

Newsletter from Software Engineers Association

Volume 4, Number

4-5 1989

# 目 次

		1
野村	敏次	2
		25
		31
		35
藤垣	裕子	38
野村	敏次	56
		59
		61
		63
	藤垣	藤垣 裕子

**ソフトウェア技術者協会(SEA)**は、ソフトウェア・エンジニアの、ソフトウェア・エンジニアによる、ソフトウェア・エンジニアのための団体であり、これまでに日本になかった新しいタイプのプロフェッショナル・ソサイエティたることを目指して、1985年12月20日に設立されました。

現在のソフトウェア技術が抱える最大の課題は、ソフトウェア・エンジニアリング研究の最前線(ステイト・オブ・アート)と、その実践状況(ステイト・オブ・プラクティス)との間に横たわる大きなギャップを埋めることだといわれています。ソフトウェア技術の特徴は、他の工学諸分野の技術にくらべて属人性がきわめて強い点にあります。したがって、そうしたテクノロジー・トランスファの成否の鍵は、研究者や技術者が、既存の社会組織の壁を越えて、相互の交流を効果的に行うためのメカニズムが確立できるか否かにかかっています。SEAは、ソフトウェア・ハウス、計算センタ、システム・ハウス、コンピュータ・メーカ、一般ユーザ、大学、研究所など、さまざまな職場で働く人々が、技術的・人間的交流を行うための自由な<場>であることをを目指しています。

SEAの具体的な活動としては、特定のテーマに関する研究分科会(SIG)や地方支部の運営、月刊機関誌(SEAMAIL)の発行、各種のセミナー、ワークショップ、シンポジウムなどのイベントの開催、既存の学会や業界団体の活動への協力、また、さなざまな国際交流の促進等があげられます。

なおSEAは、個人参加を原則とする専門家団体です。その運営は、つねに中立かつ技術オリエンテッドな視点に立って行われ、特定の企業や組織あるいは業界の利益を代表することはありません。

代表幹事: 岸田孝一

常任幹事: 臼井義美 久保宏志 熊谷章 佐藤千明 藤野晃延 松原友夫 吉村鉄太郎

幹事: 青島茂 天池学 飯沢恒 稲田博 岩田康 岡田正志 落水浩一郎 片山禎昭 川北秀夫 杉田義明 武田知久

田中慎一郎 玉井哲雄 中来田秀樹 中園順三 中野秀男 西尾出 野村敏次 野村行憲 針谷明 平尾一浩

深瀬弘恭 藤本司郎 北條正顕 細野広水 盛田政敏

会計暨事: 辻淳二 吉村成弘

常任委員長: 臼井義美(技術研究) 久保宏志(企画総務) 藤野晃延(会誌編集) 杉田義明(セミナー・ワークショップ)

分科会世話人 環境分科会(SIGENV): 田中慎一郎 渡邊雄一

管理分科会(SIGMAN):相沢圭一 川北秀夫 芝原雄二 野々下幸治

教育分科会(SIGEDU): 大浦洋一 杉田義明 中園順三

ネットワーク分科会(SIGNET): 青島茂 野中哲

法的保護分科会(SIGSPL):能登末之

CAI分科会(SIGCAI): 大木幹雄 寺嶋祐一 中谷多哉子 中西昌武

ドキュメント分科会(SIGDOC):田中慎一郎 野辺良一

支部世話人 関西支部: 臼井義美 中野秀男 盛田政敏

横浜支部:熊谷章 林香 藤野晃延 松下和隆

長野支部:小林貞幸 佐藤千明 細野広水

名古屋支部:岩田康 鈴木智

九州支部:植村正伸 小田七生 藤本良子 平尾一浩 松本初美 中島泰彦 後藤芳美

SEAMAIL編集グループ: 岸田孝一 佐原伸 芝原雄二 関崎邦夫 田中慎一郎 中村昭雄 長井修治 成沢知子 野辺良一 藤野晃延 渡邊雄一

SEAMAIL Vol. 4, No. 4-5 平成元年7月25日発行 編集人 岸田孝一

発行人 ソフトウェア技術者協会 (SEA)

〒102 東京都千代田区隼町2-12 藤和半蔵門コープビル505

印刷所 サンビルト印刷株式会社 〒162 東京都新宿区築地町8番地

特価 1,000円 (禁転載)

# 編集部から

なぜか、このところ快調なペースで出ている SEAMAIL です.もしかしたら、伊豆半島の群発地震はそのせいかも知れません.

この号では、(1)昨年の夏に実施した環境実態調査、および、(2)この冬行ったテクノストレス調査、の2つのアンケート結果の分析報告を中心にまとめてみました。

### 1. 環境アンケート

このレポートは、昨年の7月に行ったアンケート調査の結果を、昨年秋のセミナー・ウィークおよび第2回日中シンポジウムでの発表の成果を踏まえて、調査WGのチーフをつとめられた野村さんが、ソフトウェア・シンポジウム'89 のために執筆されたものです。結果の一部はこの報告では省略されていますが、そのあたりの分析は、また別の機会にお願いしようと思っています。

#### 2. 雑多なアンケート

全会員および春のセミナー・ウィーク参加者を対象にしたいくつかのアンケートの生の結果をまとめてみました. 野村さんのレポートとあわせてお読みください.

#### 3. テクノストレス

この冬に実施したテクノストレス・アンケートは、かなり関心が高く、多くの方々から回答が寄せられました。このレポートは、調査を担当された東京大学の藤垣先生が、やはりソフトウェア・シンポジウムのためにまとめられたものです。

**4. 幸せな家庭**環境調査を担当された野村さんが、半分はジョークをこめて **SEAMAIL** に書かれた「家庭環境」に関するインフォーマル・アンケートの回答分析です.某農林水産大臣の御感想も聞いてみたい気がしますが、あなた自身はどう思われますか?

#### 5. これからの予定

編集部は、以前として壊滅状態ですが、幸いまとめる材料はたくさんそろっていますので、これから秋にかけて、うまくすれば $2\sim3$ 冊、それぞれかなり分厚いものが出せそうです。しかし、バラエティもほしいので、ぜひ、みなさんの原稿を事務局宛お送りください。

# ソフトウェア開発技法/環境の現状と動向

日本電子計算 情報システム事業部

## 野村 敏次

#### 1. はじめに

1987年3月にカリフォルニアのモントレーで行われた第9回ICSE (International Conference on Software Engineering) において、今までソフトウェア産業振興協会や情報サービス産業協会の技術委員会専門部会で、筆者が中心となって調査/研究してきた、我が国におけるソフトウェア開発環境の実態を「Use of Software Engineering Tools in Japan」という論文にまとめ発表した。これは、主として1983年に調査した、「ソフトウェア開発環境の改善に関する調査研究」をベースにしたものである。

同じ1987年10月、中国上海で第1回日中シンポジウムが開催されたが、その総括討論において、「ソフトウェア・エンジニアリングの実践状況の調査」を日中同時に行うことが合意され、調査結果の比較検討を行うことにより、日中相互の理解と技術交流を促進することになった。そこで1988年7月、ソフトウェア技術者協会の会員を中心に「ソフトウェア開発技術および環境の実態に関する調査」をアンケート形式で実施した。

本論文は、1988年に行われた調査結果をベースにして、1983年の調査結果との比較をおこなうことにより、現在のソフトウェア開発環境が、どのようなものであり、どんな技法やツールがどの程度利用されているかを明らかにするとともに、過去5年間にソフトウェア開発環境がどのように変化してきてか、また今後どのような課題があるであろうかを考察したものである。

論文構成としては、最初に1988年に行われた調査の回答者のプロフィールを紹介し、次に技法やツールがどの程度使われているかを1983年と1988年の調査を比較しながら概観する。更にソフトウェア開発の現場では、何が問題となっており、ソフトウェア開発の生産性を向上させるための要因は何であるかを、技術者/管理者それぞれの観点から探る。最後に、今後のソフトウェア開発環境構築の課題について述べる。

#### 2. 調査内容

# 2. 1 **調査方法・対象**

アンケート調査票をソフトウェア技術者協会の会員およびその周辺の技術者の方々に配布し、協力をお願いした。約2400通のアンケート用紙配布に対して、1ヵ月の調査期間中に寄せられた有効回答数は320通であり、回収率13%余りという好結果であった。

調査対象は、様々な会社・大学・研究機関に所属する人々に分布しており、特定の会社や機関に偏ってはいない。ただ、ソフトウェア技術者協会という組織の性格上、ソフトウェア・ハウスに所属する人々の比率は高い。

# 2. 2 回答者のプロフィール

# (1) 回答者の業種別分布

図2-1は回答者の業種別の分布を示したものである。ソフトウェア・ハウス、計算センタ、システム・ハウスといった所謂ソフトウェア産業に所属する人が全体の70%を占めており、特に、ソフトウェア・ハウスに所属する人が過半数を越えており最も多い。

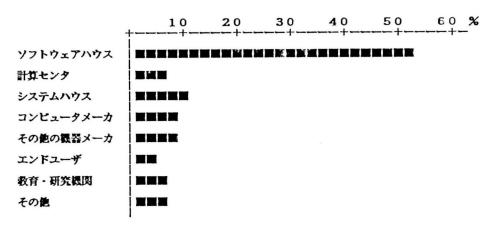
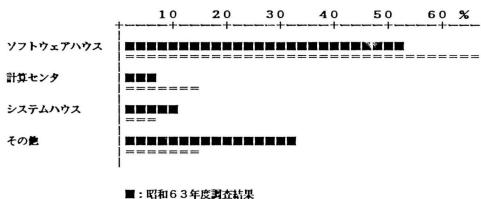


図2-1 業種別分布

5年前の1983年の調査の時は、ソフトウェア産業振興協会技術委員会主催のセミナーやシンポジゥム参加者約400名を対象としており、その時の業種別分類はソフトウェア・ハウス、計算センタ、システム・ハウスだけであった。図2-2は、今回の調査と前回の調査の業種別分類を比較したものである。



■:咱和03年及調宜輸来

=:昭和58年度調査結果

図2-2 業種別分布;前回調査との比較

前回調査では、全体の約2/3がソフトウェア・ハウスに所属する人々であり、その他に属する人は主として一般ユーザであり、全体の10数パーセントに過ぎない。今回の調査では、メーカや教育・研究機関、一般のエンド・ユーザは全体の1/3を占めており、ややソフトウェア業界以外の意見も反映されていると言えよう。

# (2)回答者の職種別分類

回答者の職種をジュニアメンバ(プログラマ/コーダ等)、シニアメンバ(SE/アナリスト等)、支援スタッフ(ライブラリアン等)、プロジェクトマネジャ、上位の管理者(複数プロジェクトを管理)に分類した結果が図2-3である。

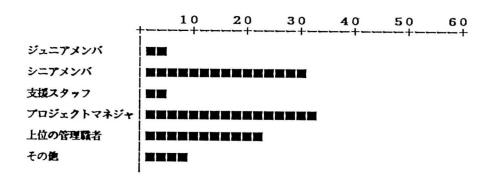


図2-3 職種別分類

これを、現場の開発担当者と管理職者の2つのカテゴリに分類(ジュニアメンバ、シニアメンバ、支援スタッフを開発担当者とし、マネジャおよび上位の管理者を管理職者と規定)した時、管理職者の比率が若干高い。前回の調査では、職種をプログラマ、システム・エンジニア、現場のマネジャ、上級マネジャに分類したが、プログラマとシステム・エンジニアを現場の開発担当者、現場のマネジャと上級マネジャを管理職者として分類し、今回の調査と比較した結果が図2-4である。

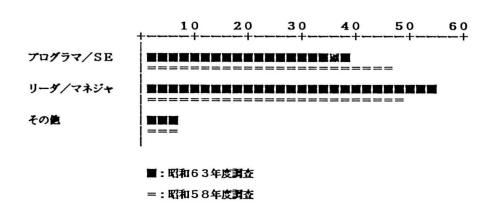


図2-4 職種別分類;前回調査との比較

今回の調査の方が10数パーセント管理者が多くなっており、管理職者の意見がやや強く反映されているということが出来よう。

## 3. ソフトウェア開発内容

#### 3. 1 開発対象システム

回答者がどんな種類のシステム開発に従事しているかを示したものが図3-1である。これは同時に、世の中でどんな種類のシステム開発が、どの程度行われているかを示すものであると言えるが、事務処理が最も多く、全体の1/3を占めている。また、開発支援ツールの開発に従事する人も20パーセント以上を占めており、かなり多い。ここに分類された以外の「その他」と答えた人も15パーセント程度と多いが、これは大学や研究機関での様々な研究開発活動が反映されているものと思われる。

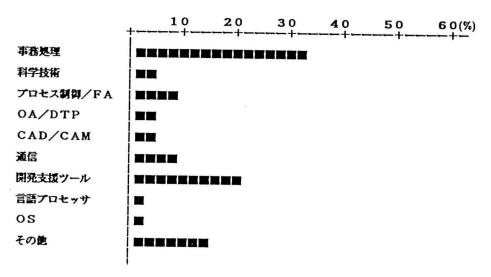


図3-1 開発対象システム

前回調査との比較のため、開発対象システムを「事務処理」「事務処理以外のアプリケーション」「システムソフト」の3つの分野に分けて比較する。図3-1に示された分類の内、「事務処理」はそのままであるが、「事務処理以外のアプリケーション」として科学技術、プロセス制御/FA、OA/DTP、CAD/CAM、通信を1つのカテゴリにまとめる。「システムソフト」として開発支援ツール、言語プロセッサ、OSを一つにまとめる。

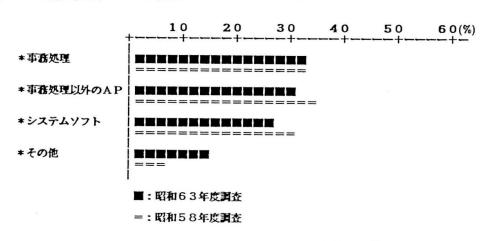


図3-2 開発対象システム;前回調査との比較

3つの大きな分野にまとめた結果を前回調査と比較したものが図3-2である。前回調査も、今回の調査も大きな相違は見られず、それぞれの分野の開発が約1/3づつを占めている。強いて相異を挙げるなら、今回調査で「その他」が増えた分だけ、「事務処理以外のアプリケーション」と「システムソフト」の比率が前回に較べ若干減少しているということである。

## 3. 2 使用言語

回答者が開発に利用している言語の集計結果が図3-3である。 C言語が圧倒的に多く、全体の40パーセント以上を占めている。 2番目はCOBOLであり、全体の約1/4を占めている。 FORTRANやASSEMBLERが意外と少ないことが注目される。

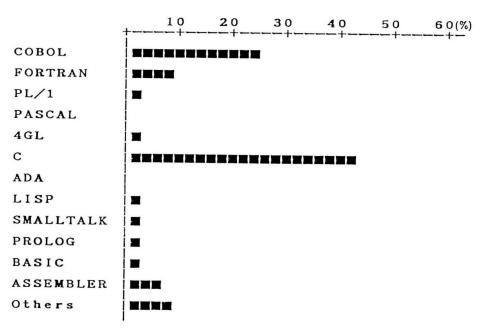


図3-3 使用言語

これを、5年前と比較したのが図3-4である。COBOLは若干減っているものの大きな変化はない。FORTRAN、ASSEMBLER、PL/1の大幅減少とCの大幅増加が特徴的である。これは、今までFORTRANやASSEMBLERで書かれていたプログラムがCで書かれるようになってきたことの証明であると思われる。特に、現在ワークステーションが非常な勢いで普及してきているが、これらを利用したソフトウェア開発の多くがCを利用していることを考え合わせると、C言語利用の比率の高さも納得できるものである。

では、この C 言語がどんな分野に多く使われているかを見てみよう。図 3-5 は、C 言語の利用を開発対象システムの分野別に集計したものである。1/3 以上が開発支援ツールの開発に利用されているが、プロセス制御/FA、OA/DTP、CAD/CAM、通信等でも幅広く利用されている。その他の分野での利用もかなり多いが、おそらく大学や研究機関で様々な実験/研究に利用されているものと推測される。

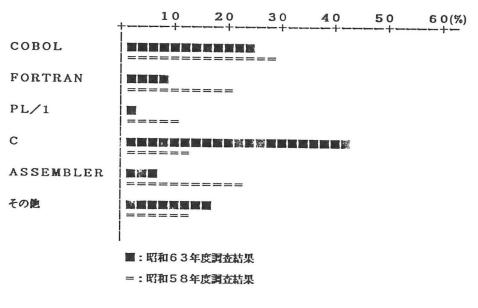


図3-4 使用言語;前回調査との比較

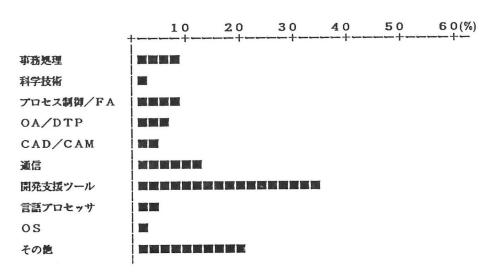


図3-5 C言語利用の分野

## 4. ソフトウェア開発技法/ツールの利用程度

一般的に言って、ある体系的なソフトウェア開発方法論に基づき、その実現を支援するものとしてソフトウェアの開発環境が存在するものとされている。そのソフトウェア開発環境上での具体的な諸活動を支援するものとしてソフトウェア開発技法があり、開発技法を具体化したものとしてツールが存在している.

しかしながら、個々の技法やツールは一つの環境や方法論に固定化されるものではなく、様々な使い方がなされている。従って、技法やツールの利用は、ソフトウェア開発のプロセスやプロセスモデルとの関連ではなく、単にソフトウェア開発における生産性の観点からのみ直結されて評価されていることが多い。

この一連の調査研究においても、技法やツールの利用が生産性と深く関わっているという視点から、その利用程度を調べたものである.

# 4.1 ソフトウェア開発技法の利用

図4-1は、ソフトウェア開発技法の利用程度を工程別に調べた結果である.

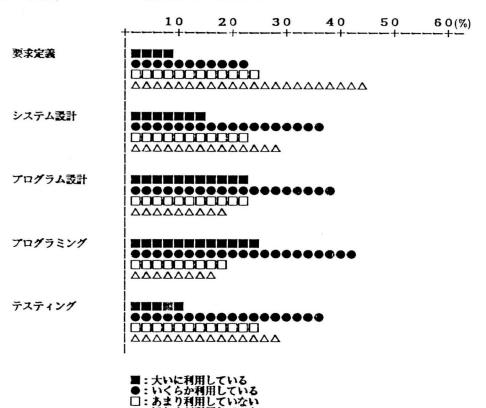


図4-1 開発技法の利用程度

□: あまり利用していない △: ほとんど利用していない

「大いに利用している」「いくらか利用している」「あまり利用していない」「ほとんど利 用していない」の4段階に分けて回答を求めたわけであるが、要求定義工程を除き何等かの形で 開発技法を利用している人が50パーセントあるいはそれ以上おり、かなり高い比率を示してい る. 「おおいに利用している」と答えた人に注目するなら、テスティングのフェーズでの利用が 意外と低いことが注目される.また、「いくらか利用している」という人に注目するなら、各工 程別にみて極端な開きはない。

それでは、これらの開発技法の利用程度をユーザ別に比較してみよう.すべての工程につい て表示することは出来ないし、各工程ともほぼ同じ傾向を示していることから、ここではシステ ム設計フェーズを代表として取り上げる.表4-1は、システム設計工程における開発技法の利 用程度をカテゴリ別に集計したものである。

まず、言語別にCOBOL言語の利用者とC言語の利用者についてみると、「大いに利用し ている」「いくらか利用している」人はCOBOLユーザの方が断然多い.現場の開発担当者と 管理職者とでは、「大いに利用している」と考えている人は管理職者に多い.汎用機のTSS、 ワークステーション、パソコンの利用者別では、TSSの利用者が最も多く開発技法を使ってい ることになっている.更に、開発対象ソフトウェア別に見れば、事務処理での利用が圧倒的に多 く、事務処理分野の開発に携わっている人の2/3近くが何等かの開発技法を利用しているとい う結果となっている.

