーに集中する方が良い。一般に多勢の合議制で仕様を固めるよりは一人の個性を伸ばした方が良いシステムができる可能性が高い。この場合各専門家の役割りは、十分にいろいろな角度から意見を具申することにある。たとえその意見が採用されなかったとしても、その人が役に立たなかったわけではない。

(5) リーダに必要な事は、耳ざわりの悪い意見にも耳を傾ける度量と、自分に反対の意見でも良いものも採用してゆく柔軟性である。

(6) 構造化されていない問題から要求を定義してゆくことは、多分に感覚的な仕事である。工学的な方法と、洞察力、センス、直感、経験など人間的な要素を両立させなければならない。このため、リーダーには独自の哲学を持ち、それに基づいて行動することが要求される。

(7) 要求仕様には追加や変更はつきものである。たとえ環境の変化がなくても、始めからすべての要求仕様が完全に整うことは、現実にはあり得ない。設計者は不完全な要求仕様に基づいて仕事を開始する。要求定義者は設計者や製造者の質問に答えることによって、要求仕様の細部を完成させてゆくことになる。したがって要求定義リーダーは、チームのコミュニケーションを円滑に保つことが大切である。

(8) 要求は多くの場合、白か黒とか1か0というようにはっきりとできるものではない。人間が考えることは、“どちらでも良い”とか“ややこちらの方が良い”というようにアナログ的なものである。また多分こうなりそうというような確度の問題もある。要求仕様はこのような要求の強さや確度が表現できる方が自然である。

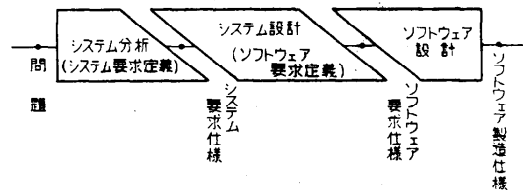


図-5 要求定義の工程

(9) 要求定義という言葉自体あいまいなものであるが工程と要求仕様との関係は図-5のように考えられる。これらの工程の区切りは時間的なものではなく、その要求仕様の意思決定者が誰かに依存する。したがって最初の工程の一部が、コーディング段階まで残っていても不思議ではない。要求定義のツールや方法論はこの状況に対応できなければならない。

(10) 設計や製造のやりやすいか否かによって、完成時期やコストが異なる。つまり要求の満足度と納期やコストとはトレードオフの関係にある。この関係を調整し、要求を最適なレベルになるようにするのはリーダーの重要な役割である。

(11) 一般にビジネスアプリケーションでは問題の構造が不明確な場合が多い。問題の構造を明らかにする方法として KJ 法は有効である。KJ 法はまた参加者の参加意識を高め、コンセンサス作りに役立つという効用もある。

(12) 要求分析の方法論には、帰納的アプローチや演繹的アプローチ、またトップダウンやボトムアップなどいろいろのものがある。これらの選択基準はどれが勝れているかではなく、むしろどれがそのチームの人に理解しやすいかが重要である。これはいわば個人の頭脳構造と、方法論との同調現象と考えられる。

## 5. おわりに

これまで述べてきたように、より良いソフトウェア製品をより効率良く開発してゆくための人間的側面は、技術面からも管理面からも解明してゆかなければならない点が多い。

参考文献にも見られるように、アメリカでは、Annual Computer Personal Research Conference (ACM SIGCPR) のような専門会議も行われているようであるが、質量ともに十分という訳にはいかないようである。今後の発展が期待される。気がかりな点は、上記会議が 1977 年に行われた際にはかなり多くの論文が発表されたけれども、1978 年には開催されず、1979

年には少数の論文にとどまっている点である。

日本においては、ソフトウェア製品生産における人的側面からの専門的研究は、アメリカにおけるよりも更に少ない。しかしこの分野は、日米の相違が大きい分野でもある。したがってアメリカの成果がそのまま日本にあてはまるとは限らない。今後大いに日本的な人的側面の研究が行われ、成果を収めることを期待する次第である。

### 参 考 文 献

- 1) Taylor, F. W.: 科学的管理法, 産能大出版部.
- 2) 進藤勝美: ホーソンリサーチと人間関係論, 産能大出版部.
- 3) Tutorial: An Introduction to Computer-Assisted Planning Using the IFPS, Execucom (1979).
- 4) Anthony, R. N.: 経営管理システムの基礎, ダイヤモンド.
- 5) Brooks Jr., F. P.: ソフトウェア開発の神話, 企画センタ.
- 6) Esterling, B.: Software Manpower Costs: A Model, Datamation, 1980-3, p. 164.
- 7) Walston, C. E. and Felix, C. P.: A Method of Programming Measurement and Estimation, IBM System Journal Vol. 16, No. 1, p. 54 (1977).
- 8) Wolverton, R. W.: The Cost of Developing Large Scale Software, IEEE Trans. on Computer (1974).
- 9) Cryster, E.: Programmer Performance Standard, Journal of Systems Management (1978-2).
- 10) Yates, A.: A Strategy for Computer Programmer Productivity Improvement, Proceedings SIGCPR, 1975, p. 85.
- 11) Pelz, D. C. and Andrews, F. M.: 創造の行動科学, ダイヤモンド(1971).
- 12) Woodruff, C. K.: Consideration of Selected Personality—Job Satisfaction Constructs Relevant to Project Management in Data Processing Organizations, Proceedings SIGCPR(1979).
- 13) Barns, P.: Programmer Paranoia Revised, Proc. SIGCPR, 1975, p. 114.
- 14) Baker, F. T.: Chief Programmer Team Management of Production Programming, IBM System Journal (1972-11).
- 15) Shabe, G. et al.: Team Dynamics in Systems Development and Management, Proc. SIGCPR, 1977, p. 164.
- 16) Unger, B. W.: Improving Team Productivity in System Software Development, Proc. SIGCPR, 1977, p. 104.
- 17) Frost, W. and Vaughn, C.: Within Context The Application of Team Concepts to Work Teams, Proc. SIGCPR, 1977, p. 1.
- 18) Bartol, K. M.: Building Synergistic EDP Teams, Proc. SIGCPR, 1977, p. 18.
- 19) Skeen, D.: User Improvement with EDP Systems Development, Proc. SIGCPR, 1977, p. 62.
- 20) Garayalde, V. G.: Role of the User as a Team Member, Proc. SIGCPR, 1977, p. 72.
- 21) Coverdale, S. E.: A New Method for Improving Communications Between User Groups and Systems Specialists, Proc. SIGCPR, 1977, p. 80.
- 22) 川喜田二郎: 続発想法, 中公新書.
- 23) 川喜田二郎: 知の探検学, 講談社現代新書.
- 24) 中山正和: 創造思考の技術, 講談社現代新書.
- 25) 浜脇英一: 経営革新の技法, ダイヤモンド (1979).
- 26) Vester, F.: 考える・学ぶ・記憶する, 講談社.
- 27) Koestler, A.: 創造活動の理論 (上, 下), ラテイス(1964).
- 28) 市川亀久彌: 創造工学, ラテイス(1977).
- 29) 三田守久: ソフトウェアプロジェクトの管理—そのあるべき姿をもとめて—, Proc. ソフトウェアコンベンション, 1978, p. 69.
- 30) Weinberg, G. M.: The Psychology of Computer Programming, Van Nostrand Reinhold Company (1971).

(昭和55年6月12日受付)