カテゴリー	大いに利用	いくらか利用	あまり利用せず	ほとんど利用せぐ
全体	13.8	35.7	22.3	28.3
COBOLユーザ	17.9	44.8	13.4	23.9
C言語ユーザ	7.1	36.3	23.0	33.6
現場の技術者	7.8	35.9	29.1	27. 2
マネジャ	19.1	34.7	20.4	25.9
TSS利用者	19.8	36.5	24.0	19,8
WS利用者	8.2	32.9	28.2	30.6
PC利用者	13.6	27.3	13.6	45.5
事務処理	20.0	44.7	11.8	23.5
システムソフト	9.2	30.8	32.3	27.7
事務処理以外のAP	10.1	31.7	26.6	31.7

表4-1 システム設計における開発技法の利用

汎用機のTSSでCOBOL言語を使い、事務処理系のシステム開発を行っている人達が、比較的良くソフトウェア開発技法を利用しているということになる。このことは、ソフトウェア工学上の新しい開発技法が考案され使われだしたとか、ワークステーションを利用した新しい開発技法が広く普及し始めたとかいったことではなく、SEA/1とかEAGLEとかLINCといった、メインフレーマ提供の汎用機上での一貫支援ツールあるいは第4世代言語と呼ばれるものが比較的幅広く使わればじめたことに起因しているのではないかと考えられる。

### 4. 2 ソフトウェア開発支援ツールの利用状況

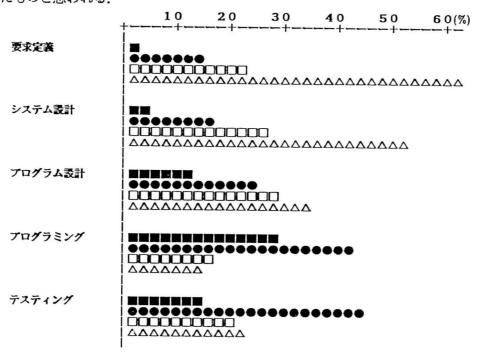
図4-2は、ツールの整備/活用状況を工程別に調べた結果である。「かなり整備/活用されている」のはプログラミング工程の30パーセントが最も多く、テスティングやプログラム設計工程では10数パーセントであり、要求定義やシステム設計フェーズでは、僅か数パーセントにすぎない。また、「いくらか整備/活用されている」工程もプログラミングとテスティング工程以外はかなり少ない。これは、開発技法がかなり利用されているという結果と比較すると意外な感じを受ける。

ッールの工程別利用に関して、5年前の調査結果と比較してみよう。図4-3は前回の調査で各工程別に「ツールが世の中に存在すると思うか」「実際に自分で使っているか」「将来欲しいと思うか」というの3つの質問に対する回答を集計した結果である。

若干分類方法が異なるが、プログラミングとテスティングについては、世の中にある程度存在すると思っているし、現に利用もされている。しかし、要求定義と設計工程に関しては、世の中にそんなにあるとも思われていないし、また実際の利用程度も僅か数パーセントにすぎない。即ち、ツール利用に関しては5年前も今も大きな違いはないと言っても良い。今回はツールのニ

# Survey Report

ーズに対する調査は行っていないが、もし、行ったとしたならば5年前と同様にかなりのニーズがあったものと思われる。



■:かなり整備/活用している ●:いくらか整備/活用している □:あまり整備/活用されていない △:ほとんど整備/活用されていない

図4-2 ツールの整備/活用状況

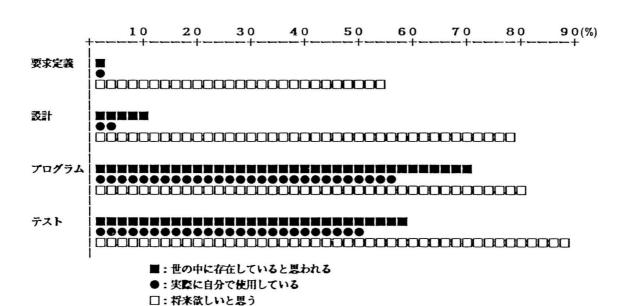
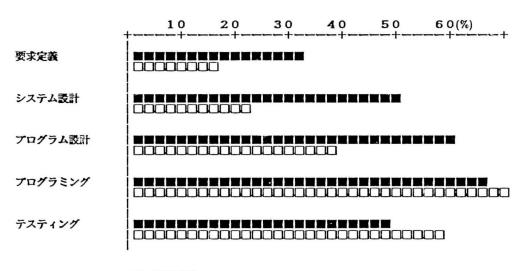


図4-3 前回調査におけるツールの利用程度

技法は利用されているが、ツールはほとんど利用されていないという結果になったが、工程

別に両者の利用程度を較べてみよう。技法については「大いに利用している」と「いくらか利用している」を合わせたもの、ツールについては、「かなり整備/活用されている」と「いくらか整備/活用されている」を合わせた数字で対比したのが図4-4である。プログラミングとテスティングについては、余り大きな相違はないが、要求定義、システム設計、プログラム設計についてはかなり大きな開きがある。これは、先に述べた一貫支援ツールとか第4世代言語といったものも、考え方とか方法は固まっているものの、具体的な開発支援ツールとしては未だ充分に機能していないということではないだろうか。



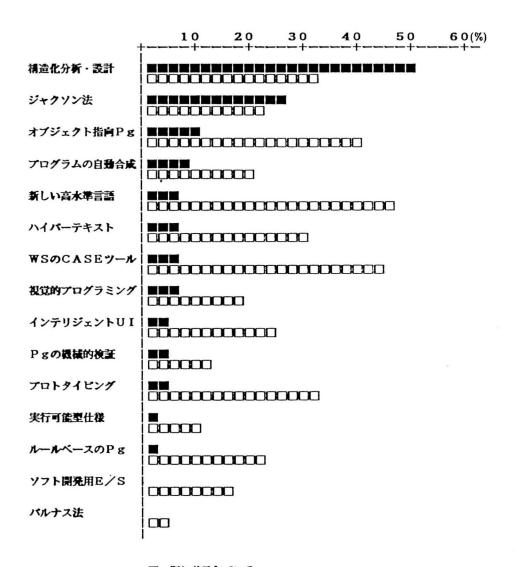
■: 閉発技法□: ソフトウェア・ツール

図4-4 技法とツールの工程別利用程度比較

#### 4. 3 技法/ツールの実用化および普及状況

図4-5は、いくつかの具体的なツールや技法に関して、それがどの程度普及あるいは実用化されているかを調べた結果である。構造化分析/設計は約50パーセントの人が既に普及していると考えており、ジャクソン法も約30パーセント弱の人々が既に普及していると考えているが、それ以外の技法やツールに関しては、普及していると思われているものは非常に少ない。特にAdaのような高水準言語、ワークステーションのCASEツール、オブジェクトオリエンテッドなプログラミング等は、半数近くの人々が実用段階ではあるが普及はこれからだと考えている。

これら、わずか数パーセントの人が普及していると考えている、技法やツールは、これからの普及にどの位時間がかかるのであろうか。過去の技術移転の経緯を見るに、おそらく10年単位という長いレンジで徐々に普及していくのではないだろうか。コンピュータ社会の技術的進歩は非常に急激であり、日々どんどん進化しているように言われているが、技法やツールに関して言うならば、実用/普及段階にまで達するのには、かなりの年月を必要とする。



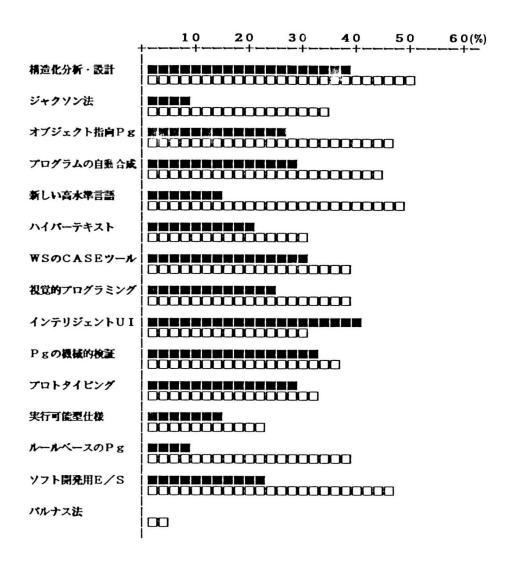
■: 既に普及している

□:実用段階だが、普及はこれから

図4-5 技法/ツールの実用化と普及

# 4. 4 技法/ツールの有用性の評価

それでは、これら技法やツールは、どの程度有用だと考えられているのであろうか。有用性について聞いた結果が図4-6である。ある程度有用であるというものを含めれば、ここに挙げた技法やツールは、かなり有用だと考えられている。特に、構造化分析・設計、インテリジェント・ユーザ・インタフェース、プログラムの正しさの機械的検証、ワークステーション上のCASEツール、プログラムの自動合成、プロトタイピング等は30パーセント前後の人々が極めて有効であると思っている。



■: きわめて有用だと思う
□: ある程度は有用であろう

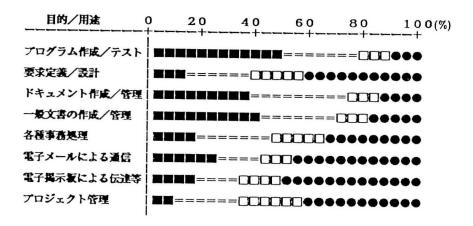
図4-6 技法/ツールの有用性

#### 5. ソフトウェア開発支援環境の現状

### 5.1 開発支援環境の目的・用途

図5-1は、回答者が日常利用しているコンピュータに基づく開発支援環境の目的/用途と、その利用程度を調べた結果である。

「プログラムの作成/テスト」に最も良く使われているのは当然のこととして、「開発関連ドキュメントの作成・管理」「一般の文書や手紙などの作成・管理」にも比較的良く使われている。「電子メールによる通信」や「電子掲示板による情報収集/伝達」も、1/3程度の人々は利用しているものの、残りの2/3の人々は余り使っておらず、半数以上の人は、殆ど利用していないと答えている。



■:大いに使っている =:いくらか使っている □:あまり使っていない ●:ほとんど使っていない

図5-1 開発支援環境の目的と用途

それでは、これらの目的/用途のうち、比較的良く使われている「一般の文書や手紙などの作成・管理」と「電子メールによる通信」に関して、利用者のカテゴリ別に分類した結果を見てみよう。表5-1は、一般文書や手紙の作成・管理について、まとめたものである。

表5-1 文書作成における利用程度

カテゴリー	大いに利用	いくらか利用	あまり利用せず	ほとんど利用せず
全体	40.4	29.6	10.4	19.6
COBOLユーザ	15.6	22. 2	15.6	46.7
C言語ユーザ	49.5	34.1	3.3	13.2
現場の技術者	27.4	33.3	14.3	25.0
マネジャ	44.8	25.7	9. 5	20.0
UNIX	48.7	33.3	7. 2	10.8
MS-DOS	50.0	30.0	6.7	13.3
メーカの独自OS	26.8	24.4	17.1	31.7
TSS利用者	26.7	31.4	13.3	28.6
WS利用者	52.1	27.1	9.4	11.5
PC利用者	53.6	32.1	3.6	10.7
事者处理	21.8	23.6	14.6	40.0
システムソフト	51.7	30.0	10.0	8.3
事務処理以外のAP	36.2	39.79	12.1	12.1

使用言語でみると、C言語ユーザの約半数が「大いに利用している」と答えているのに対し、COBOL言語ユーザは、僅か15パーセントが利用しているに過ぎない。また、OS別に見てもUNIXやMS-DOS利用者は約50パーセントの人が、「大いに利用している」のに対し、メーカの独自OS利用者は、約25パーセント強が「大いに利用している」と答えているにすぎない。これは、ワークステーションやパソコン利用者とTSS利用者と比較しても、ほぼ同様なことが言える。即ち、これらの文書作成や管理のための環境は比較的良く使われているとはいえ、ワークステーションやパソコンでUNIXやMS-DOS上のC言語を使って開発に従事している人々が中心であることがわかる。

一方、表5-2は電子メールによる通信に関して同様にまとめたものである。大いに電子メールを利用しているのは、ワークステーションのUNIX上でC言語を使って開発に従事している人々であり、汎用機のTSSを利用してCOBOLで事務処理を行っている人々の利用比率は極めて低い。

カテゴリー	大いに利用	いくらか利用	あまり利用せず	ほとんど利用せず
全体	23.4	14.3	11.7	50.7
COBOLI-#	2. 2	4.4	11.1	82. 2
C言語ユーザ	33.7	17.4	12.0	37.0
現場の技術者	25.0	17.9	13.1	44.1
マネジャ	19.8	9.4	12.3	58.5
UNIX	39.3	18.8	9.8	32.1
MS-DOS	13.3	6.7	10.0	70.0
メーカの独自OS	6.1	12.2	13.4	68.3
TSS利用者	13, 1	12.2	14.0	60.8
WS利用者	39.6	18.8	9.4	32.3
PC利用者	7.4	7.4	11.1	74. 1
事畜処理	7.1	5.4	14.3	73.2
システムソフト	35.0	18.3	10.0	36.7
事務処理以外のAP	15.0	17.2	15.5	48.3

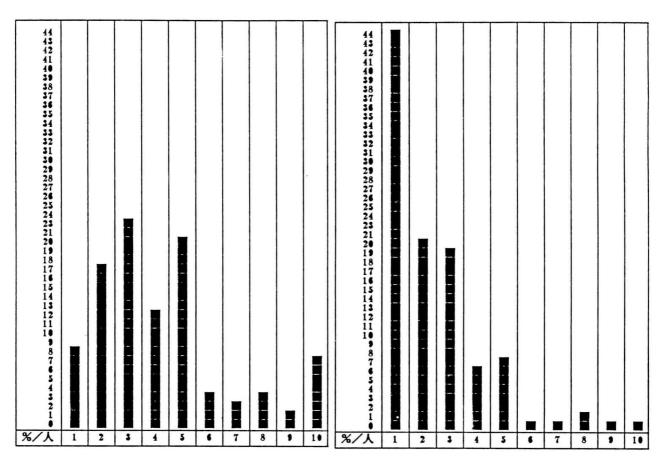
表5-2 電子メールの利用程度

#### 5.2 ターミナルの利用状況

端末/ワークステーション/パソコン等のターミナルを1台あたり何人の人が利用しているか調べた結果が図5-2である。左が5年前の1983年に調査した結果であり、右が今回の調査結果である。前回調査では、1台のターミナルを平均約6人で使用していたものが、今回の調査では1台平均約2人であり、ターミナルが非常に普及していることを窺わせるとともに、1人一台のターミナルの利用比率が最も高くなっていることに驚かされる。

技法やツールといった所謂ソフトウェアの利用/普及は比較的緩やかであるのに対し、ハードウェアの普及は非常な勢いでなされているということが出来よう。また、逆にこのハードウェアの普及活用が、恰もソフトウェア開発環境が進展しているという錯覚を与えていることも事実

であろう。ハードは揃えたがソフトが追い付かないというのが実情だと思うが、ハードがなければソフトの活用も思うように行かないことを考えれば、傾向としては歓迎すべき方向に進んでいると言えよう。



1983年度調査結果

1988年度調査結果

図5-2 ターミナル1台あたりの使用人数

開発対象システム別に、前回調査との比較をしたものが図5-3である。事務処理分野では、4人に1台であったものが3人に1台になった程度であるが、システムソフトウェアや科学技術計算等のその他のアプリケーション開発においてターミナル1台あたりの使用人数が大きく減少しており、その結果全体として1台あたりの使用人数が減少している。おそらく、エンジニアリング・ワークステーションと呼ばれる32ビットワークステーションの普及が大きく影響しているものと思われる。

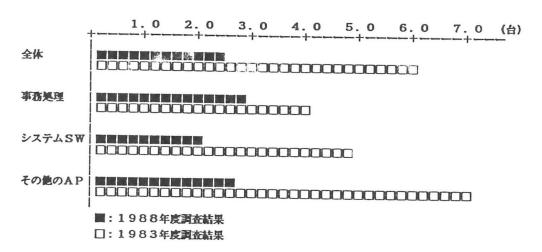


図5-3 分野別使用人数の前回調査との比較

## 5.3 環境のホスト・マシン

回答者の方々が利用している環境のホスト・マシンが何であるかを調べた結果が図5-4である。汎用メインフレームと32ビットワークステーション、およびそれ以外がほぼ1/3づつ占めている。前回調査では、ワークステーションについては質問していないが、汎用機については大型/中型/小型併せて約50パーセントを占めていたことと較べると、全体として汎用メインフレームの利用比率が相対的に低下してきていることは事実である。ただ、分野別の利用状況から明らかなように、事務処理においては汎用メインフレームの利用は70パーセント近くなっており、圧倒的な優位を誇っているし、それ以外の分野でも20パーセント前後は使われており、この状態が急激に大きく変化していくことは考えられにくい。

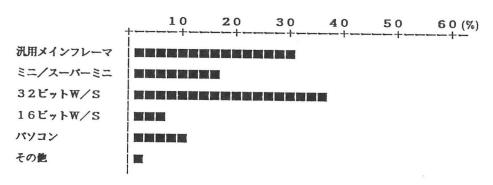


図5-4 環境のホスト・マシン

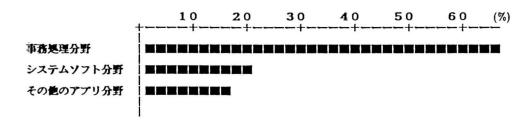


図5-5 分野別メインフレーマ利用状況

やはり、大量のデータ処理や高速かつ複雑な計算を要する業務には汎用機が最も適しているであろうし、個人ベースの分散開発環境構築にはワークステーションが適しているであろうことを考えると、今後はそれぞれの業務に最も適した機種が使われていくようになっていくであろう。即ち、機種毎の専門化が進んでいくものと思われる。

# 5. 4 ネットワーク接続

現在使われている環境が、どの程度ネットワークに接続されているかを示したものが図5-5である。まったくネットワーク接続されていない環境は、約30パーセントであり、それ以外は何等かの形でネットワーク接続がなされている。

前回調査の時は、LAN接続されている比率はわずか7パーセントであり、接続したいが約60パーセント、5年後には接続したいが約15パーセントであったことを考えると、LANの普及は目覚しいものがある。

また、LANの外部とのネットワーク接続にしても、前回調査では、接続されている比率はわずか1パーセントであり、接続したいが38パーセント、5年後には接続したいが17パーセントであったことから考えると、かなりの早さでLANの外部への接続が実現されていったことがわかる。

これは、ワークステーションとUNIXの普及に大きく影響されているものと思われ、スタンドアロンでマシンを利用する環境は、今後はほとんどなくなってくるのではないかと考えられる。

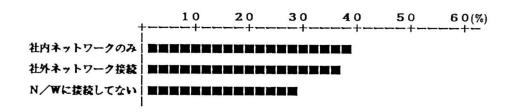


図5-6 ネットワークの接続状況

- 6. ソフトウェア開発における問題点
- 6.1 現状の問題点

ソフトウェア開発における問題点と思われる項目を20 個挙げ、最も問題であると思われる項目を3 項目を3 項目である。一番の問題は人の問題であり、開発要員が質量ともに不足していることや技術者教育が不充分であることが上げられている。また、方法論/技法/品質等に関する技術の未整備も問題点の上位にランクされている。

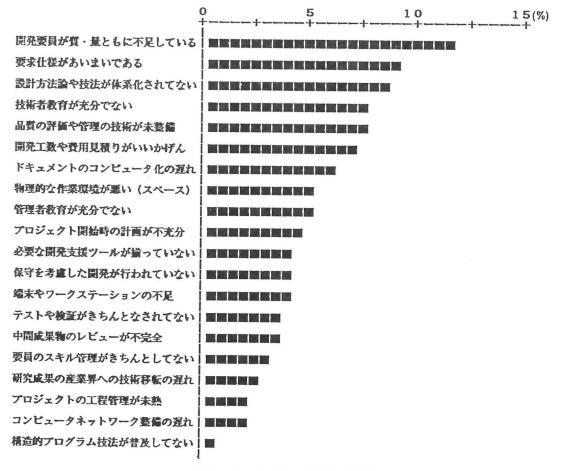


図6-1 ソフトウェア開発の問題点

これらの問題点を、ソフトウェア開発現場の技術者と管理職者の2つのカテゴリに分けて集計した結果が図6-2である。まず、現場の技術者の立場から最も問題とされるのは、要求仕様があいまいであることであり、2番目が設計方法論や技法が体系化されていないことである。3番目が技術者教育が充分でないことがあげられ、4番目として開発要員が質・量ともに不足していることが挙げられている。

このことは、ソフトウェア開発における所謂上流工程と呼ばれる、要求定義/設計工程において有効な支援方法が未だ存在していないということを如実に物語っている。また、技術者の質的な不足、技術者教育が不充分であるという指摘の多さは、ソフトウェア業界においては、技術者に対する教育というものがまだかなり遅れているということではないだろうか。

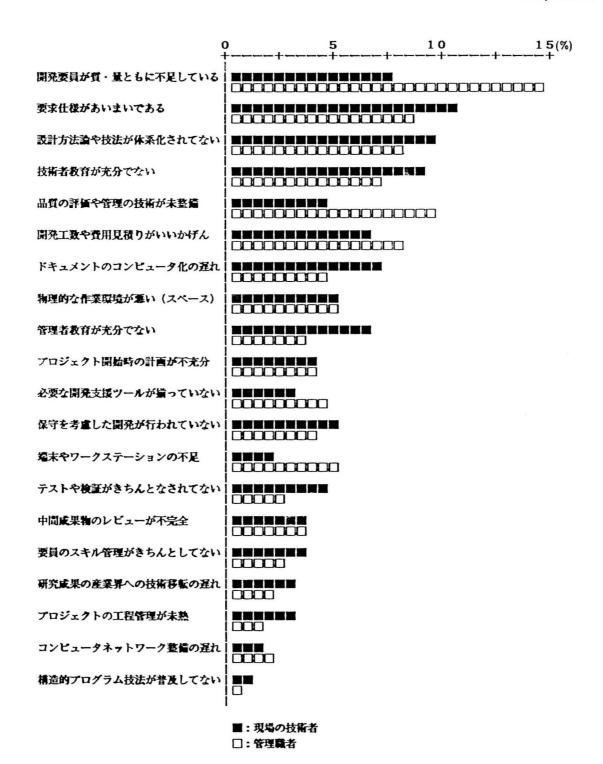


図6-2 カテゴリ別のソフトウェア開発の問題点

一方、管理職者が指摘する問題点は、開発要員の質・量ともに不足してことが圧倒的に多く、2番目として、品質の評価や管理技術が未整備であることが上げられている。仕事は入ってくるが、それをやらせる人の手当がつかない。特に、プログラミングだけなら何とかなるが、システム全体の開発を任せられる優秀な人材がいないというのが現状であろう。更に、何とか人の手当

をしたものの、それらに対する管理技術というものは整備されていないため、スケジュール通り に仕事が進まない上に、出来上がった成果物に対する品質のチェックもきちんと行われていない ため、常にトラブルに対する不安を抱えているということである。このことは、ソフトウェア保 守に多くの人を投入せざるを得ない結果ともなり、それが更に新規開発への人の投入に制限を加 えているといった悪循環を繰り返すことにもなっている。

# 6.2 生産性向上の要因

ソフトウェア開発における生産性向上の要因として、重要と思われる項目を3つ、あまり重要でないと思われる項目を3つ、それぞれ上げてもらったものを集計した結果が図6-3である。圧倒的に多いのが「開発要員の質的レベルの向上」である。以下「一貫した方法論に基づく統合的支援環境の整備」「要求分析・定義を支援するツールの整備・拡充」「設計支援ツールの整備・拡充」「テスト支援ツールの整備・拡充」と続いている。

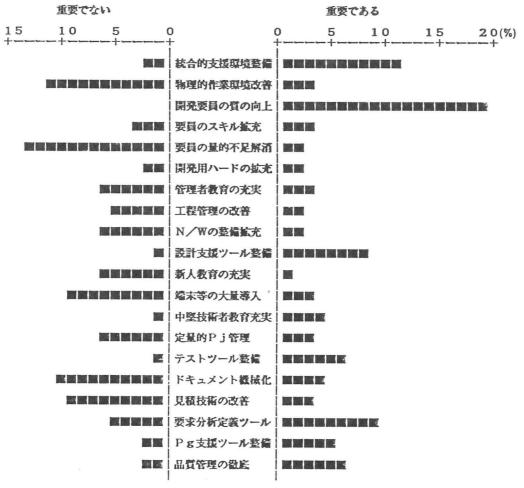


図6-3 生産性向上の要因

前節でソフトウェア開発における問題点として、「開発要員が質・量ともに不足している」 が最も多く指摘されていたが、この生産性向上に関する設問において、「開発要員の質的レベル の向上」が重要な要因のトップを占めているのに対し、「開発要員の量的な不足の解消」が、あまり重要でない要因のトップを占めるという非常に対照的な結果が出ている。このことは、問題点として上げられた「開発要員の質・量の不足」は、実は量の問題ではなく、質の問題が最も重要視されていたということである。

また、あまり重要でない要因として「オフィス・スペースなど物理的な作業環境の改善」「端末やワークステーションの大量導入」「ドキュメントの完全なコンピュータ化」等が上げられていることは、物理的な「物」の豊富さは、生産性向上とは余り関係ないということを示している。

ここで上げられたことを素直に解釈していけば、「ソフトウェアの生産性を向上させるためには、質の良い開発要員を使って、一貫した方法論にもとづく統合的支援環境のもとで、それぞれの工程に合ったツールを利用して開発を行うこと」ということになりそうである。逆に、「余裕のあるオフィス・スペースに、大量の人と、大量の端末やワークステーションを投入して開発を行い、ドキュメントの完全なコンピュータ化を図り、見積技術の改善を図っても、それだけでは生産性の向上にはならない」ということのようである。

表6-1は、前回調査においてソフトウェア生産性向上の要因として上げられたベスト8を示したものである。前回は開発要員の質とか量については聞いていないので、一概に比較することは出来ないが、「工程の標準化」を「一貫した方法論にもとづく統合的支援環境の整備」と類似な要因と考えるなら、その時々における人数の多少はあるものの、生産性向上の要因として考えられていることに大きな変化はない。

生産性向上の要因	人数(人)
テスト支援ツールの充実	94
工程の標準化	82
プログラミングツールの充実	80
設計支援ツールの充実	6.5
要員の管理	46
中堅技術者教育	32
プロジェクト管理のためのレビュー	3 1
TSS端末の増加	7

表6-1 前回調査における生産性向上の要因

## 7. 今後の課題

現在のソフトウェア開発技法/環境の現状について、前回の調査と比較しながら述べてきたが、それらを踏まえて今後の課題を探ることにする.

まず、5年前の調査と比較して顕著なことは、

- ・C言語の利用が大幅に増加したこと
- ・ワークステーションが急激に普及したこと
- ・ネットワーク(特にLAN)が幅広く普及したこと

を上げることが出来る。一方、ある程度は利用され始めたり、普及し始めてはいるものの、大きな変化が認められないこととしては、

- ・技法やツールの利用状況
- ・生産性向上に関する意識

を上げることが出来る。勿論、ワークステーションの普及に伴い、ドキュメントや文書の作成に、これらが比較的良く使われていること、また、ネットワークの普及に伴い、電子メールや電子掲示板も少数といえども使われていることは、技法やツールの利用方法が変化してきていることを示すものである。

しかしながら、これからソフトウェアの分散開発環境が急激に進むかと言えば、「YES」とは答えにくい。特に事務処理分野のソフトウェア開発において、汎用機上でCOBOLを利用した開発は約1/3を占めており、この状態が急激に変化していくことはないと思われる。更に、実際に稼働するシステムは汎用機上のものが多く、開発だけをワークステーション上に移すということは、ワークステーション上の環境が整備され、汎用機の環境と統合化される必要がある。しかし、現実の技法やツールの利用状況からみるに、その実現のためには、かなり長い時間がかかるものと思われる。従って、これから作成されていくシステムが、ワークステーションとネットワークを利用した分散処理システムとして作成され、大型の汎用機との間で機能分担とでも言うべきものが、ある程度明確に行われるようになる必要がある。

現在の流れから推測して、今後ワークステーションやネットワークを利用した分散開発環境が、普及していくことはほぼ間違いない方向とはいえ、より良いシステム化のためには、次の事項が容易に行えるようになる必要がある。

- ・ホスト・コンピュータとLAN環境の統合化
- ・異機種のネットワークへの容易な接続
- ・特に開発の上流工程に的を絞った技法/ツールの利用と普及
- ・分散開発環境上におけるソフトウェア開発管理方法の確立

これらは、既に一部実現されているものもあり、近い将来においてかなりの部分が解決されていくものと思われる。ただ、やはり最後に残るのは、技法とツールの利用と普及に関する問題ではないかと思われる。

## 8. おわりに

丁度5年間という区切りの良い時期に、ソフトウェア開発環境の現状に関する調査を実施し、 それなりの変化を捉えることが出来たことは、幸いなことであった。同じ人に対して、ほぼ同じ 質問を行うことが出来たなら、もう少し変化の状況を的確に捉えることが出来たかもしれない。

更に5年後の1993年に、今回と類似の調査を行い、どのように変化していったかという 観点から比較が出来たなら、非常に面白い結果が得られるのではないかと思うし、時系列的な変 化を捉えた貴重なデータとなるものと思われる。

最後に、このアンケート集計の分析ワーキング・グループに参加し、貴重な御意見を述べて下さった、岸田さん・熊谷さんを始めとするメンバの方々、機械集計のためにデータ入力に協力して下さったSRA専務室の秘書の方々、および、このアンケート調査に積極的に回答して下さった多数の方々に、この紙面を借りて感謝の意を表します。

## 【参考文献】

- 1. ソフトウェア開発環境の改善に関する調査研究報告書,
  - 1984.3, [(社)ソフトウェア産業振興協会]
- 2. 野村;"ソフトウェア開発環境の現状" ソフトウェアシンポジゥム84論文集, 1984.6, [(社)ソフトウェア産業振興協会]
- 3. 野村;"ソフトウェア開発環境" IPA技術センター 第3回発表論文集, 1984.10, [情報処理振興事業協会]
- 4. ソフトウェアツールデータベース構築に関する調査研究報告書,1985.3, [(社)情報サービス産業協会]
- 5. ソフトウェア開発の生産性向上のための開発環境構築報告書, 1985.3, [(財)産業研究所]
- 6. Nomura; "Software Engineering Tools in Japan" 1986.8, Proceedings of the International Workshop on Software Engineering Environment, [Beijing China]
- 7. 野村;"日本におけるソフトウェア開発環境の現状" 1986.10, [長野県情報サービス振興協会]
- 8. Nomura; "Use of Software Engineering Tools in Japan" 1987.3, 9th ICSE Proceedings, [Monterey California]
- 9. Nomura; "Survey on Software Development Technology and its Support Environment" 1988.10, 2nd China-Japan Software Symposium, [Hangzhou China]

# あなたにとって理想の開発環境は?

# 会員およびセミナー参加者への一口アンケート

#### 0. はじめに

この春、全会員あてに、いま利用している開発環境および理想の開発環境についての一口アンケートを実施しましたが、結局20通ぐらいしか集まりませんでした。そこで、3月のフォーラム(プロトタイピング)および春のセミナー・ウィークの受講者の方々にも御強力をお願いし、最終的に147通の回答を得ました。以下にその集計およびみなさんから寄せられた自由回答のコメントを収録しました。

#### 1. 現在使っている環境

現在どんなマシンを使って仕事をしているかについての 回答は、次の表1に示す通りです.

W S	P C	T S S	W P	その	人	比
		S		他	数	率
0					30	21.9%
0	0				24	17.5%
	0				16	11.7%
0		0			12	8.8%
	0	0			9	6.7%
	0		0		8	5.8%
0	0	0			8	5.8%
		0			7	5.1%%
0	0		0		6	4.4%
	0	0	0		3	2.2%
0		0	0		3	2.2%
		0	0		3	2.2%
				0	3	2.2%
0	0	0	0		2	1.5%
0	0		0	0	1	0.7%
0			0		1	0.7%
	0	0	0	0	1	0.7%

表 1 利用環境

ただし、WS はワークステーション、PC はパソコン、TSS は TSS 端末、WP はワープロです.

したがって、それぞれの種類のマシンの保有率は表2のようになります。また、2種類のマシンの組み合わせの状況は、表3の通りです。

種類	人数	比率
WS	87	63.5%
PC	78	56.9%
TSS	48	35.0%
WP	28	17.5%
その他	5	3.6%

表2 各マシンの保有状況

組み合わせ	人数	比率
WS & PC	41	30.0%
WS & TSS	25	18.2%
PC & TSS	21	15.3%
PC & WP	20	14.6%
WS & WP	13	9.5%
TSS & WP	12	8.8%
WP & その他	2	1.5%
PC & その他	2	1.5%
TSS & その他	1	0.7%

表3 2種類のマシンの組み合わせ

# 2. 環境を何に使っているか?

次に、そうした開発環境を使ってどんな仕事をしている かですが、集計結果は、表4のようになります.

用途	人数	比率
プログラム開発	63	46.0
ドキュメント作成	46	33.6
管理/レポート作成	43	31.4
一般的文書処理	37	27.0
デバッグ/テスト	31	22.6
環境設定/ツール開発	19	13.9
電子メイル/ニュース	15	10.9
技術調査/ソフト評価	13	9.5
保守	9	6.6
デモンストレーション	2	1.5

表4 環境の利用目的

ほぼ、予想通りの数字ですが、電子メイルやネットワーク・ニュースを読んでいる人が若干少ないのが気になります。まだ、分散環境はそれほど浸透していないということ

でしょうか?

#### 3. 理想の開発環境は?

以下は、みなさんから寄せられた理想の開発環境に関するコメントです.

横山 博司: 制御系の仕事をしていますが, (1)プログラムソース作成や修正, (2)コンパイル, (3)テスト,という3種類の作業が同一マシン上で行なえればよいと思います。テスト工程では,実機と ICE で接続されていることが必要です。さらに,設計ドキュメントからコードが自動生成されれば,いうことはありません。

**岡崎 博樹**: システムの仕様に関する情報とプログラム本体が常にリンクしていて、簡単に相互参照のできる環境.また、利用時に憶えなければならない事項が少ないほうがよい、物理的には、できるだけ大きくて、なおかつ長時間使用しても疲れないディスプレイがほしい。

藤原 利晴: まずWSがパソコンなみに使えること. 自分専用のWSを rootとして自由に使いこなしたい.もちろん,他にパブリックなWS群も何台かある.ソフト的には、基本OSが公開されており、パッケージ・プログラムとアプリケーションのインタフェースが簡単にとれるようになっていてほしい.現状では、開発時に、まず「どうすればできるのか?」あるいは「できるのかできないのか?」を調べる作業に手間取り、実際のシステム設計や具体化にとりかかるまでの道のりが長すぎる.このあたりを支援してくれるツールもほしい.

田村 整: [イメージ支援ドキュメントレス開発] 人間の考えをそのまま、キーボードを使わずに入力できる.また、入力結果はCGを用いて即座にディスプレイされる.さらに、入力を解析しドキュメントなしにプログラム (特にユーザアプリケーション)の開発が行なわれる.少々子供っぱくマンガ的な発想であるが、数年来いわれているSEの不足はなくなるのでは?(ひょっとすると我々も失業かもしれないけれど....)

宮下 高弘: 部品化、ディクショナリ化されたプログラムソースがあり、それを組み立てて行く要領でプログラムが生成できる。画面イメージは、ワープロのように線を引いたり、字を入れたりすれば生成される。専用の EWSを用いて Mac 風のウィンドゥシステムを操作しながら組み立てて行く。ちょっと疲れたかなと思うと、斉藤由貴に

似た女の子がコーヒーを入れて来てくれる. ナンテいいな a....

**漬篷 牟**: パソコンの操作環境が WYSIWYG (Mac 風) になること. 実用に耐えるパソコン用の日本語対応 DTP. そして, HDD 付ラップトップ・パソコン (これは 単にお金の問題か?).

**目黒 敏樹**: 開発するアプリケーション・システムの 要求定義・基本設計・詳細設計を端末から行えば、ユーザ 一への説明資料、システムのドキュメント作成、データ辞 書の作成、プログラムの生成まで自動的に行える(一言で いえば、コンピュータをアプリケーションの開発に使用す る).

**松尾 正**: どこに何があるか,だれがいつ何をしたか,今しつつあるか,何はどのようにして生まれ,だれに何をされて現在に至ったか,何と何はどのような関係にあるか等,物と行動と作用を,時間と空間の両面で一元的に管理できる環境.こうなれば,他人のドジに悩まされずにすむ.また,自分のやったことや,やっていることを客観的に評価できる.いま,何が起こっているか,これから何をやるべきかがはっきりする....

船切 誠: ディスク容量は無制限.複数(最低5本)のプロセスが同時に動作してもパワーが落ちない CPU. モノクロームのビットマップ・ディスプレイとポインティング・デバイス (マウスでなくてもいい). OS は何でもかまわない. 可能ならば, OS を変更できる (切り換え可能な)システムがよい. ネットワーク機能は必須. 社内外のデータベースやネットワーク・システムにアクセス可能.強力な文書処理システム. プログラムソースを見やすくするフォーマッタ. エラーの場所と修正を提示するコンパイラおよびエディタ. 実行時にエラーから自動的に起動するデバッガ. プログラムの状態を乱さないデバッガ.

土屋 旨夫: リカバリー機能のあるスプール・ファイルを高速で出力する.ジャムらないレーザプリンタを独占的に使用できて、要求分析書から、プログラムソース、テスト・レポートまで同一のエディタで編集でき、自分の機密は保護されて、しかも、他人の資源は最大限に利用できる.そんな環境を僕は望む.

**安西 宏**: バカデカイ WS の CRT 画面だけが薄い 板状になり、鞄に入れて運べること. 自然言語でそのまま プログラムが書けること.頭で考えたマクロな概念で機械を動かせること.職場,自宅,電車内,自動車内を問わず「板状」CRTからネットワークにアクセスできること.

**滝口 亨**: 作業場所がきれいであること. それぞれ目標と手段,達成状況,要員配置状況が機械の上で管理できること. 上記の情報を入れたマシンがカバンの中に格納でき,持ち運びできること.

**本多 清忠**: 空間がゆったりしていること(個人別の間仕切り). 休憩スペースがあること. ネットワーク+ DB のあること. 個人用の WS+電話+FAX. 要するに,アトリエ感覚で自由な発想ができる作業環境.

杉山 英俊: AP 開発環境のサポート. 特に, そこに おいて発生する大量のドキュメントの作成・保存・メンテナンス等をサポートするシステムが欲しい.

**岸本 剛一**: 空気がきれいなこと(禁煙である).人語のノイズレベルが低い.よいキーボード.すなわち,押しはじめの反発力が小さく(クリック感のあるのはダメ),ストロークエンド近くで大きくなり(ガツンとならない),かえりが速い(指についてくる).ネットワーク(部門内):ファイルサーバ,プリントサーバへのアクセス.ネットワーク(部内外):社内外のメール.

権嫌 敏朗: 個々の技術者に、コストパフォーマンスの優れたコンピュータ環境が提供され、自由かつ容易にネットワークに接続することができ、高度なコミニュケーションや、資源の共同/相互利用がはかれる、ソフトウェア開発の全ライフサイクルを支援するツールがあり、それらは自由にカスタマイズでき、充分に実用的であること、プログラミング環境は、ターゲットから独立していること、また、各種のプログラミング言語の相違による制約をうけないこと、ユーザー支援を含めた強力な保守支援環境があること、テクニカルな面だけでなく、技術者のメンタルな面を配慮した環境が必要だと思います。

藤野 晃延: 朝日と小鳥の囀りで眼を覚し、ブランチを食べたら、「さて仕事でもするか!」と CRT のスイッチを ON にする. BGM には Bohemians か何かを流しながら、画面上に CNN ニュースを写し見る. 同時に Mailと News をウィンドゥに表示し、返事しなければならないものに応答する. 今日の作業スケジュールをプロットしたウィンドゥを見て、少し変更を加える. CNN ニュースを

閉じ、報告書と提案書の概要をとりあえず作り、それに必要資料をリンクして、ハイパーテキスト(メディア)モードで書類を作って行く....

桜井 麻里: 自分専用に使える WS があり、社内のネットワークにつながっている。社内ネットワークは、ゲートウェイを通じて junet など外のネットワークにつながっていて、各種情報のやりとりができる。WS にしろツールにしろ、いつでも使いたいときに使いたいように使える環境が必要。どんなツールがどこにあるか、だれに聞けばよいかが、いつでもわかるようになっておいてほしい。WS のディスクのダンプをとってくれる人が他にいると、とてもラクです。日本語環境、マルチメディア環境も必須ですね。(それから、今日のフォーラムでの深瀬さんの話を聞いて、設計技法に関するパブリックなスタックウェアなどがあればとてもいいだろうと感じました)。

**鈴木 裕信**: ビジュアル+サウンドが優れたユーザインタフェース. 人間のアイデァを助けてくれるシステム(人工知能か?). ネットワーク上のプロセッサ,メモリ等を自由に使用できる分散システム. アーキテクチャレベルでの分散. いま入手できるマシンとしては NeXT が一番近い.

匿名: ホストとして大容量のファイルが取り扱え、ある程度の CPU パワーを持ち、高速のディスクおよび LAN があること. OS は、 Unix あるいはライクなものがのっている. LAN 上の各 WS が有機的につながること. たとえば、現在はイーサーネットと Apple Talk を使用しているが、このように複数の LAN が使えること. こまわりのきく WS を各メンバーに持たせることが望ましい.

西岡 健自: 同じ内容を二度以上入力しなくて済む. 一箇所直せばすべてに反映される. 日本語の入力が容易である. マルチウィンドゥ/マルチメディア自由自在. 在宅勤務可. プロジェクト管理/製品管理の情報は, 開発プロセスから自動生成(レポート不要).

**鈴木 隆**: 目標は,不定型なアイディアから,既存のアイデアで使えそうなモジュールをガイダンスするようなツール.新しいアイデアや手法は,自分で楽しく作りたい.

福本 英二: 結論からいうと、現在の2次元のディスプレーやシステム表現方法では、もの足りない(というか限界かな?)ので、3次元でシステムを表現する方法とそ

れを実現するツールが欲しい.また、1台の WS で要求分析からテストまですべてこなせる自分専用の環境が欲しい (ということは、現在そうなっていないということです).もっといえば、AI-WS なんかがあって、それと協力しながらシステムが作ればよいですね.

**岸田 孝一**: 画面の大きな Mac 風の WS が, 自宅と会社の両方にほしい. もちろん, ネットにつながっていて, いろいろな人との共同作業を強力に支援してほしいのだが, そのための分散型 OS がまだ十分に整備されていないのが残念. また, 旅行に出ることが多いので, もう少し使いやすいポータブル・マシンもほしい. 気の利いたビジネス・ホテルに情報コンセントが備わっていればいうことなし.

中村 暢夫: 販社ですので、パソコンから中・小型機のシステムを、色々な言語で開発しなければならない、開発は1ヶ所で行なって、それからそれぞれのターゲットマシンに変換できることが望ましい。仕様(機能?)のライブラリがあって、それを検索・組み合わせる形で、設計(プログラム仕様作成)を進める。仕様書ができたら、ソースを自動的に生成してくれる。後は他のだれかに送って、テストとか妥当性をチェックしてもらう(ネットワークでつながっている)、ドkュメントをどこで作るかが問題になるが、ソース生成する直前で、形式的記述がいるし、プログラムが完成してからも出力できないと困る。ソースではなくてももっと大雑把なものが作れるとよい。

**匿名**: プログラム作成を混然一体に行える環境.具体的には:プログラミングとデバッグを同時に行える,全部のコーディングをしないでも部分の検証が行える,ドギュメントやマニュアルも一緒に記録することができる,協同プログラム開発が可能な環境,といったこと.もちろん,マルチ・ユーザ環境であることは当然として,絵や音も一緒に扱えないといけない.

**堰口 馨**: ブロードバンド LAN によるデータ・音声・画像等の私有・共用化(1台の WS で、色々なメディアアクセス). 適切な情報を早くきれいに収集できる環境(自然言語データベース・流動的スキーマ). メモの電子化(思考・発想の過程の蓄積). 個人ゾーンとグループゾーンと区分けと合体(フレキシブル空間).

匿名: 生産物すべての電子化(仕様書,トラブルシート,連絡票等). 管理データの自動収集. 開発支援 DBの構築. ネットワーク化による遠隔地での共同開発作業の支

援. ソースコードからテスト項目の自動作成.

匿名: 分散開放型のシステム構成で、1人複数台のマシンを利用可能としたい. 使用マシンは、ビジュアルインタフェースであることを望む. また、A4サイズの携帯可能な DynaBook のイメージのマシンも、1人1台としてほしい。

匿名: 50万円以下で手に入る WS (機能は現在の200万円台のマシン並み)があったら欲しい. 現在のイーサネットの10分の1のコストで LAN が構築できるような道具か材料があればよい. 新規作成部分を50%以下に押さえられるようなライブラリキットがあって,かつそれは安い値段で手に入ればよいと思います.

匿名: EWSが1人1台. OAテーブルは充分に広く,作業が容易なこと. また各個人の作業空間は,低いパーティションで区切られ,かつ静かであること. 構造化分析/設計ツールおよび Lower CASE ツールが完備し,既存ソフトの再利用が容易である.

谷崎 隆士: 要求仕様からソースまでの関係が視覚的 にとらえられ,変更の影響が把握できる. またチェックリストとしても役立つ. 分散環境による資料の一元管理さらにはプロジェクトの進捗把握も容易である. 情報連絡には, Mail および News 機能を用いる.

匿名: WS 上に、ワープロ並みのエディタがあること (Nemacs に図形入力機能がついたものがほしい). マイコン搭載システムとのオンライン・デバッグを、WS からのコントロールで行なうことができるとよい.

大木 幹雄: 論文や報告書を熟読することができるように、ワープロやパソコンのない世界を望む. 発想をメモするノートと、文献検索や作業依頼ができる秘書のいる環境. もちろん電話は直接かかってくるようなことはない.

**岡本 吉史**: プロジェクト管理のための各種統計データを自動的に算出してくれる. プログラムのバージョン管理を自動的に行ってくれて, プログラマが注意しなくてよい. 使いやすいデバッガもほしい.

畑瀬 晃: 重複なく、でき得る限り多くのメンバが、同一の sheet、同一の format 上で作業できること.階層 属性を持つオブジェクトに関しては、そのオブジェクトの 階層位置、および人的組織上での位置が明確であること.

**中野 秀男**: 目の痛まないディスプレイ. Hyper-Media. 1人1台の WS. 世界即時ネットワーク. 30分~1時間以内に皆が集まれるおいしい居酒屋.

**匿名**: 人数に見合うスペースが十分に確保できるプロジェクトルーム. 考える作業のための個人用スペース(個室). 自分専用の WS.

片岡 賢治: 現状の環境に客先のマシン (PC, 大型 機

WS等)を連結させてのデータの有効活用.現在使用している汎用ソフトと WSとの関係をうまくマッチさせたい.

**仲野 孝幸**: ホストコンピュータと WS の役割を明確に切り分ける. WS 上でプログラム開発を行い,ホストはサーバーとして使うような分散型の環境を実現するのが理想です.

川島 浩: 速い・心が通い合う・静か・そしてでしゃばらないマシン. 仕事の context も理解してほしい.

匿名: 多種多様の開発プロジェクトを抱えているため、 人的要員を戦略的に配置できるような柔軟的ローテーションを考えた教育が望ましい. 基礎調査からテストまでを一環して支援してくれるシステム的環境.

**佐原 伸**: 遊びも含めて, すべての作業をハンドヘル ド型の WS でできること. ソフトは Unix + Smalltalk + Mac の融合したもの.

**匿名**: 広い机と2台のキャラクタ・ディスプレイまたは PC.

**渡辺 美樹**: ターゲットシステムの実行環境を意識せずにソフトウェアの開発が行えるような環境であってほしい.

**柳下** 真琴: ほしいときにほしい資料がすぐ検索でき, それを見ながら,または流用しながら開発できるようなシ ステム.

**匿名**: 自由に日本語が使える専用ターミナル.ディスプレイ平面で角度自由,大きい(A4×3面以上).

匿名: ハード面では、EWS 1台1人、OA テーブルが広いこと. 一方、ソフト面では、CASE ツールが完備していること.

**吉田 勇一**: 仕様書とソース・コードを区別する必要がないような環境.

佐藤 由美子: LAN 等で連結されたワークステーション上で、プログラミング開発ができ、異ホスト間でのデータ伝送等が可能である.

**鈴木 雅哉**: 大型マシンの端末兼ワープロ機能を持つ WS が、1人1台ずつ各自の机にあり、システムの開発状況等がいつでも文書に編集される.

匿名: 人間の思考を中断しないインテリジェントな MMI を持つ WS による分散ネットワーク環境.

匿名: 大型ホストを WS 環境から使用したい.

岡本 健二: プロセスプログラミングの実現.

佐藤 賢司: 静かで熱の放出の少ない WS と静かな 部屋と興味のわく仕事.

大崎 正巳: Mac みたいにユーザフレンドリーな環境.

**玉村 元**: データベースによりソースコードその他の 資料がすぐ検索できること.

匿名: 1人1台の端末がほしい.

伊東 直幸: 1人に1台ずつ PC または WS があり、 ソフトウェアの設計、製造からドキュメント作成までを一 貫して作業することができる. ネットワークを通じて、連 絡や簡単な打ち合わせができる.

田中 和夫: 計算機環境は、いうまでもなく1人1台以上の WS があることが望ましい。また、外乱を避ける環境(電話その他の用件でインタラプトされない環境)にするための工夫も欲しい。新しい技術を拒むことなく導入する。新しくよいものを使わないと新しい発想は生まれない。

匿名: PC 等は、1人1台の時代になりつつあると思われます。仕事に集中したい時には、1人で個室に閉じこもれるようになっているとよい。もちろん、コミュニケーション・メカニズムは忘れずに。

匿名: 1人1台の WS. 水平分散&ホスト連係. 現状の OA ツールはあまりに貧弱だから,当然改善される必要がある.

**匿名**: 思考段階での対話型の補助ツールが、最終的な姿か?

匿名: Unix WS を使ってプログラム開発をしたい!

**匿名**: 大型ーミニー PC のネットワーク.

**岡芹 晃:** 当面できそうな範囲は、仕様書・設計書・ソースをすべて DB 化して管理し、共有や再利用を容易にすることであろう。そのさい、あまり大げさな仕掛けはよくないので、PCのLANにサーバーを付けた構成とする、ツールとしては、一太郎・花子など既存ソフトをベースに、それらをカスタマイズした形を考える。理想をいえばもっととんでもない夢はあるが、まず現状の改善から進めて行かざるを得ない。

匿名: 以前は研究開発部門にいて、SUN を使用していたが、現在は情報システム部でTSO端末を使っている。 TSO は今年から使いはじめたのだが、WS と比べて、あまりに落差が激しい。TSO 端末のかわりに簡単に WS がつなげればよいのだが....

匿名: デスクが広いこと. 資料をある程度ストックできるスペースがあること. 端末エミュレーション, ワープロ, 表計算などの機能をそなえた PC があること. グループ内の他メンバーと会話が自由にできる配置となっていること.

**匿名**: 余裕のある開発スケジュール,経験豊かなメンバ,充分な人員.

**匿名**: MS-DOS および OS/2 環境で、ハードディス クが 100M,メモリが 6M以上、かつ何台かに1台レーザ ープリンタのついた 32bit の PC (オアシスとのフォーマット変換ユーティリティーも必須).

**匿名**: 現在,社内に支援環境が充実していないので、 客先で作業を行っているが、客先にあるターゲット・コン ピュータに、社内の PC を端末として使用できるようにし たい。

**野見山 和則**: Apple が考えている Knowledge Navigator があれば、それだけで満足すると思う。たぶん?

**匿名**: プロジェクトの目的をメンバ全員が把握しており、不明点はすぐに解決できる体制.

**埼野 直樹**: 各種のツールを統合している環境のファクタを使用者がコントロールできるようになっていてほしい.

内藤 求: SDLC すべてを支援する CASE 端末が近くにあること.

**鈴木 芳郎**: 楽しく,人を常に感動させられるのではないかという可能性を持った環境.

**矢吹 節史**: 開発環境のしっかりとしたネットワーク 化.

匿名: 1人1台の端末,データは集中管理.

**匿名**: 1人1台 WS または PC または端末. 管理者 用(数値データ把握・分析のため)マシンの設置. (パソコン)ネットワークの充実.

**匿名**: なるべく, 言語やターゲット・マシンに依存せずに開発作業が進められるような環境であってほしい.

**匿名**: 充実したプロダクツ (ツールや部品) が用意されていて、その検索機構が簡単かつ高レスポンスであること、また、各プロダクツ間のインタフェイスの整合性が簡単にチェックできること、またはすでに統一されていること、

# プロジェクト計画・管理そしてレビュー

# 春のセミナー・ウィーク参加者アンケートから

1. これまで、プロジェクト計画や管理の定量化に関して、いつも心掛け実践してきたことは?

内藤 求: 開発手順を決め、開発工程を細かい作業に分けて、そこでの成果物を定義する. 各作業での成果物と工数を集計し、見積り、生産性向上の目標にしている.

**大久保 功**: 各開発部門から、ソフトウェア製品の出荷 後の不具合情報を集め、過去と比べてよくなっているかど うかをチェックしている。

矢吹 節史: プロジェクト構成員全員の日報提出,プロジェクト内会議及びレビュータの実施,プロジェクト完了報告書の作成,プロジェクト構成員との個人面接実施.

**齋藤 孝行**: 今般,新しく品質管理部門を預かり、ソフトウェア開発工程での品質保証体系作りに着手したところです.プロジェクトリーダーへの「品質意識」の植え付けからスタートして行くつもりですが、現在は品質の「定量化」について工夫をしています.

**片岡 賢治**: 開発ステップ (想定)を軸として,配員・スケジュール管理を行なう.スケジュールを中心としたメンバー管理と,機械化によるデータ量の管理.

**岡本 健二:** FP (ファンクションポイント) 実績収集 の機械化. ただし, 現在問題意識を持っているのは, 要求 分析等の基本計画段階の見積りの定量化である.

村田 幸夫: プログラム規模見積り、プロジェクトスケジュール (線表) 上での進捗管理、バグ件数の集計・分析、

匿名: 開発担当者が自分で見積りすること.

**匿名**: プログラムの作成キロステップおよび評価キロステップの管理,マイルストーンの管理.

匿名: 見積り工数と実績の定期的な突き合わせ.

**匿名**: 無理のないスケジューリングと客先との交渉,経験豊かな人員の獲得,正確な見積り.

匿名: 採算管理,見積り.

**匿名**: 各作業単位で工数を出し、何に時間がかかっているか等、分析メンバ単位に障害の内容を分析している.

2. これまで、プロジェクト計画や管理の定量化に関して、やりたかったにもかかわらず、なかなか具体化できなかったことは?

**鈴木 雅哉**: 規模・工数の見積り、進捗状況、テスト項目の消化度などを見て、プロジェクト・メンバの心身の健康状態の管理をすること。

匿名: 開発フェーズごとの量的計測,工数見積り.

**匿名**: ごく基本的なことですが、諸連絡票をしっかりと 記入させて、全体(全員)で進捗状況に関する情報をシェ アすることです。

**岡本 吉史**: 正確な開発工数の見積りとプログラムの品質の均一化.

匿名: データ分析.

**匿名**: 進捗管理の定量化,特に,役に立つアウトプットを出すこと.

**匿名**: 外注に一括発注が多いため、極力定量データによる管理を行いたい.

**匿名**: 適正な工数見積りに基づくスケジュールの作成. スケジュールに従った進捗管理(特にレビュー). 仕様変 更のコントロール. 管理作業全体における定量・定性化.

大久保 功: 開発プロセス内に管理指標(たとえば、出荷前不具合情報)を設定し、それをリアルタイムで集めて、現に進行中のプロジェクトに何らかの有益なフィードバックを行いたい.

匿名: 開発形態,ドキュメント等,すべての標準化.

**匿名**: 個別生産性の把握を基にしたプロジェクト運営. ソフトウェア生産性の尺度の分析.見積り尺度.品質の的確な把握.

**匿名**: 部門,全社へと結びつけられる採算管理メカニズム.数量面だけでなく,アクションをスッキリと結びつけたい.

3. 定量的なプロジェクト計画や管理の実践を阻害して

#### いる主な障害は?

**鈴木 雅哉**: 実作業に追われて,データを管理する時間がない. 過去のデータの蓄積がない. プロジェクト管理用のツールがない(知らない).

**匿名**: 当社,客先ともに定量的な計画や管理について経験がほとんどないため,案は出ても具体化しない.当社の業務が派遣から請負に切替ってばかりで,客先との間で工数の計測に隔たりがある.

**匿名**: チームによる開発の経験はないが、テクニックは よく知っている古参のいわゆるフルダヌキの人。また、い わゆるソフト技術者然としたメンバ、上記の両パターンと も個人用プレーに走りがちだと思います。

**岡本 吉史**: メンバがプロジェクト外の仕事(過去のプログラムのデバッグ等)に予想外の時間をとられている. レビューやテスト等に充分に時間をかけられない. 部門として、そういう作業に対する認識があまりない.

匿名: 納期.

**匿名**: 技術力 (メンバーの) が高くない. また,システムが未熟である.

**匿名**: 担当者の経験年数が短い.レビュアが特定の人(経験の長い人)に偏っていて、大きな時間がレビューのために費やされる.

内藤 求: システムの規模,特徴によって,異なる開発 手順を使用したほうがいいと思われるが,なかなかむずか しい.生産性に影響を与える要因が複雑で把握できない. プロジェクト要員の能力差の定量化ができない.

**匿名**: 自分の分坦作業が多く、管理まで手が回らない。 プロジェクトが小さく、ほとんど1人または2人なので、 それほど管理の必要がない。

**匿名**: ユーザ仕様の不確定性(すなわち,初期仕様の遅れ+突発的な仕様変更+表現の曖昧性).管理・運用部門とのコミュニケーションの不足(当社の場合,マシン運用が他会社に委託されており,開発に対して積極的な協力姿勢がみられない).現場に即した管理手法が(当部門では)未確立であり,データが不足している.既存の手法のカスタマイズが不十分で,定着していない.

**匿名**: リーダ自身が担当者でもあるので、レビューの時間がなかなか取れない。ダブルキャストができていない。 標準が徹底していない。 大久保 功: 開発プロセスが定常でないか,またはよく 見えない. 開発部門が納期と生産性ばかり気にして,デー 夕がとれない. マネジメントの関心が低い.

矢吹 節史: 仕事の受け方(工数及び納期). プロジェクト内の構成(しっかりとした構成がとられない).

**匿名**: メンバーのスキルの差が大きいため、力のある者 に負荷がかかる。旧来の1担当=1プロジェクトの習慣が ぬけない。そういった理由でドキュメントが残らない。

**匿名**: ユーザ要求による設計変更+仕様変更.各メンバのスキルの低さ.客先要求中心の契約.

匿名: 開発環境に対する検討に対して、なかなか人と時間がさけず、プロジェクト実施時の見積りを正確にできない場合が多い. (当社固有の問題かもしれないが)要員の能力評価がなされていない(適切な配員を考えるのにものすごく時間がかかってしまう). そもそも根本的に要求分析段階での不備が目立つプロジェクトがあまりにも多い.

**齋藤 孝行**: 取り組んでいる分野(商品ターゲット)により、手順・管理手法・管理項目・基準などがバラバラであり、それぞれに応じた対応を考えることがむずかしい。また、従来からの伝統的な仕事のやり方を変えようとしない「技術者」たちへのアプローチも一工夫しなければならない。

**匿名**: プロジェクトによって,個人の力量がちがう.プロジェクトによっての難易度がちがう.

**匿名**: 個人スキルの違い.システムの難易度の尺度.見 積りと規模のギャップ,仕様変更等.

**片岡 賢治**: メンバーの個人個人のスキルの把握ミス. 管理者=開発者となる事態が多く,管理と開発の分離ができない.上司からの介入(プロジェクトリーダと上司との権限分担ができてない).

**匿名**: プロジェクト内部で、開発とサポートを併存させて行っていること、サポートにより開発スケジュールが遅れる. 内部要員が不足の場合、外部への発注を行ったりする際の問題.

**岡本 健二**: 要求分析段階での見積り技術.システム開発はプログラムだけでなく,他の環境定義 (DB/DC,帳表書式等)の負担が大であり、これらはプロジェクトによっては、大きな変動因子である.

匿名: 自身の経験不足. 日常会話の不足. メンバへの周

# **Facts and Opinions**

知事項の不徹底.

**村田 幸夫**: プロジェクトを運営していく上で、その手法がわかっていないため、負荷がリーダに集中してしまう. 作業環境が悪い.

4. これから、プロジェクト計画や管理の定量化を目指して、どんなことを試みたいと思っているか?

**鈴木 雅哉**: 定量的にデータをとりながらプロジェクト を運営し、次のプロジェクトの参考にするとともに、その プロジェクトの見通しを立てるようにしたい。

**岡本 吉史**: 今回の堀江さんの講義は、教科書的な話でなく、実際に現場での経験にもとづいた話だったので、たいへん参考になった。利用できることから、利用してみたいと思う。

**匿名**: 各工程ごとに、見積りと実績の差を細かく把握し、 次工程に反映していくようにしたい、レビューは、現在6 人位(グループメンバ全員出席)でやっているが、特定の 人間しか発言せず、その他の人間は退屈しているのが現状 である。今後は、1対1のレビューを試みてみたい。

匿名: 当社の場合、昨年秋に情報システム部門が新しく設置されたが、知識・技術・経験ともに未熟な若手メンバーに加え、管理者及び管理のノウハウが欠如している.このような現状で、まず管理の体制・ルール作りから初め、当部門に敵した管理手法を確立して行きたいと考えている。今回のセミナーは多々参考になった。実践に活かしたい

**匿名**: 標準化の徹底,ダブルチャンス化を進めたい.セミナーの内容は,日頃何となく考えていた問題点を明らかにしてくれて非常に参考になるものであった.

大久保 功: 品質保証部門にいる関係上,ソフトウェア の品質向上を推進せねばならない。そのための第一歩とし て出荷後の品質のレベルを把まえることはできた。あまり よくなく,かつ年々よくなっているわけではない。そこで, とりあえずは,工程内の管理指標を設定したい。

矢吹 節史: 今後,レビュー回数を増してみたいと思います.

匿名: ドキュメントだけでも、まず標準化したい.

**匿名**: プロジェクト内メンバのローテーション.正確な 見積りと客先交渉+対応.

匿名: 短期のプロジェクトに対する採算性を重視した管

理手法の開発を行いたい. 人員の有効利用, および対客先 へのサービスレベルの向上といった意味で, かなりの効果 があると考えている.

**齋藤 孝行**: 将来的には、プロジェクトの品質を評価し、 それを TQC 活動に反映させたい. ただ、その導入にあたって当業界において小集団活動(サークル活動)の運営形態に若干の疑問を抱きつつあるところである. 新しく品質管理部門として発足し、かつ TQC 活動の導入を検討している現在でも、この障害の解決策に苦慮している.

**匿名**: プロジェクト発生開始と,終了時に定量化を試み たい

**匿名**: データ収集をもとに、ソフトウェア生産性の尺度 ・要因の分析を行いたい、バグ分析による品質傾向の把握 /対策を心がけたい、そのための要員教育も大切である。

村田 幸夫: 最近,レビュー技法等を意識するようになって来ており、今回のセミナーでレビュー時期(タイミング)や重要性がわかった.

5. あなた自身が最近関係したプロジェクトでのレビューやウォークスルーの実施状況は?

(1) 全体設計レビュー

徹底的	に実施	5
まあま	あ実施	12
あまり		3
17とん	J.	2

(2) 詳細設計レビュー

徹底的に実施	4
まあまあ実施	11
あまり	4
ほとんど	3

(3) コード・ウォークスルー

徹底的に実施	1
まあまあ実施	3
あまり	10
ほとんど	8

## 6. レビューやウォークスルーの効果は?

**匿名**: 思い込みによる誤りの発見.他人の分担を理解することによる全体の把握.有用なテクニックの共有→標準化.

# Facts and Opinions

**岸野 直樹**: 教育的効果. 設計者に対する心理的抑圧からの解放. 標準化.

**山田 修一**: 「動作付きで動作する」ように、バグの発生している部分がしばれる.

**匿名**: 他のプロセスの理解に役立つ. モラルが向上する. 工程の見直しができる.

**匿名**: バグが早期発見できる. 設計/コーディング技術が向上する.

**鈴木 雅哉**: レビューで検出されたバグの中で多いものは、次の開発時に作者自身が注意するようになる. 単体テストが事実上省略できる. 個人の能力をみきわめることができる.

**岡本 吉史**: 簡単なミスで本人が気づかないものが、容 易に発見できる.システムの内容に関する情報が共有でき る.

**片図 賢治**: 業務内容の固定化(大幅な仕様変更の防止). 客先との意志の統一. 詳細な点でのモレが救える(すべてとはいえないが, 機能もれ, 考え方の不一致が救える)

矢吹 節史: 目に見えた効果はないように思われる.

匿名: オーソライズが得られる.

**匿名**: 詳細設計レビューに効果が見られる(もっと積極的に実施したい).

**権熊 敏朗**: 局所的な誤りが発見しやすい. 担当者相互 の仕事内容が理解し合える. 納入時, 報告書が書きやすく なる.

匿名: やり出して間がないため、まだ効果はわからない.

**匿名**: 全く望むものと異なるソフトウェアができてしまうことはない.

7. レビューやウォークスルーを行なう上での主な問題 点や注意事項は?

匿名: なかなか時間の調整がむずかしい.

**岸野 直樹**: 準備(資料の事前配布,チェック). スケジューリング. 進行のコントロール.

**匿名**: プログラム自体を作ることに追われ、しっかりしたレビューを行なう時間を取ることが困難なケースが少な

くない. 各担当者に、レビューの重要性・方法論・意義を認識させることのむずかしさ(正しい動機付け).

匿名: 時間が足りない. レビューを行えるメンバの不足.

匿名: レビュアーの選択.

**鈴木 雅哉**: リーダが全体のレビューに参加するため, リーダの時間が足りなくなり, 待ちが発生してしまう. レ ビューは, マシンデバグに比べて, 集中力等, 強い精神力 が必要であり, レビュアの人材に左右されやすい.

**岡本 吉史**: レビューのための充分な時間が確保できない. レビューのための資料の準備 (コピー他) に時間がかかる. 会議室を確保するのがむずかしい. レビューに対するメンバの意識が充分ではない.

**片岡 賢治**: 必ず合意すること、(客先とでもどことでも). 合意していない箇所は再レビューとする. レビュー 実施報告書を記述して,必ず回覧すること. 責任者が参加 すること.

**匿名**: 現在,参画中のプロジェクトは火消しが大目的であり,またプロジェクトも最終ステップに近づいており、 レビューは実施できなかった.これは,忙しさにかまけての反省点であり,各種チェックリストを使用してのレビューを行なうべきであったと実感している.

**匿名**: コードレビューに時間 (工程) がかけられるよう なスケジュールを立てることがむずかしい.

**椎熊 敏朗**: 時間的制約. なれあい (特に上司のモラル が低い). 場所がない.

匿名: 時間がかかる. やり方が適切かどうか自信がない.

匿名: 具体的な方法論を確立していない. 時間がない.

**匿名**: 納期や予算との折り合いをいかにうまくとって時間をひねり出すか?

**匿名**: プロジェクト内部に限らず汎用的なたとえばチェックリスト等も形式化して、場合に応じた形で利用できるように考えていきたい。また、個々のレベルに応じた教育についても、考えてみる必要があると思う。

### 定量的管理のためのモデル

春のセミナー・ウィーク参加者アンケート

### 1. あなたが最近関係したプロジェクトでは、エラーや バグにかんするデータをどの程度収集していますか?

	要求仕樣	設計	プログラム	テスト
かなり	0	1	3	5
まあまあ	6	6	7	7
あまり	6	5	5	4
ほとんど	2	3	2	1
まったく	4	1	0	0

### 2. そうしたデータの分析および結果のフィードバック はどの程度行なわれていますか?

	要求仕樣	設計	プログラム	テスト
かなり	0	1	2	3
まあまあ	6	6	7	8
あまり	2	2	3	2
ほとんど	3	6	3	1
まったく	6	3	3	3

### 3. その分析にさいして、信頼性推定モデルなどの統計 学(推計学)的な手法をどの程度利用していますか?

かなり	0
まあまあ	1
あまり	3
ほとんど	8
まったく	5

# 4. 信頼性データの収集や分析に関して、現在どのような問題がありますか?

**匿名**: 収集すべきデータ項目が定まっていない. 分析する手法がわからない.

**野田 祥一**: プロジェクトが終了した際の完了報告に、 バグ件数等を記載しているが、分析者はリーダだけであり、 作業者までフィードバッグしていない。作業中でのデータ 分析を行っていない。作業開始時の予測を行っていない。 テストの終了は、期間・納期での終了、もしくは、たまた ま障害曲線をかねた場合というのが多い.

**匿名**: データ収集・分析の重要性をユーザーにアピール するための実績を作るのに必要な工数と人員が十分に確保 できない.分析のスペシャリストが不在.

**片岡 賢治**: データ蓄積はしているが、分析・フィード バックまで行かない、データの蓄積法を思考中である。同 じく、どう分析するかが....

匿名: 収集すべき情報 (データ) を明確にすることがむずかしい. 現状では「収集しやすい」データしか収集されていない (たとえばログ等, 静的なものばっかり). 動的なデータ収集がむずかしい. 実際にアクティブな作業の中で, リアルタイムに動的データ収集をするためのメカニズム, 時間, マンパワーがない. 要求段階のエラーやバグについては, 分析の仕方がわからない. 「誤った要求仕様」ということはいえるが, 「正しい要求仕様」を確定するすべがない.

**匿名**: 収集の基準が明確でない. 各段階での成果物を作成した者の報告だけが頼りである. データ収集用データベースが整備されていない.

三村 誠: 修正による基礎値データの変化による品質の 予想. テスト項目の設定方法により, データの変化がちが うこと. 品質の信頼性の妥当性及び十分性.

**匿名**: データ収集をする仕組みができていない. データ 収集・分析(つまり QA) に対する意識が低い. データ自 体がほとんど蓄積されるようにならない.

**匿名**: データ収集に、時間と手間がかかる. グラフ化、 モデル化がむずかしい. モデルの意味することが読み取れ ない.

**匿名**: ソフトウェアメトリクスが自社なりのものとして確立されていない. データ収集の基準, 仕方, 蓄積方法について, 具体的な手段が雑である. データ分析の各種技法についての勉強が不足しており, 結果のフィードバックがなされていない(弱い).

**匿名**: データ分類に統一的な基準がなく、個人差が出ている。報告の遅れによる強化策の遅れ(目前の対応). テストケース設計、テスティングの標準化.

5. 今後ソフトウェアの信頼性モデルをどのように活用 できそうか?

**匿名**: まず参考書により、モデル作成およびパラメータ の推定を試行したい.

**野田 祥一**: ソフトウェアの品質保証に関していえば、 現状テスト結果の報告でしか行っておらず、品質の予測・ 分析のフェーズがない。品質をチェックする工程・手法を 標準化する必要がある。

**匿名**: ソフトウェアの品質を得点化でき、品質向上につながり、また納品済のソフトウェアに対するユーザークレーム数とそのための工数・人員の予想がある程度できると感じました.

**匿名**: 本当はキッチリとした捉え方をして,現場にフィードバック (適用) してみたいのだが,現時点ではスタッフ機能が充分でないこともあって,製品品質部門以外は現実には行っていない.しかし,遅かれ早かれやらなければならない(と思う).

**匿名**: 開発管理用のデータベースの開発の参考にしていく、今回のセミナーでは、モデルに関する基本的な考え方を分かりやすく解説してくれたので理解しやすかった.

**匿名**: テストによるバグ・データの収集及びモデル化に よる次の予測等が少しでも行えそうな感じがする.

**匿名**: 自社におけるデータ収集およびモデル化するため のパラメータを決定することがむずかしいと思う.

**匿名**: ウォータ・フォールモデルのどこから適用するかが、むずかしいと思われる. ユニット・テスト後もモデルの適用とは必ずしも結びつけにくいが、収集の目安となるものを模索したいとは思う.

6. 現在, あなたの周辺では, プロジェクトのコスト見 積りはうまくいっていますか?

> かなり 0 まあまあ 5

あまり 7 ぜんぜん 1

7. COCOMO モデルについての予備知識は?

よく 0 少し 5 名前だけ 3 全然 4

8. 見積りにはどんな技法やツールを使っているか?

**匿名**: 過去開発したプログラムの開発率,流用率等を基 に見積っている.

**匿名**: 専門家による判断. 類推法. トップダウン. ボトムアップ. Mac Project.

野田 祥一: 標準生産性から算出される工程ごとの必要 工数と、稼働実績からの各工程での差異をチェックする.

**笠原 後哉**: WBS. Mac Project 2. ステップ見積り. 工程別比率.

**匿名**: COCOMOモデル (1つのデータとして利用). あとは (運と勘) + (高めの人件費設定) + (リスク見積).

9. 見積りがうまく行かない原因は?

**匿名**: 開発設計段階の仕様が詳細的に明確でない.テスト段階での仕様変更が多い. 開発期間に制約がありきびしい.

野田 祥一: ユーザの要求仕様の決定が遅れたり、あまかったりするために、工程の手戻り、仕様変更が多い、システムテスト時の計算機コストの見積りが正確でない。

笠原 俊哉: 作業量および作業内容が不明確 (リスクの 予測が甘い) 時点での見積り要求が多い. ハードウェア絡 みのトラブル予測がむずかしい (ファームウェア主体の開発が多いため). 予実績のデータ収集・分析・フィードバックが体系づけられていない.

匿名: 見積り時点で、要求仕様が明確によることがまずない、プロジェクトにネットで費やされた人件費やマシン費などの算出がむずかしい、アイドルがかなりまじってしまい、コスト超過になっている、実プロジェクト関係者の見積りに対する意識の低さ、技術的なブレークスルーさえ

できれば、コストは度外視するといった傾向があり、しか もそれが認められてしまう.

**匿名**: 実績値の収集が統一化されていない. 見積り方法が未確定.

**匿名**: 仕様の変更量の多さ. 委託会社の配員メンバーの 技術レベルがまちまち. マシン環境が一定化していない.

内藤 求: 規模の見積りがうまくいかない(初期の段階で顧客から提示される資料だけでは、すべてを読みきれない).途中の工程から受託する場合が多いが、実際には、前工程が完全に終わっておらず、そのことを前提にした見積りは、現実に合わなくなってくる.

**匿名**: 今までは、ほとんど実工数を問題にしていたので、 見積作業自体には、それほど真剣に取り組んでいなかった。 プログラマーの経験だけを頼りにしている。

**匿名**: スケジュールが政治的に決定されている. 開発者 の過去の実績がわからないため, 見積れない.

匿名: 開発規模の拡大,要員の問題等,不確定要素による影響.ハードウェア環境,個人スキルのバラツキ(生産性の違い).見積り自体の甘さ.経験と勘のみに頼ることが大部分(このぐらいでできるだろろう....)客先との対応:仕様変更,納期,要求仕様の不明確さ,品質管理の不備.

10. COCOMO モデルは、今後どのように活用できそうか?

匿名: かなり使えるだろう.今後の仕事の中で COCOMO をもとに見積りを行い,いくつかデータおよび 実績を取ってみたい.

**匿名**: 今後は数多く導入されていく方向だろうが、現状はまだまだ未知数の部分が多い。また、導入にあたっての モデルのカスタマイズに関するる問題も多そうです。

**笠原 俊哉**: 使えると思う.実際に富士通で作られたツールを使用してみたい.

**匿名**: 1つの参考データを得る手段としては、すぐにも利用できるだろう. ただし、パラメータ値の見直しや効果の評価は、てまがかかる割には実際の精度が不明なので、余り厳密に考えなくてもよいのではないか.

匿名: COCOMO のカスタマイズに関して検討をして

みたい. ただし、客先を固定しないとムリなように思われる.

内藤 求: DSI (Dilivered Source Instructions) の見 積りが正確にできるようにしたい. そのあとで、 COCOMO モデルを活用したい. 4GL とかソースの自動 合成が進んでくると、DSI を基準にしたコスト見積りは、 現実に合わなくなってくるのではないか.

**匿名**: このようなコスト・モデルの話は初めてだったので、興味深く聞いた.しかし、プログラムステップ数が、プロジェクトの開始の前にどの程度正確に予想できるのかわからないのが心配である.

**匿名**: 活用できそうであるが、カスタマイズが大変そうだ、パソコンで市販ツールがあれば、急速に普及すると思う.

匿名: モデル適用のカスタマイズ方法が問題,データ収集の方法,各種要因のカスタマイズに,主観的要素が入り込みそうである.冷静なデータ収集・分析によるカスタマイズを目指したい.うまくカスタマイズできれば,感度分析による代替案の設定は,有効な手段となりそうである.また,予算等の関係上,客先へのオーソライズも問題かと思われる.開発システムやアプリケーションの特性による分類も必要かと思われる.

### ソフトウェア開発とテクノストレス ~SEAのアンケート結果をもとに~

東京大学 総合文化研究科 藤垣 裕子

### 1. はじめに

情報処理産業の成長に伴い、ソフトウェアを開発する技術者の数も年々増加しつつある。そのソフトウェア開発に従事する技術者の職務負担について、「テクノストレス」といった言葉が使われているものの、現実のソフトウェア開発作業の特性に基づいた負荷の実態がきちんと報告されている研究例は少ない。本報告では、SEAにおいて1989年2月に行ったアンケート調査(ソフトウェア開発技術者の作業負担に関するアンケート調査)をもとに、その負荷の実態について考察する。

### 2. 先行研究

「テクノストレス」という言葉は、C.Brodの書いた同名の本 3に端を発している。Brod がこの本の中で扱ったストレスの内容は非常に多岐にわたっており、コンピュータを扱う作業者の知的作業負荷、長時間労働による疲労の話から、コンピュータの論理に自分を適応させようとすることからおこる人格変容(テクノ依存症やテクノ不安症)の話まで含まれている。今回のアンケート調査では、適応症候群としての人格変容の話は扱わず、作業内容や勤務体制からくる負担や疲労、およびメンタルヘルス上の問題について扱うこととする。

上記の先行研究の問題点は、まず経営学サイドからの研究は、組織統制に中心的視点があるために、組織関連のマクロストレス(給与、命令体系、学習機会の問題など)の分析は細かいが、作業内容に立ち入った分析が粗いという点である。また、医学・保健学サイドからの研究は、健康調査や意識調査の結果が、ソフトウェア開発の作業内容のどのような性質、特性に起因するのかについての考察が希薄な点である。

### 3. アンケート調査の視点と負荷分析の方法

### 3.1. アンケート調査の目的

2. のところで述べた先行研究の問題点を考慮に入れ、筆者は今回の調査に先だって、ソフトウェア開発の工程別の負荷、および納期切迫による負荷の分析を行っている。601 これは、現場の技術者へのインタビュー結果をもとに問題を絞って行った調査分析であるが、その分析の結果、特に調査・概略設計の上流工程において精神的負荷のかかる傾向のあること、およびデバッグ工程において心身ともに大きな負荷のかかること、納期が迫ってくるにつれて精神的負荷の高まること、等が示された。

この工程と納期の2つのファクターは作業内容による負荷を分析する上でかなり重要な要因ではあるが、その後のインタビュー調査から、この2つの他に、以下の様な負荷影響項目の存在するこ

とが示唆された。それは、大きく作業内容関連、作業環境関連、作業体制関連、組織上の問題の4つにグルーピングされる。作業内容関連では、開発分野(事務、OS、通信、ツールなどの開発内容)、新規開発の割合、ソフトウェアの規模、社会への影響の度合、要求品質、納期の融通、仕様変更の有無などの項目が含まれる。作業環境関連は、作業場所、開発マシンやターゲットマシンの場所、開発環境、開発技法などの項目である。作業体制関連には、プロジェクトチームの構成およびチーム内のポジション、同時平行担当プロジェクト数などが含まれる。組織形態関連では、会社の規模、技術力、種類(メーカ系、独立系など)、ユーザーとの関係などがある。

今回のアンケート調査では主に、以上の、工程と納期以外に抽出された項目の作業負荷への影響を調べることを目的に設計された。その構成は次の節で述べる。

### 3.2. アンケートの構成

今回のアンケートは、上で述べた項目に、SEAで1988年9月に行われた開発環境実態調査の項目の一部を付け加えて作成されている。大きくわけて6つの質問群(Q1~Q6)から成っている。Q1は基本データで、性別、年齢、勤続年数、経験年数、残業時間などの10項目から成る。Q2は組織と所属に関する設問で、会社規模、種類、組織内所属部門などの4項目である。Q3はプロジェクト内容で、3.1のところで述べた作業内容関連、作業場所関連、作業体制関連の20項目から成っている。Q3がプロジェクトに関する客観的事実を尋ねているのに対し、Q4では、自分のプロジェクトに対する主観的な印象評定を求めている。要求品質、納期評価、開発環境整備の度合、マシンの利用時間、仕事の難点、コンピュータ親密度など11項目から成る。以上の4群が負荷要因についての設問である。Q5とQ6は、負荷分析のための設問となっている。

### 3.3. 負荷分析の方法

負荷の検証には、ストレス評定尺度(Q5)と、CFSI(蓄積的疲労徴候調査、Q6にあたる)を用いた。

ストレス評定尺度法は、上記の先行研究9、18、23でも用いられているものである。ソフトウェア開発においてストレス要因となると思われる22の項目(納期の時間的切迫、仕事量の多さ、残業の多さ、勉強時間の不足、チーム内やユーザー間の人間関係、仕様の曖昧さ、要因の質的不足など)に対し、それぞれ7段階評定(1:全くストレス要因ではない、2:殆どストレス要因ではない、3:あまりストレス要因ではない、4:どちらともいえない、5:少々ストレス要因である、6:かなりストレス要因である、7:非常におおきなストレス要因である)での応答を求めている。

CFSIは、労働科学研究所の越河によって開発された方法 かで、上記の研究 6 における工程別の負荷の分析でも用いた方法である。この方法は、心身の違和感や疲れの感じなどが何日も停滞している状況(蓄積的疲労)を尋ねる評定法で、その疲労感の訴えのパタンから作業者にかかっている作業負荷を探ろうとするものである。数カ月から数年にわたる開発期間をもつソフトウェア開発の作業負荷を調べる上で有効であり、メンタルヘルスを含めた心理アセスメント的要素を持っている。81個の質問に対し、 $\bigcirc$ か×で答える形になっているが、この81個の質問は、過去の調査における約12000件のデータの因子分析によって、8つの特性群(不安徴候、抑うつ状態、一般的疲労感、イライラの状態、労働意欲低下、気力減退、慢性疲労、身体不調)に分けられている。応答結果から、各特性項目群に属する項目への平均訴え率を求め、それを図示することができる(例:図3)。このパタンから、作業者にかかっている負荷の様子を探る手がかりが得られる。

#### 4. 結果と考察

### 4.1. アンケート回収状況

アンケートは、まずSEA-MAILで個人会員に約1100部配布した。また、SEA会員の所属する会社にグループ回答を依頼する形で、各社に20部ずつ、約2000部を配布した。有効回答数は、個人回答で270件、グループ回答で18社28部門553件、計823件である。

### 4.2. 単純集計

男女比は約9:1 (男子713件)である。年齢構成は、20代57%、30代35%、40代5%となっている。平均勤続年数は4.5年、平均経験年数は6.5年である。残業時間は月平均30~39時間の人が15%で一番多く、平均約45時間となっている。会社の種類はソフトウェアハウスが75%を占め、その内訳は、メーカ系32%、ユーザ系8%、独立系35%となっている。

### 4.3. ストレス評定

男女別のストレス評定の順位を図1に示す。男女とも、納期の時間的切迫、仕事量の多さ、仕様の曖昧さがストレスとして高い評定になっていることがわかる。女子では、チーム内人間関係、仕事の難度、人材育成機会の不足の評定が男子に比べて上位の評定になっている。男子では、報酬への不満足が、女子にくらべ評定順位が高い。この評定は、チーム内ポジションによって異なる。(図2)シニアメンバでは仕様の曖昧さがストレスとして評定が高いのに対し、上位の管理者では、仕様の曖昧さもさることながら、雑用の多さ、要員の質的不足の評定値が高くなっている。また、この評定は会社ごとにかなり異なっている。図3にその例をあげるが、a社では全体の平均値に比べ、ユーザ間人間関係、納品後のトラブル、仕様の曖昧さなどの、ユーザとの関係に起因するストレスの評定が高くなっているのに対し、b社では、開発環境や作業環境の悪さ、報酬への不満の評定が高くなっており、またc社では納期の時間的切迫、仕事量の多さ、休日不足、残業時間の多さなど、仕事の多さに起因する評定が高くなっている。このように、このストレス評定値を全体の平均と比べることにより、その会社において何が技術者達にとって大きな負荷要因となっているかを調べることができる。

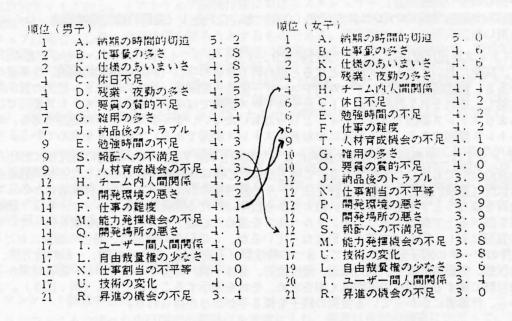
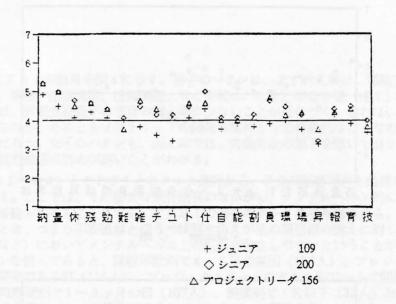


図 1. ストレス評定順位



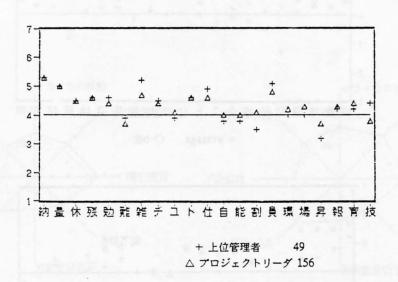
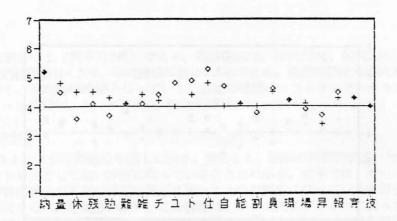
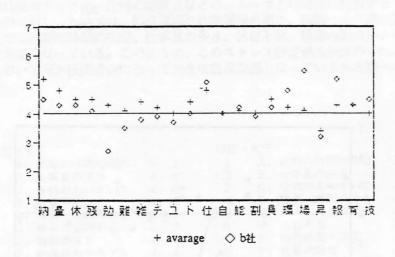


図2. チーム内ポジションとストレス評定



+ avarage 🔷 a社



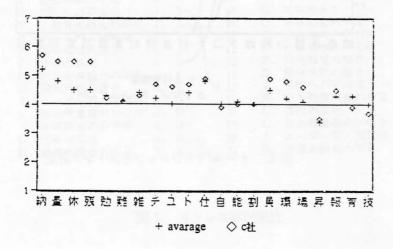


図3. 社別ストレス評定の例

### 4.4. CFS I

男女別のCFSIの結果を図4に示す。男子のパタンは、まず訴え率は、電機労連の12000のデータ(職種は、事務、生産現場、情報関連、など多岐にわたる)のQ3値(注1)を少々越えている。形の歪みでは、労働意欲低下の項目の訴え率の少ないことが目につく。これは、今回の標本集団に特徴的なことなのか、それともソフトウェアの開発作業にとって特徴的なことなのか、今後調べていく必要があるだろう。女子のパタンも、訴え率では、労働意欲の低下を除いてQ3値を越えている。形の歪みでは、気力減退の訴えが高いことがわかる。

各問い(Q1~Q4)とCFSIとのクロス集計から、各負荷影響項目と負荷の関係を調べることが可能である。ここでは、その膨大な集計結果のなかから、メンタルヘルス的に少々問題があると思われる結果を拾ってみることにする。CFSIの過去のデータの積み重ねから、そのパタンの歪みが左上に偏るとき、つまり不安徴候と抑うつ状態の訴えが他の項目群の訴えに対して大きい時に、その集団(職場など)においてメンタルヘルス上の問題が発生しやすいということが確認されている。そのようなパタンを拾ってみると、経験年数別で4-5年の集団(134人)、プロジェクトの性格別ではプロダクト開発である群(115人)、プロジェクトチームの大きさ別で一人で開発している群(28人、図5)開発期間別で1~3ヶ月の群(107人)、規模別で1 K以下(32人)および500 K以上(52人)の群でその様なパタンの特徴がみられた。したがって経験年数4-5年、プロダクト開発、一人での開発、1~3ヶ月の開発期間、ソフトの規模が小さすぎるあるいは大きすぎるなどの作業条件が、技術者のメンタルヘルスの状況に影響を与え易い条件であることがわかる。

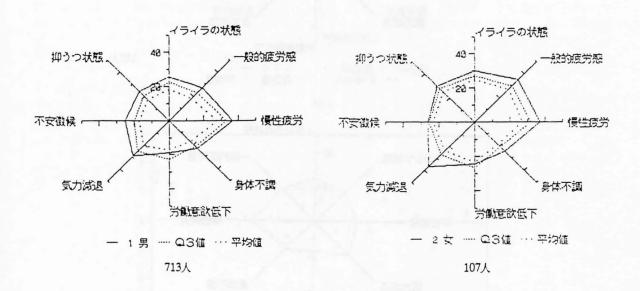
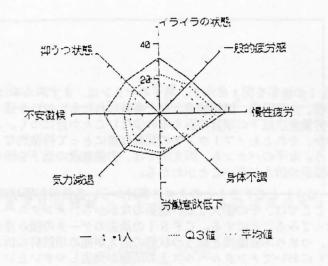
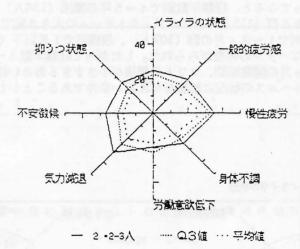


図 4. 男女別 CFS I の全体結果

以上は結果の全体的傾向のまといである。以下にトピックスとして、開発分野別、工程別、開発環境整備状況別(すべて男子のみ)の結果をまとめることにする。





167人

28人

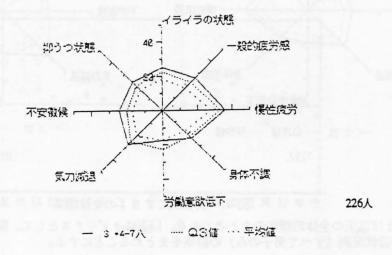
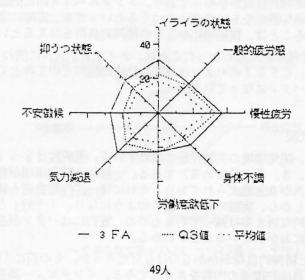


図 5. プロジェクトチームの人数とCFSI

### 4.5. 開発分野別の負荷の状況

開発分野は設問Q3-4において問われており、選択肢は、事務、科学技術、FA、OA、CAD、ツール、通信、言語、OS、その他の10つに分かれている。これら10分野のうち、事務(162人)、CAD(56人)、ツール(86人)に関してはCFSIのパタンが平均的で安定しており、ストレス評定も、事務においてユーザー間人間関係の評定が高い他は、めだった特徴はなかった。これに対し、CFSIでメンタルヘルス的に問題のある左上の偏りのパタンを見せたのは、科学技術(24人)、FA(49人、図6)、OA(37人)の3つである。ストレス評定では、科学技術は全船的に評定値が高く、FAでは仕事量に関する評定値が高く、OAでは仕様の曖昧さの評定値が高かった。各分野の作業内容のなかにメンタルヘルスの状況に影響を与え易い要素が含まれていると考えられる。その要素として考えられるのは、科学技術ではアルゴリズムの難しさ、FAではトラブルが生じたときの社会への影響の大きさである。また、この3つとも、プログラミングの結果をすぐに目でみて確かめること(CADやツールでは可能)ができない、という点で共通している。



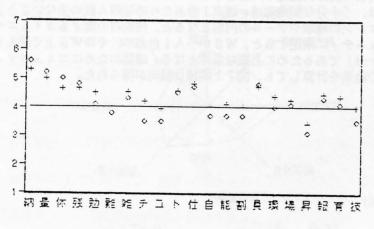


図 6. 分野別CFSIとストレス評定(FAの場合)

+ AUERAGE 0 3 FA

その他の分野では、通信(45人)において仕事量のストレス評定値が低いにもかかわらず、CFSIにおいて身体不調、不安徴候の項目にパタンの歪みがでているのが特徴的である。言語(21人)は、CFSIもストレス評定もどちらも訴え率が高く、OS(21人)では、CFSIパタンで気力減退、一般的疲労感の突出が見られ、ストレス評定値は、要員の質的不足、開発環境や開発場所の悪さ、報酬や育成機会の不足などの項目が高い値を示している。

### 4.6. 工程別の負荷状況 - 通常設計法とプロトタイピングの比較

この調査の前に行った研究6において、調査・概略設計などの上流工程で精神的負荷がかかりやすいのは、客先のニーズをつかみ、それを実現させる大枠を考える際の先の見えなさや不安感によるものであることが示されている。プロトタイピング手法は、そういった上流工程で客のニーズを摑みきれないので、まず簡単な機能をもつモデルソフトを作り、それに機能を追加して客のニーズにあったものを作る(仕様の定義→コーディング→デバッグ→テスト→評価のループを繰り返す)というものである。これにより、仕様の定義などの上流工程の負荷が少々軽減されるのではないかと予想することができる。ところが実際の結果は、プロトタイピングの仕様定義工程のほうが、通常設計法の調査・概略設計工程より精神的負荷がかかっており、メンタルヘルス的に問題の多いパタンになっていることが示された。簡単な機能をもつモデルをつくるといっても、実際に客のニーズのはっきりしないうちに設計を開始することは、技術者にかなりの精神的負荷を与えるということが予測される。

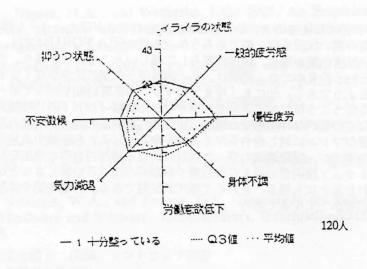
今回の工程別集計で他に新たにわかったことは、メインテナンスに携わる技術者の精神的負荷が高いということである。CFSIのパタンは通常設計法の他のどの工程とくらべても左側に偏っており、精神的負荷の高いパタンになっている。

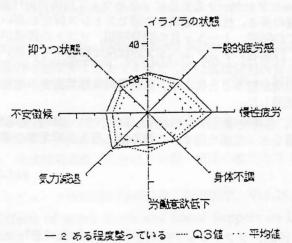
### 4.7. 開発環境別負荷の状況

設問Q4-6において、開発環境の主観評定を求めている。選択肢は3つ(1. 十分整っている、2. ある程度整っている、3. 不十分である)である。一般に、開発環境が整うほど、つまらない反復作業が減り、生産性があがると考えられており、それに伴って作業負荷も軽減するのではないかという予測もたてられる。しかし、実際の結果は図7のようになり、「十分」と答えた群は、「不十分」と答えた群に比べ、量的な訴え率は減っているものの、質的にはパタンが左に偏っており、精神的負荷のかかっている状況におかれていることがわかる。

開発環境が整うほど精神的負荷がかかるのはなぜだろうか。その前に「開発環境が十分整っている」と答えた技術者の具体的環境の内容をさぐってみる。インタビュー調査によると、技術者間の一番広い共通認識は、「十分な開発環境=端末1台あたりの使用人数の少ないこと」という認識である。より具体的なマシンの種類やツールの内容となると、技術者の属するコミュニティによって異なる。UNIXコミュニティに限定すると、WSが一人1台あり、そのWS上で使えるツールのあることが、「開発環境十分」であるために必要な条件となる。確認のためにコミュニティをツール開発に限定してCFSIの結果を計算しても、図7と同様な傾向が得られた。

307人





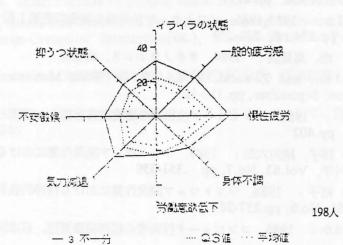


図7. 開発環境とCFSI

マシンが自由に使えること、ひいては一人1台のWSが使え、ツール群の整っていることが、なぜ精神的負荷の高い状況を招くのであろうか。5月のSEA-FORUMでは、「開発環境の整備によってつまらない反復作業が減り、精神作業として難しいところだけが残り、その比率が高くなるからではないか」という意見がでた。問題は、その頭を使う作業の比率が高くなるという、質的な作業密度の増加にあるようである。他にも「今まではマシンの応答時間を待ってボーッとする時間があったのに、WSをを使うようになってからそういうじかんが減った」「同じ時間分作業すると、WSのほうが疲れる」などの量的な作業密度の増加(WSの作業能率の向上、処理の高速化により、単位時間あたりに作業者がマシンに対し命令を与える頻度が高くなる)を指摘する意見もある。これらの質的量的作業密度の増加が、開発環境の整った状況における精神的負荷の増加を招くと考えられる。この問題は、ソフトウェア技術者の過大な残業量や納期前の追込みの厳しさなどが開発環境の整備などによって解決されたあとにも残る、きわめて現代的問題である。労働時間や休憩時間のありかたなどに一考を加える必要がでてくるであろう。

### 5. まとめ

SEAでおこなったアンケートをもとにソフトウェア技術者の作業負荷について論じた。納期の時間的切迫、仕事量の多さ、仕様の曖昧さの順でストレス評定の高いこと、経験年数4-5年、プロダクト開発、一人での開発、 $1\sim3$ ヶ月の開発期間、ソフトの規模が小さすぎるあるいは大きすぎるなどの作業条件が、技術者のメンタルヘルスの状況に影響を与え易い条件であることが示された。また、開発分野ごとの負荷の状況の違いや、プロトタイピング手法が作業者の負荷を増加させている傾向のあること、開発環境が整うことにより、質的量的な作業密度が増加し、精神的負荷の高まる傾向のあることについて議論した。

注1) Q3値とは、各特性項目群ごとに訴え率の小さい順に各職場を並べ、その4分の3のところに位置する職場の値をとった値を指す。これを越えると、訴え率の多いほうから数えて25%以内に入ることになる。

### Reference:

- 1) 朝倉隆司ほか: 1987. ソフトウェア技術者におけるいわゆる"テクノストレス" について, 産業衛生学会講演集, pp.119
- 2) 荒井保経ほか: 1985,1986. ソフトウェア技術者の健康調査第1報~第3報,産業 衛生学会講演集,pp.174,pp.272-273
- 3) Brod, C. 池、高見訳: 1984. テクノストレス, 新潮社
- 4) Couger, J.G., and Zawacki, R.A.: 19 78. What Motivates DP Professionals?, Datamation, September, pp.116-123
- 5) 原谷隆ほか: 1988. コンピュータ技術社の職務内容と抑うつ症状との関連性,産 業衛生学会講演集, pp.402
- 6) 林 (藤垣) 裕子, 越河六郎: 1987. ソフトウェア開発作業における精神的負荷の 分析(1), 労働科学, Vol.63, No.7, pp .351-359
- 7) 林(藤垣)裕子: 1988. ソフトウェア開発作業における精神的負荷の分析(2), 労働科学, Vol.64, No.6, pp.257-267
- 8) 市村俊明ほか: 1986. コンピュータ技術者の精神健康管理, 日本医事新報, No.3 266, pp.47-49
- 9) Ivancevich, J.M., Napier, H.A., and Wetherbe, J.C.: 1983. Occupational Str ess, Attitudes, and Health Problems in the Information System Professional, Comm unications of the ACM, Vol.26, No.10, pp800-806

- 10) Ivancevich, J.M., Napier, H.A., and Wetherbe, J.C.: 1985, An Empirical Stu dyof Occupational Stress, Attitude and Health among informational Systems Perso nal, Information&Management, Vol.9, pp.77-85
- 11) Keenan, A., and Newton, T.J.: 1987, Work Difficulties and Stress in Young Professional Engineers, Journal of Occupational Psychology, Vol.60, pp.133-145
- 12) 河野友信: 1985, テクノストレスと疾病, 産業衛生学会講演集, pp.170-171
- 13) Lo, M.W.: 1987. Occupational Stress in the Information Systems Professio n, SIGCHI Bulletin, Vol.18, No.3, pp.25-29
- 14) 百渓浩ほか: 1986, ソフトウェア設計者のストレスに関する調査研究, 産業衛生 学会講演 集, pp.271
- 15) Newman, R.C., Newman, W.A., and Evance, G.E., Examining the Industry Compa rative Traits of Hardware and Software Manufacturers, Information&Management, V ol.13, pp.171-178
- 16) 日本情報処理開発協会編: 1986, ソフトウェア白書
- 17) Razarus, R.S.: 1966, Psychological Stress and the Coping Process, MacGraw-Hill Book Co.
- 18) Saleh, S.D., and Desai, K.: 1986, Occupational Stressors for Engineers, IE EETransactions on Engineering Management, Vol.EM-33, No.1, pp.6-11
- 19) 千田ほか: 1986, ソフトウェア産業従事者の生活パタンに関する事例研究, 産業 衛生学会 講演集
- 20) 山崎喜比古, 朝倉隆司: 1986, ソフトウェア技術者の心身の健康と労働生活との 関連に関する調査・分析, 産業衛生学会講演集, pp.267
- 21) 山崎喜比古: 1988, 今次技術革新下における労働・職場の変化とストレス, 都立 労働研究 所報, No.9, pp.129-148
- 22) 渡辺登: 1986, コンピュータ技術者の精神保健, 精神医学, Vol.28, No.3, pp.3 37-344
- 23) Weiss, M.: 1983, Effects of Work Stress and Social Support on Information Systems Managers, MIS Quarterly/March, pp.29-43
- 24) Worsley, M.: 1988, ICSE Focuses on people, IEEE Software, Vol.5, No.4, pp. 93-94
- 25) Zavala, A.: 1984, Stress and Factors of Productivity among Software Develo pment Workers, in Human-Computer Interaction(ed.), Elsevier Science Publishers B.V., pp.365-370
- 26) 藤垣裕子、1989、ソフトウェア開発作業における負荷要因の分析、産業組織心理学研究、 Vol.3,No.1,61-68
- 27) 越河六郎、藤井亀、1987、「蓄積的疲労徴候調査」(CFSI)について、労働科学、 Vol.63,No.5,229-246

### Appendix) アンケートの自由意見

### <仕事の流動性や作業内容に関する意見>

- ・ ソフトウェア業界は他業種からみても作業体制、作業内容がかなり流動的で先が見えない為、 やりがいがある反面、ストレスもたまりやすいと思われる。
- ・・・(中略)・・仕事の量の浮き沈みが激しく、忙しい時には、徹夜仕事が続くので、それがストレスになっている。
- ・ ソフトウェア開発に対して、スキルの高い人材が決定的に不足していると思います。同じ流れの仕事が続いているのならば人材育成も容易でしょうが、開発システム毎に作業が異なることが難しくしています。
- ・ (i)最も腹立たしいことは、本人の得意分野、適性や現在の仕事の状況とは無関係に、別の新たな仕事がたらい回し的に割り込んで来ること。
  - (ii)一般のオフィスよりもはるかに合理化がおくれている。(体力本位、精神主義)
- ・ ソフトウェア開発は作業上の「けじめ」がつけにくい。上手に仕事を切分けてプライベートな時間をもつ事は至難の技である。これは日本全体がまだ余暇を持ててない事に関係ある(つまり一種の風潮のせい)のか、それともこの仕事がどうしてもそういうものなのか、考えあぐねている。(本当は自分のパーソナリティのせい?!)
- ・ 上流工程(ユーザー要件、EDP化業務の決定等の不明確さにより)が全体工程に対する割合 を多く消化してしまう。
- 設計のフェーズにストレスを感じる。
- ・ コンピュータ・システムに限らず「効率化」を行うと、作業量が増える、(作業時間を減らさない)というのが実情である。「効率化」によって余裕や余暇を増やしたいと考えて日夜努力しているというのに、人員削減や一人当りの作業量増加という汚い仕打ちを返してくる。

### <需給のアンバランスからくる問題、人材不足、教育・知識伝授の難しさなど>

- ・ システム開発に対しての供給と需要のバランスが取れていないため、教育が不足したまま仕事を進めなければならない事がこの先重要な問題として発生すると考えられる。
- ・ 特にSE不足が続いており、プロジェクトの初期が大変つらい。又、尾を引く。
- ・ もっと良い仕事をしたいと常々思うが、量をさばく(売上)のに手一杯で、かつ要員の質の面を考えると、絶望的になる時もある。技術と労力の提供があいまいになっている現状を明確にすべきではないかと思う。
- ・ 仕事は多く有るが技術者 (コーダ、プログラマより上の) が足りないと思う。そのために技術 者全体の底辺を上げて、技術者の質の向上を業界全体で考えていく必要があると考える。
- ・ 当社の場合計画的な人材育生(キャリアパス、教育等)が不十分なため、過去の経験、知識が生かされない、あるいは蓄積されないというのが大きな問題だと思う。
- ・ 現在、人に対して作業量が多すぎるために、ストレスが発生していると思う。仕事のやり方、 作業内容の効率化をもっと考えないと、ますます、ストレスが増大すると考えます。

### <納期にきつさ・仕事量に関する意見>

- ・チェック量が多い割に納期がきびしい。
- ・ ソフトウェア開発の負担は、開発環境もさることながら、日程のキビシさにある。それは日程 の見積もり精度のあまさに起因している。正しい見積もりができる仕組みの構築が必要と思う。

### <人間関係に関する意見>

・・・・(中略)・・・ユーザよりもやっかいな社内の上司とのネゴシエーションに少々疲れぎ みである。

- ・ 若いメンバーをまとめる立場として、どの様に対処していいのかわからずに頭を悩ませている 今日この頃です。
- ・ 「新人類」(部下)の扱いに困り果てている。ソフトウェア開発そのものより、その扱いに関する負荷が大きく存在しているといえる。どの職場でもそうであろうが。自分ひとりならば、 どれほど仕事がはかどるか。
- ・ システムは複数の人間で作らなければならない。大変なのは次の3つだと思う。(i)グループを まとめる。(ii)グループ全体の技術レベルを上げる。(iii)「おもしろいものを作るんだ!」という気持ちのエネルギーを保ち続ける事。

### <プログラマの地位向上に関する意見>

- ・ ソフトウェア業を本来の楽しいものにするには(戻すには)地位向上とそれに伴う質の向上 (逆か?)が必須。このままではバカばかりになってしまう。「士農工商プログラマ」から脱却できない。
- ・ソフトウェア開発者の社内での地位の低さ(はじめにハードありき←メーカー指向) ・SEAの各セミナーへ参加する場合の費用をどうしているのかアンケートしてほしい。自分で、 会社で半々?)
- ・ 近代産業としてはやく、ルールの確立が必要だと思います。 (特に、契約問題については)
- ・ ソフトウェアの重要性と重労働であることが日本社会において正しく理解されておらず、正当な評価を受けておらず、こまったものです。
- ・ 社会的に重要性が大きいわりには、環境、認識が不十分である。
- ・ 会社の管理職(部長以上)のEDP部門の人間に対する評価(能力、仕事量)が非常に低い。

### <個人的経験談>

- ・ 私は実際にテクノストレス(自律神経失調症)を最近体験しました。この調査結果に興味ある。
- ・ 現在の仕事は順調のせいかあまり苦痛は感じないが、以前は、休日出勤、徹夜等の為にかなり 精神的に不安定な時期もありました。
- しんどい。目が悪くなった。胃が痛くなった。
- ・ 私達 (ソフト技術者) は働き過ぎさせられています。

ソフトウェア開発技術者の作業負担に関するアンケート(質問表) January 1989

- Q1 まず、あなたご自身のことについて、およびあなたの生活時間についておうかがいします。
- Q1-1 性別 1. 男 2. 女
- Q1-2 年齢は?
- Q1-3 今の会社の勤続年数は?
- Q1-4 ソフトウェア開発の経験年数は?
- Q1-5 いつも仕事のある日の起床時刻は何時ですか。
- Q1-6 いつも仕事のある日の就寝時刻は何時ですか。
- Q1-7 最近一ヶ月間の残業時間は何時間ですか。
- Q1-8 最近一ヶ月間に深夜作業 (10時以降) は何回ありましたか。
- Q1-9 最近一ヶ月間に泊まりの作業(徹夜)は何回ありましたか。
- Q1-10 最近一ヶ月間に休日出勤は何回ありましたか。
- 02 次にあなたの所属する組織 (会社、学校、研究所、等) および、そこでのあなたのお仕事につい ておうかがいします。
- 02-1 あなたの会社の規模はどのくらいですか。
  - 1. 1人~9人 2. 10人~49人 3. 50人~99人 4. 100人~999人 5. 1000人以上
- Q2-2 あなたが所属する組織は、次の業務分類のうち、どれに該当しますか。
  - 1. メーカー系ソフトウェアハウス 2. ユーザー系ソフトウェアハウス
  - 3. 独立系ソフトウェアハウス 4. 計算センタ 5. コンピュータメーカ
  - 6・その他の機器メーカ 7. エンドユーザ 8. 教育・研究機関 9. その他
- Q2-3 あなた自身は、その組織内で、どんな部門に属していますか。

  - 1. 開発・製造の現場2. システムの運用・保守3. 技術調査4. 企画開発5. 教育・研究6. 営業・市場調査7. 経営・管理8. その他
- Q2-4 あなた自身の、その部門内での主なお仕事は?
  - 1. 管理 2. 調査研究 3. ソフト開発 4. 事務・営業 5. その他
- Q3 現在あなたが関係しているソフトウェアの開発の仕事(プロジェクト)について、おうかがいし ます。
- Q3-1 あなた自身は、現在、ソフトウェア開発のプロジェクトに関係しておられますか?
  - 1. している →Q3-2にすすんでください
  - 2. していない -Q5 にすすんでください
- Q3-2 いま同時に並行してすすんでいるプロジェクトの本数は?
  - ・・・複数のプロジェクトに関係しておられる方は、以下の質問において、おもな<u>1つの</u>プ ロジェクトについて答えて下さい。
- Q3-3 プロジェクトの性格は?
  - 1. 一括受託開発 2. 一部受託開発 3. プロダクト開発 4. 研究あるいは試作 4. その他
- Q3-4 開発内容は?
  - 1. 事務処理 2. 科学技術 3. プロセス制御/FA 4. OA
  - CAD/CAM
     通信
     開発支援ツール
     言語プロセッサ
  - 9. OS 10. その他
- Q3-5 その内容を具体的に1行でご説明下さい。(例:金融機関データベース合併システムの作成)
- Q3-6 その開発プロジェクトのソフトウェアは、先例(お手本にするような既存ソフト)があります か、それともまったくの新規開発ですか。
  - 1. 先例がかなりある 2. 先例が一部ある 3. まったくの新規開発である
- Q3-7 プロジェクトチームの大きさは?
  - 1. 1人 2. 2~3人 3. 4~7人 4. 8~15人 5. 16~30人
  - 6. 31~50人 7. 51~100人 8. 100人以上
- 03-8 チーム内でのあなたのボジションは?
  - 1. ジュニア・メンバ (コーダ等) 2. シニア・メンバ (アナリスト等)
  - 3. 支援スタッフ 4. プロジェクトマネージャ
  - 5. 複数プロジェクトを担当する上位の管理者 6. その他

Q3-9 あなたの今の仕事における担当工程は次のうちどこですか。担当している範囲に○をつけて下 さい。また今現在進行中の工程に◎をつけて下さい。

<通常の設計法の方>

- 1. 調査 2. 概略設計 3. 詳細設計 4. コーディング 5. 単体テスト
- 6. 総合テスト 7. メインテナンス
- <プロトタイピング手法の方>

A. 仕様の定義 B. コーディング C. デバッグ D. テスト E. 評価 評価ループは何回目ですか。

Q3-10 開発期間は?

- 2. 1~3ヶ月 3. 3~6ヶ月 4. 6~12ヶ月 1. ~1 4月
- 5. 1~2年 6. 2年以上
- Q3-11 最終納品日まであとどのくらいですか。
- 03-12 今、何割方できていると確信していますか。
- Q3-13 テスト工程は全工程の中の何パーセントをしめますか。
- 03-14 開発するソフトウェアのおおよそのボリューム (ソース・コードのステップ数) は?
  - 1. ~0.5K 2. 0.5~1K 3. 1~10K 4. 10~100K 5. 100~500K 6. 500K以上
- Q3-15 ターゲットマシンは?
  - 1. 汎用メインフレーム
     2. ミニ/スーパーミニ
     3. ワークステーション
     4. パソコン/オフコン
     5. 組み込み型チップ
     6. その他
- Q3-16 開発に用いている主な言語は?
  - 1. Cobol 2. Fortran 3. PL/1 4. Pascal 5. 4GL 6. C 7. Ada
  - 8. Lisp 9. Smalltalk 10. Prolog 11. Basic 12. PL/M 13. アセンブラ 14. その他
- Q3-17 開発に主として利用するハードウェアは?
  - 1. ターゲット 2. 専用の開発マシン 3. ターゲットと開発マシンの併用 4. その他
- Q3-18 現在の主な作業場所は?
  - 1. 自社内 2. 派遣先
- 03-19 現在主に使っているマシンはどこにありますか。
  - 1. 自分の仕事場内 2. ユーザー先
- Q3-20 現在のプロジェクトにおいて仕様変更はどのくらいありましたか。
  - 1. かなり 2. しばしば 3. たまに 4. ごくたまに 5. なし
- Q4 あなたの今のプロジェクトに対する印象や評価をおうかがいします。
- Q4-1 今のプロジェクトの納品後にトラブルやミスが発見された場合、社会に与える影響は大きいで
  - 1. 非常に大きいと思う 2. 少々大きいと思う 3. たいしたことはないと思う
- Q4-2 今のプロジェクトで最も要求されている品質は何ですか。
  - 1. 信頼性 2. 使いやすさ 3. 処理の速さ 4. その他
- Q4-3 今のプロジェクトの納期に対してどう思いますか。
  - 1. 短すぎる 2. ちょうどよい 3. 長すぎる
- Q4-4 納期はどのくらい駐通がききますか。
  - 1. 全く融通がきかない 2. ある程度融通がきく 3. 自由に延ばせる
- Q4-5 納期に間に合わなかったときはどうしますか。
  - 1. 仕様を一部削る(機能に制限を設ける) 2. 納期を延ばす 3. その他(
- Q4-6 開発環境 (デバッガー等のツールや開発支援システム等) は整っていますか。 1. 現状で十分整っている 2. ある程度融通がきく 3. 不十分である
- 04-7 マシンの使用時間は十分ですか。
  - 1. 十分自由に使える 2. 足りないときがある 3. 不足している
- あなたは今の仕事において次のうちどの面が一番難しいと感じていますか。
  - 1. 開発言語の習得 2. マシンの運用法 3. アルゴリズムの構成
  - 4. ユーザインタフェイスの設計 5. 客先との応対 6. その他
- Q4-9 あなたはコンピュータを操作するのがどのくらい好きですか。
  - 1. 大好き 2. まあまあ 3. それほど

- 04-10 今やっているプロジェクトはおもしろいですか。
  - 1. 十分面白い 2. まあまあ面白い 3. あまり面白くない 4. つまらない
- Q4-11 もっと技術的に面白い分野のことをやりたいと思いますか。
  - 1. 非常にやりたい 2. まあまあ 3. あまり
- Q5 次のA~Vまでの項目に対するあなたの評価を7段階評定でお答え下さい。 たとえば、項目Aの時間的切迫に対し、かなりストレス要因であると思われる方は、下の

t.	素にスケールのものと	こころん	しそうけ	CLSA	•			
		1	2	3	4	5	6	7
A.	納期の時間的切迫	+		-+		+	-+	-+
		全で	ほ要	あと	F	少な	かに	非ス
		<11	と因	まは	5	マつ	なな	常要
		スな	んで	りい	5	スて	りつ	に因
		141	どは	スえ	٤	141	スて	大で
		レと	スな	トな	Ł	しる	141	きあ
		ス思	+ W	LU	43	スと	レる	なる
		要う	レと	スと	え	要思	スと	スと
		因	ス思	要思	ts	因う	要思	卜思
		-100	3	因う	41	12	因う	レう

- 項目 A. 納期の時間的切迫
  - B. 毎日の仕事量の多さ

  - B. 毎日の仕事堂の多さ C. 休日や休暇のとれないこと D. 残業や夜勤の多いこと E. 自分の勉強時間のもてないこと E. 自分の勉強時間のもてないこと F. 仕事内容の難しいこと G. 雑用の多いこと

  - H. プロジェクトチーム内の人間関係 I ユーザーとの人間関係
  - I. ユーザーとの人間関係
  - J. 納品後のトラブル
  - K. 仕様がなかなか決まらないこと
  - K. 仕様がなかなか決まらないこと L. 仕事上で自由裁量で決められることが少ないこと
  - M. 自分の能力が生かされないこと
  - N. 仕事の割り振りが平等でないこと
  - の. 技術的に質の高い同僚や部下をもてないこと P. 闡尋環境が整っていないこと
  - P. 開発環境が整っていないこと
  - Q. 開発場所の居心地の悪いこと R. 昇進の機会の少ないこと S. 報節に満足できないこと

  - T. 人材育成がおろそかにされていること U. 技術の変化に追いついていくことの困難さ V. その他
- Q6 回答用紙の2枚目に心身の状態について、81の項目が書いてあります。これらを読んで、自分の 近ごろのことにあてはまるものには○を、あてはまらないものには×を、( )の中に記入して下
- 07 ソフトウェア開発の負担に関するご意見、またこのアンケートに関するご意見などがありました ら、回答用紙1枚目の下の空間に自由にお書き下さい。

ソフトウェア開発技術者の作業負担に関	する			仕事仲間とうまくいかない 腰が痛い	(	
アンケート調査 回答用紙 (2)		4	40.	IC IO 7H V	`	
Q6 あてはまるものにO、	100		41.	体のふしぶしがいたい	(	
あてはまらないものに×を記入して	下さい	,(	42.	くつろぐ時間がない	(	
			43.	考えごとがおっくうでいやになる	(	
			44.	むやみに腹がたつ	(	
	10		45.	なんとなく落着かない	(	
1. このところ食欲がない	(	)	46.	何かしようとしても、いろんな事が頭に浮ん	でき	
2. 根気がつづかない	(	)		て困る	(	
3. ちょっとした事でもすぐおこりだすことがあ	34	)	47.	家族の世話で追いまくられている	(	
4. 生きていてもおもしろいことはないと思う	(	)	48.	働く意欲がない	(	
5. ものを読んだり、書いたりする気になれない	, (	)	49.	このところ、やせて来たようだ	(	
6. やっている仕事が単調すぎる	(	)		自分が他人より劣っていると思えて仕方がな	vi	
7. 気がたかぶっている	(	)			,	
8. 動くのがおっくうである	(	)	51	よく下痢をする	1	
9. このところ毎日ねむくてしょうがない	(	í		何かでスパーッとウサばらしをしたい	1	
10. 家族と一緒にいてもくつろげない	(	)		目がかすむことがある	1	
20. 20.00	1	′		物音や人の声がカンにさわる	ì	
11. このところ頭が重い	1,17	,		気がちって困る	?	
12. 朝、起きた時でも疲れを感ずることが多い	(	)		すぐ気力がなくなる	,	
12. 初、起さた時でも扱れを思りることが多い。	(	)			,	
	(	)		仕事に興味がなくなった	1	
14. 心配ごとがある	(	,		目がつかれる	,	
15. 一人きりでいたいと思うことがある	(	)		よく肩がこる	1	
16. 理由もなく不安になることがときどきある		)	60.	眠りが浅く、夢ばかりみる	1	
17. 動作がぎこちなく、よく物を落したりする	(	)		+ 4 mm + 4 1	170	
18. このところ寝つきがわるい	(	)		すぐ風邪をひく	(	
19. ちかごろ、できもしないことを空想することが3	31(			ちかごろ元気がない	(	
20. 友人とのつきあいなどおっくうである	(	)		将来に希望がもてない	(	
01 8 8 9 7 141 1				だれかに打ち明けたいなやみがある	(	
21. 胃・腸の調子がわるい	(	)		自分の好きなことでもやる気がしない	(	
22. 仕事が手につかない	(	)		頭がさえない	(	
23. すぐどなったり、言葉づかいがあらくなってしま	= 7	,		このごろ足がだるい	(	
24. なんということなくイライラする	(	)		なんとなく気力がない	(	
25. 全身の力がぬけたようになることがある	(	)		ささいなことが気になる	(	
26. 自分がいやでしょうがない	(	)	70.	仕事でのつかれがとれない	(	
27. 話をするのがわずらわしい	(	)				
28. しばしば目まいがする	(	)		横になりたいぐらい仕事中につかれることが		
29. することに自信がもてない	(	)		家に帰っても仕事のことが気にかかって困る	(	
30. このごろ全身がだるい	(	)		今の仕事をいつまでもつづけたくない	(	
				夜、気がたってねむれないことが多い	(	
31. おもいっきりケンカでもしてみたい	(	)		毎日の仕事でくたくたにつかれる	(	
32. 朝、起きた時、気分がすぐれない	(	)	76.	生活にはりあいを感じない	(	
33. 毎日出勤するのが大変つらい	(	)	77.	なんとなく生きているだけのような気がする	(	
34. 職場のふんいきが暗い	(	)	78.	努力しても仕方ないと思う	(	
35. このところ、ボンヤリすることがある。	(	)	79.	何をやっても楽しくない	(	
36. 何ごともめんどうくさい	(	)	80.	自分の健康のことが心配で仕方ない	(	
37. 上役の人と気が合わないことが多い	(	)				
38. むねが悪くなったり、はき気がする	(	):	81.	ゆううつな気分がする	(	

## 御協力,謝々

### 野村敏次

#### 1. はじめに

「お尋ねします」を SEAMAIL に投稿してから早くも 4ヵ月経ってしまいました. 早速, 26名の方々からお手 紙をいただきましたが, 私自身, 少々忙しかったものです から, 結果を報告するのが遅くなってしまい, 申し訳あり ません. 以下に早速, 結果を報告させていただきます.

なお,アンケートにお答えいただいた方々には,この紙 面を借りて,厚く御礼申し上げます.

#### 2. 回答者のプロフィール

回答者26名の内訳は、以下の通りです.

既婚男性;23名

既婚女性; 1名

未婚男性; 1名

1000000

未婚女性; 1名

既婚男性が圧倒的に多く、しかも、平均年齢が37歳となっておりまして、私に近い(?)年代の方々の関心が高いということが判りました.血液型から分類しますと、以下の通りです.

A 型;9名

0 型;6名

B型;9名

AB型; 2名

何故か、日本における構成比とは異なった値が出ております。他の血液型は、標準構成比より小さいのに対して、B型だけが多いのです。これは、B型が多いのではなく、他の血液型の人が少ないとみなければならないと思います。やはり、積極性と気配りのB型の方々に感謝すべきだと思っております。

居住地別に見ますと、東京・神奈川・熊本がそれぞれ6名で最も多く、以下、大阪、群馬、長野、兵庫、埼玉の方々から回答をいただいております。SEAの会員は、比較的全国的に分散していることを考えると、回答は少々偏っているようにもおもえますが、余り細かいことは気にせずに結果を見てみましょう。

### 3. 回答結果

### (1)正常か異常か?

ある男の生活に関して,正常と思うか異常と思うかという問いに対しては,次のような結果となっています.

正常

; 14名 (56%)

異常

; 6名(24%)

どちらとも言えない; 5名(20%)

これは、ある男にとって非常に喜ばしい結果となっています。集まった回答数は少ないとはいえ、過半数の人々が、正常であると認めてくれたことは、ある男にとっては大きな自信に繋がるのではないでしょうか。

因みに、正常と答えた方がそれぞれの血液型の中で占める割合は、A型;44%、O型;50%、B型;56%、AB型;100%となっており、これまた日本の血液型による構成比とは正反対の結果となっています。また、B型に対するある種の偏見もこれで解消されるのではないかと考えられます。

では、異常と答えた方は、何を異常と考えておられるのでしょうか、実は、[異常]と答えた方の中には、2種類の方がおられました、ひとつは、この男の生活が[自立していない]あるいは[甘えている]といった観点からの異常であり、もう一つは、この男以上に妻に頼り切っている人から見た異常なのです。

後者の異常と答えた方の生活を、チョット紹介しましょう. [朝は妻に起こされる]、 [脱いだ背広とコートは妻が方付け、パジャマを用意してくれる]、 [食事は妻と一緒にとる] 等々... だいたいにおいて、 [妻が用意した食事を食べないなんて、許せない、おかしい] という御意見なのです.

これは、自立している・いないの観点から論ずるならば、 むしろ、正常に○をすべき人々であると言えるのではない でしょうか.

前者の異常の立場の方が、正常と考える生活については、 某HS\*社のHさんが、ある男の生活と対比しながら、克明に記述してくれましたので、Hさんの許可を得て、最後 に転載させていただきました。是非、御一読を!

#### (2) 朝食の用意は誰が?

この結果は以下の通りです.

自分でする; 6名 (23%)

奥様がする;19名(73%)

二人でする; 1名(4%)

7割以上の方々が、奥様に朝食の用意をさせているようです. ある男の生活を異常と答えた方の多くは、[自分でする]と答えております.

### (3) 洗濯をするのは?

洗濯物は誰が洗いますか?という問いに対する答えは,

自分で洗う; 2名(8%) 奥様が洗う;23名(88%) 二人で洗う; 1名(4%)

但し、自分で洗うという2名の方は、女性なので除外し ますと、ほぼ全員の方々が奥様に洗ってもらっていると言 ってよいでしょう.

### (4) 酒は誰と飲む?

一人で飲む; 7名(29%) 奥様と飲む;16名(67%) 家族と飲む; 1名(4%)

家で酒を飲まない方が2名おられ、総数は24名です. また、家族と飲まれる1名の方は独身女性の方ですので、 除いて考えますと、約70%の方々が奥様と飲んでおられ ることになります. 古手川裕子さんのコマーシャルではあ りませんが、「奥様と飲んでる...」素敵な方が多いよ うですね.

ある男の生活を異常と答えた方の、何と2/3の方が一 人で酒を飲んでおります. やはり, この辺がキーポイント になっているのではないかと、勝手に想像させてもらって おります.

#### (5) 片付けは?

食べたり、飲んだりした後の片付けは誰がするのでしょ うか. やはり, 奥様がやるか, 二人でやる方々が大半を占 めております.

自分でする; 2名(8%) 奥様がする;18名(69%) 二人でする; 6名(23%)

ちょっとしつこいですが、自分ですると答えられた2名 の方は、二人共ある男の生活を異常と答えておられます. この辺もポイントとなるのではないでしょうか.

### (6) 雨の日は車で?

雨の日は、車で送り迎えをしてくれるか、という問いに 対する回答は以下の通りです.

YES; 7人(27%) NO ; 19名 (73%)

これは、車を持っておられない方、奥様が運転出来ない 方等々すべて, NOに入っておりますが、4人に1人は車 で送り迎えをしてもらっているということは、意外と多い とは思いませんか?

### (7)煙草は?

煙草を吸わない方を除くと,以下の結果となります. 家の中で煙草が吸える;13名(68%)

以下の通りです. 家の中で煙草が吸えない;6名(32%)

意外と多くの方々が吸っているのに驚いています. 家族 のためにも,程々に! (こんなことを言える立場ではあり ません. わかっているけどやめられないのです)

### (8) 着替えの手伝いをしてもらう?

YES; 2名(8%) NO : 24名 (92%)

大半の方がNOと答えておられます. 神戸のMさんや大 阪のUさんの話からすると、YESの方がもっとおられる と思っていましたが、結果は御覧の通りです. 皆さんの老 後は安泰です.

#### (9) 出張や旅行の準備は誰がする?

出張や旅行の準備は、主として誰がするかという問いに 対する回答は以下の通りです.

自分でする:15名(58%) 奥様がする; 8名(31%) 二人でする; 3名(11%)

6割近い方々が、主として自分で支度されているようで す.

### (10) 自分の持ち物のしまってある場所を全部知っ ている?

YES;14名(54%) NO ; 12名 (46%)

約半々といったところでしょうか. これは、前間の出張 や旅行の準備を誰がするかと,高い相関があります.しま ってある場所を知っている人(14名)の内, 12名の方 は出張・旅行の準備を自分でしており、残りの2名の方は 奥様と一緒に準備しております. 問題は、しまってある場 所は知らないが、出張・旅行の準備を自分でなさっている 3名の方です. しまってある場所を全部は知らないが, 準 備するに必要な物については知っているということでしょ

## 4. SEAMAILに掲載された「お尋ねします」(

次のストーリーは、ある男の家庭におけるごく標準的な 1日を、特に夫婦間の言動に限って記述したものです。こ れを読んで、以下の問に答えて下さい. (なお、これはフ ィクションであり、特定のモデルは存在しません)

6時半,「オキロ!,オキロ!」という目覚まし時計の 声で跳び起きる. ウン, 未だ昨夜の酒が残っているようだ と感じつつ階下へ降りる. 妻が朝食の用意と子供の弁当を 作っている. 「オハヨ!」と声を掛け、トイレと洗顔を済

ませ、再び2階へ、整理箪笥の決められた場所から下着/ 靴下/ハンカチを取り出し、洋服箪笥からワイシャツ/ネ クタイを適当に選んで着替えを終えると既に7時、食堂の 椅子に腰掛けるとすぐ、熱い御飯と味噌汁と惣菜を妻が運 んでくる、食事の終わるころ、妻が「コーヒーにする?そ れとも御茶」と問う、「二日酔いでコーヒー、ついでに胃 薬」と答える、コーヒーと薬と水が用意される、食後、煙 草を一服して、「行ってきます」と声をかけて玄関へ向か う、妻が門の外まで一緒に出てきて、「行ってらっしゃい 」と見送る、「午後雨が降るっていうけど、早めに電話ち ょうだい、迎えにいくから」と妻、「わかった」と一言、

会社から帰るのは、飲まない日でも夜9時前後、幸か不 幸か雨は降らずに歩いて帰宅. 「ただいま」と居間に顔を 出してから2階に上がり、背広とコートをハンガーに掛け、 下着を持って階下に降りると、「御飯は?」と妻が問う. 「未だ」と答えて風呂へ向かう. シャツや下着や靴下を洗 濯篭に投げ入れ風呂に飛び込む. 数分すると妻が風呂場に 顔を出し、「何か飲む?」と聞く、「寒いから日本酒がい いな」と気楽に答える. 風呂からでて居間に戻ると, 熱澗 とつまみが用意されている. テレビ映画を観たり、妻の話 を聞きながら酒量が増していく. 11時近く, 「もう,酒 いらん」というと、「御飯どうする?」と妻が問う、「い らん」と一言. 「一口食べれば良いのに」と言いつつ、妻 が片付け始める. 「明日からの出張は3日間よね」「うん 」「2階の部屋に下着2組とシャツ出してあるから確認し て鞄に入れてね」「わかった」「明日はいつもと同じ時間 でいいの?」と妻が聞く、「9時の新幹線だから30分位 早くでる」「じゃ, 6時40分頃食べられればいいのね? 」「うん」炊飯器のタイマーをセットする音が聞こえてく る. 11時半過ぎ、「寝るよ!」と一声かけて2階へ向か い、用意されたものを鞄に詰める.後から上がってきた妻 が「忘れものしないようにね」と声をかける.

## 5. お尋ねします (正常編) (Hさんの書かれたもの: 原文のまま)

次のストーリーは、ある男の家庭におけるごく標準的な 1日を、特に夫婦間の言動に限って記述したものです。( なお、これはフィクションであり、特定のモデルは絶対に 存在しません)

6時半,「オキロ!,オキロ!」という目覚し時計の声で跳び起きる.ウン未だ昨夜の酒が残っているようだと感じつつ階下へ降りる.奥様はまだ気持ち良く寝ておられる.「さっ,コーヒー,コーヒー」とあわただしい私の一日が始まる.良い香りのするコーヒーが入った時,「オーイ,コーヒー入ったよう」と,奥様の機嫌を損ねないように優しく声をかける.コーヒーを飲みながら朝刊に一通り目を

通すと既に7時. そろそろ奥様の起きてこられる音がしだした. その間, 味噌汁のダシを用意し, 具を冷蔵庫の中から選び, 準備する. 無言のまま, 奥様が起きてこられる. なるだけ目を合わせないようにしながらダイニングを出て, トイレ, 着替えをすます. ダイニングに戻ると相変わらず無言のままで子供の弁当を作られている. 味噌汁の準備は出来ているようなので, 自分でよそおい, テーブルにつく. 朝食が終わると, 流しまで汚れた食器を運び, すぐ会社に向かう. 玄関で靴を履いていると, 「今日, 帰りは何時位になるの?食事がいらない時は早めに電話して」と奥様. 「はい」と一言. 外に出ると, 急いで煙草に火をつけ, 大きく吸い込む. と, 窓越しに, 「午後雨が降るっていうから, ちゃんと傘持っていってよ」と奥様. またも, 「はい」と一言.

会社から帰るのは、飲まない日でも夜10時前後、幸か不幸か雨は降らずに歩いて帰宅、「ただいま」と居間に顔を出してから着替えをすます。「早く帰るのたまにしかないんだから、先にお風呂に入って子供を入れて」と、奥様のお言葉、「はい」と言って下着を用意し、風呂に飛び込む、数分すると子供達が大騒ぎしながら飛び込んできた。たまにはゆっくり風呂につかり、大きな声で歌でも歌いたいものだ。風呂から上がると奥様はテレビを御覧になっている。なるだけ音を立てないようにして、夕食を温めなおし、自分でテーブルに運ぶ。自分で水割を作り、一杯、二杯と飲んでいると、「まだ飲んでの、休肝日はいつ!」と、奥様の声、「後一杯だけ」と恐る恐る言うと、また無言のままテレビを見ておられる。食事が終わる頃、奥様は「そろそろお風呂に入ろうかしら」と言いながら、奥様は浴室に向かわれる。後片付けをしながら、もう一杯だけと盗み酒。

奥様が風呂から上がられると「明日からまた出張よね, 3泊だったっけ」「はい」帰る前にはちゃんと電話してよ, 食事の準備があるんだから」「はい」. 「じゃあおやすみ 」と奥様は疲れたように階段を登っていかれた.

一通り出張の準備をすませて寝室に入っていくと, 奥様が「静かにして,子供が起きるでしょう」と声をかける.

#### 6. 筆者の感想

筆者は問題を出し、皆様の回答を集計しただけであり、 この結果に対して、云々言える立場ではありませんが、一 言感想を述べさせていただきます。

人には様々な生活があり、その中で人々は、それなりに楽しんでいるのではないでしょうか、筆者の書いた、ある男の生活も、Hさんの書かれた男の生活も、それなりに正常な生活だと思います。いろいろな人がいて、いろいろな生活形態がある。だから、世の中は面白いといった、悟りにも似た境地になっております。御協力いただいた皆様方には、心より謝謝。

### 1989年度総会報告

今年度 SEA 年次総会は、さる6月14日 (ソフトウェア・シンポジウム'89 第1日の夕刻、昼間の技術セッションが終わって夜の情報交換パーティが開かれる合間の時間を利用して)、東京虎の門パストラルで開催された。

会則(第24条)には、本来、「通常総会は毎年1回5月に開催する」となっていたが、3月度の幹事会において、「せっかく毎年のメイン・イベントであるソフトウェア・シンポジウムを6月に行い、そこには多くの会員が集まるのだから、総会も同時に開くのがリーズナブルである」という意見が多数を占めたため、今年度はあえて会則を曲げて総会を6月に移し、併せて会則の条文を「5月または6月に開催する」と変更することとなったものである。

出席者は45名,他に委任状提出者419名で,合計464名.定足数107名を大幅に上回った.

総会では、まず前年度の会計決算(表1および表2)が報告され、満場一致で承認された、引き続き今年度予算案 (表3)が、これも異議なく可決された、また、新年度の役員人事については、前年度幹事会において検討された候補者リスト(次ページ)が提示され、原案通り承認された。

表1 収支計算書 自 1988年4月 1日 至 1989年3月31日

支出の部	
人件費	203,920
事務所費	4,290,433
印刷費	5,194,300
通信費	3,098,145
会議費	345,040
広報費	160,380
消耗品費	293,045
雑 費	2,065,386
当期収支差額	3,109,158
合 計	18,759,807
収入の部	TARRET !
新入会費	3,273,000
更新会費	5,313,000
賛助会費	3,400,000
寄付金	4,625,691
雑収入	2,148,116
合 計	18,759,807

表2 貸借対照表 1989年3月31日現在

6,721,854
1,315,000
2,400,000
10,436,854
S
669,000
2,000,000
7,767,854
3,109,158)
10,436,854

この収支計算書は、公益法人会計基準に示されているような収支ベースのものではなく、損益ベースによって作成してある(企業会計の損益計算書に該当するものである).

表3 予算書 自 1989年4月 1日 至 1990年3月31日

支出の部	
人件費	2,400,000
事務所費	4,800,000
印刷費	6,000,000
通信費	3,600,000
会議費	600,000
広報費	240,000
消耗品費	360,000
雑 費	1,200,000
当期収支差額	1,660,000
合 計	20,860,000
収入の部	
新入会費	3,600,000
更新会費	5,460,000
賛助会費	3,600,000
寄付金	5,800,000
雑収入	2,400,000
合 計	20,860,000

### 1989 年度 ソフトウェア技術者協会役員名簿

代表幹事	岸田 孝一	(株) SRA
常任幹事	臼井 義美	日本電子計算(株)
	久保 宏志	富士通(株)
	熊谷 章	(株) PFU
	佐藤 千明	長野県協同電算(株)
	藤野 晃延	富士ゼロックス情報システム(株)
	松原 友夫	日立ソフトウェア・エンジニアリング(株)
	吉村鉄太郎	(株)管理工学研究所
幹事	青島 茂	富士ゼロックス情報システム(株)
	天地 学	カシオ計算機 (株)
	飯沢 恒	三菱電機東部コンピュータシステム
	稲田 博	(株) 第一ソフテック
	岡田 正志	日本電気ソフトウェア(株)
	岩田 康	(株) SRA
	落水浩一郎	静岡大学
	片山 禎昭	日本システム(株)
	川北 秀夫	(株) ジェーエムエーシステムズ
	杉田 義明	(株) SRA
	武田 知久	日本システムサイエンス(株)
	田中慎一郎	(株) SRA
	玉井 哲雄	筑波大学
	中来田秀樹	(株) ニコンシステム
	中園 順三	(株) 富士通ビー・エス・シー
	中野 秀男	大阪大学
	西尾 出	日本ナレッジインダストリ(株)
	野村 敏次	日本電子計算(株)
	野村 行憲	(株) 岩手電子計算センター
	針谷 明	(株) 山一コンピュータ・センター
	平尾 一浩	ヒラタソフトウェアテクノロジー
	深瀬 弘恭	(株) アスキー
	藤本 司郎	日本アイビーエム(株)
	北條 正顕	(株) シスプラン
	細野 広敏	(株) エム・ケー・シー
	盛田 政敏	(株) ケーシーエス
会計監事	辻 淳二	(株) 辻システム計画事務所

公認会計士

吉村

成弘

### 第 10 回

### ソフトウェア信頼性シンポジウム

### 発表者募集

主催:高信頼性ソフトウェア研究会 ソフトウェア技術者協会関西支部 電子情報通信学会FTCS研究専門委員会

ソフトウェア開発の信頼性向上に関するさまざまなトピックスをめぐって、インフォーマルな(しかし、それだからこそ核心を笑いた)議論を展開できるユニークな1つの場として、毎年春と秋に開催されてきたこのシンボジウムが、早いものでついに第10回目を迎えることになりました.

#### 開催要領

1.期日: 1989年 10月 24日 (火) ~ 25日 (水)

2. 会場: 大阪ガーデンパレス (JR 新大阪駅から徒歩10分)

3. 定員: 100 名

4. 費用 (予定): 会員 (SEA または電子情報通信学会) 7,000円

一般 10,000円, 学生 3,000円

5. スタッフ:

世話人: 臼井義美(日本電子計算) 浦野義頼(国際電信電話)

プログラム委員: 鳥居宏次 (大阪大学)

ほか高信頼性ソフトウェア研究会会員一同

#### 6. 応募要領:

この春の第9回シンポジウムには、90人を越える方々に御参加いただき、

- ・定量的品質管理のメカニズム
- バグ・データの収集と分析
  - ・ソフトウェア信頼性推定モデル
  - ・ソフトウェア品質設計上の諸側面
  - ・ソフトウェア品質改善努力の事例
  - 高品質ソフトウェアへのアプローチ

という6つのセッションにおいて、合計16件の発表をもとに熱心な討論が行われました.

今回も、みなさんからの事例発表を中心に、魅力あるプログラムを構成したいと考えています。たとえば、ソフトウェアの信頼性向上を目指した現実的努力の事例(成功例、失敗例いずれでも)、信頼性の定量的把握のための試みや提案、いろいろな開発技法やツールの品質面での効用に関する経験的評価、プログラム・バグの分類や原因追跡など、何でも結構です。身近な経験にもとづく具体的な討論の材料を出していただければ幸いです。

なお、このシンポジウムはあくまでもインフォーマルな性質のものですから、形式の整った論文を書いていただく必要はありません。審査はアブストラクト(発表要旨)にもとづいて行い、シンポジウム当日は発表用のメモと OHP さえあれば結構です。その意味で、どなたでもお気軽に御応募ください。

発表希望者は、裏面の応募用紙に必要事項を記入の上、 1989年 8月末日までに、郵便または FAX で、世話人の浦野宛お送り下さい. 応募者多数の場合には、プログラム委員会で応募内容を検討した上、プログラム全体でのバランスを考慮して、採否を決めさせていただきます.

審査結果の応募者への通知と,確定したプログラムにもとづく一般参加者の募集は,9月中旬を予定しています.

### 申込み先:

〒356 埼玉県上福岡市大原 2 - 1 - 15

KDD 上福岡研究所

次長 浦野義頼

TEL: 0492 - 66 - 7302 FAX: 0492 - 66 - 7510

送付先:

申込日付: 月 日

〒356 埼玉県上福岡市大原 2 - 1 - 15 KDD上福岡研究所 浦野義頼

Fax: 0492 - 66 - 7510

第 10 回 ソフトウェア信頼性シンポジウム 発表応募用紙

氏	名:	(ふりがな)	MY.
年	龄:	歳) 性別: 日男日女 血液型:	
種	別:□ SEA会員	□ 電子情報通信学会会員 □ 一般 □ 学生	
会社	社名:		
部	門:	役職:	
住	所: (〒 )_		2 . 1
TEI	L: ( ) - (	)-(  )内線(  )	
発	表の題目:		
- No	+		

### 発表の要旨:

紙面が足りない場合は別紙を追加していただいてかまいません.また、関連する論文または技術資料などがありましたら、添付してください.

## 実践的ソフトウェア開発環境に関する集中討論

参加者募集

主催:ソフトウェア技術者協会



今度で第5回目を迎える恒例の SEA 環境ワークショップは、思い切って北の港町・釧路に飛び、また、運営のスタイルを、従来の単なるパネル討論形式から、参加者全員で開発環境改善のための新しい提案を考える「プロポーザル・ワークショップ」に模様変えして、気分を新たに、熱気あふれるディスカッションを展開したいと考えています。

#### 開催要領

- 1.期日: 1989年11月29日(水)~12月1日(金)
- 2. 会場: 厚生年金釧路市福祉会館・小ホール (北海道・釧路市)
- 3. 宿泊: 釧路パシフィックホテル
- 4. 定員: 25名
- 5. 費用 (予定): SEA 会員 55,000円, 一般 70,000円

ただし、現地集合・現地解散とし、上記の費用には、資料代、会場費および期間中の宿泊費を含みます。

6. スタッフ:

実行委員長: 深瀬弘恭 (アスキー)

プログラム委員長: 篠田陽一(東京工業大学) 平尾一浩(ヒラタ・ソフトウェア・テクノロジー) プログラム委員(交渉中): 岡本隆一(ケーシーエス) 岸田孝一(SRA) 熊谷章(PFU&富士通) 佐藤千明(長野県協同電算) 中野秀男(大阪大学) 野村行憲(岩手電子計算センター)

7. 討論プログラム (詳細は裏面):

11/29 (水) 午後 Session-1: マイクロ・メインフレーム・リンクの確立をめざして

夜 Reception

11/30 (木) 午前 Session-2: ハイパーテキストの光と影

午後 Session-3: ソフトウェア工学は現場に生かせるか?

夜 BOF: 企業間コンピュータ・ネットワークの構築と活用

12/01 (金) 午前 Session-4: ツール・インテグレーションの理想と現実

それぞれのセッションでは、まず2~3人の方に「叩き台」となるような提案をお願いし、それらをもとに、全員でそのセッション・テーマに則した現実的な方策を探る討論を行ないます。

#### 8. 申し込みおよび参加方法:

今回のワークショップでは、比較的限られた人数で突っ込んだ討論をしたいと考えています.

参加を希望される方は、裏面の各セッション詳細説明をお読みになった上で、そのうちのどれか1つのセッション・テーマを選び、御自分のお考え(そのテーマに関する意見または提案)をA4 版 $2\sim3$  ページ程度のポジション・ペーパにまとめ、実行委員長宛に郵便またはFAXまたは電子メイルで8 月末日までにお送りください。

プログラム委員会で内容を審査の上、9月下旬までに正式参加者を決定して、みなさんに結果をお知らせするとともに、詳細な参加案内をお送りします。

ボジション・ペーパ送付先

〒107-24 港区南青山 7-1-5 コラム南青山ビル

(株) アスキー 技術企画室

深瀬弘恭

TEL: 03-486-1207 FAX: 03-486-0373 E-Mail: hiro-f@ascii.junet

\*\*\*\*\* 裏面に各セッション・テーマについての解説があります \*\*\*\*\*

### 第 5回 SEA 環境ワークショップ

### セッション・テーマおよび討論の進め方

#### Session-1: マイクロ・メインフレーム・リンクの確立をめざして Chair: 中野 秀男 (大阪大学)

MML (マイクロ・メインフレーム・リンク) というキーワードは、しだいに一般化してきたワークステーションや LAN を、 なんとか既存のメインフレームを中心とするシステムの秩序を破壊せずに現場に持ち込もうという、いささか苦しまぎれのアイデ アを象徴しているように感じられます.概念としては理解できても,実際に MML を具体化しようとすると,あまりにも多くの技 術上の難問とそれらを解決するためのたくさんの選択肢があり、とくにメインフレーム・サイドからのアプローチを試みようとし た場合には、出口のない迷路に入りこむ危険が高いようです.

このセッションでは、いくつかのドメインを想定し、また初心者からウィザードまで多彩なユーザ層を考えて、システムの機能 やセキュリティ、管理運用、故障対策などの諸問題に関して、MML 具体化の正しい選択肢を見出すガイドラインを設定したいと 考えています.

#### Session-2: ハイパーテキストの光と影

#### Chair: 熊谷 章 (PFU & 富士通)

HyperCard on Mac の華やかな登場をきっかけとして、ハイパーテキスト技術の応用が、いま、要求分析や設計、ドキュメン テーション、さらにはメインテナンスといったソフトウェア開発のさまざまな側面で脚光を浴びているようです。ある意味で、そ れは、オブジェクト指向プログラミングの正しい発展型であり、知識ベース技術とうまく組み合わせれば、現在のソフトウェア開 発が抱えている多くの問題点が解決できるだろうという楽観的な予想もあります。一方、批判的な目で見れば、スパゲティ風のハ イパーテキスト・ネットワークは、GoTo 文を乱用した昔のプログラムと同じものであり、もう一度構造化革命を発動する必要が ありそうです.

このセッションでは、具体的なケースを対象に、ソフトウェア開発においてハイパーテキスト技術をどのように利用すべきか、 それによってもたらされるメリットや付随する問題点は何かを、さまざまな角度から検討してみたいと考えています。

#### Session-3: ソフトウェア工学は現場に生かせるか?

### Chair: 岸田 孝一 (SRA)

ソフトウェア工学が生まれてからすでに20年たち、その間にいろいろな技法やツールが提案されてきましたが、いまだにアプ リケーション開発の現場には普及していません.最近では,ワークステーションの高性能化とともに,これらのツールが"CASE" という新包装を身にまとって再登場しつつありますが、はたして、その効用はどうでしょうか? 「再利用」や「プロトタイピン グ」、「オブジェクト指向」などのキャッチフレーズも、やたらかけ声ばかり大きくて、実践がともなっていない感じがあります。 こうした状況は,環境のハードウェア的側面が整備されれば,自然に改善されるのでしょうか?』それとも,もっとドラスティッ クな打開策が必要なのでしょうか?

このセッションでは,開発環境と人間(あるいは組織)との関係を話題の中心にすえて,ソフトウェア工学をアプリケーション 開発の現場で実践することの今後を占ってみたいと考えます.

#### BOF: 企業間コンピュータ・ネットワークの構築と活用 Chair: 深瀬 弘恭 (アスキー)

ちかごろ、システム・インテグレーションという言葉をよく聞くようになりました。コンピュータメーカをはじめとして、ソフ トウェアハウスも計算機センタも、みな同じ方向をめざしているようです。ところが、どの会社をとってみても、そのために要求 されるすべての技術を自社内にそなえているわけではありません、当然、ゼネコン形式(外注化)あるいは、コンソーシアム形式 (企業連合)で仕事を受けることになります.ここで当然必要になってくるのは、企業間ネットワークや広域ネットワーク(いい かえればメトロポリタン・ネットワーク)の確立です.

この BOF では、「そうしたネットワークがもし形成できたら、われわれの生活環境はどのように変化するか?」という話題を 酒の肴にしたいと考えています。(ホンネをいえば)そうしたネットワークをうまく使って、人月ベースの発注に対抗する闇のカ ルテルを組もうということをねらっているのです (!?).

#### Session-4: ツール・インテグレーションの理想と現実

#### Chair: 岡本 隆一 (ケーシーエス)

ソフトウェア開発プロセスを効果的に支援するためには、開発環境の中にあらかじめ多種多様なツールを用意し、時と場合に応 じてそれらを有機的に組み合わせて利用する必要があるでしょう.しかし、もともと特定の個別ニーズに対応して別々の場所で開 発され、当然のことながら各種のインタフェイスが異なっている複数のツールを、1つの環境としてインテグレーションすること は、技術的にきわめてむずかしい問題を数多く含んでいます。また、開発環境は、それ自体が1つの生きたシステムであり、開発 対象ソフトの種類やプロジェクトの規模といった利用条件の変化や、目まぐるしいハード/ソフト技術の進歩に適応して、時間と ともに進化して行かなければなりません.

このセッションでは、ツール間インタフェイスの標準化や環境プラットフォームの統一といった技術課題の検討とあわせて、環 境の進化や適応に関するマネジメント上のガイドラインの確立を,参加者全員の知恵で考えてみたいと思います.

石川

間積

岐阜

三重

滋賀

奈良

兵庫

愛媛

徳島

高知

香川

鳥取

島根

広島

山口

福岡

大分

熊本

長崎

宮崎

佐賀

沖縄

韓国

米国

豪州

鹿児島=

岡山 =

=

愛知 = 和歌山=

京都 = 大阪 =

### SEA会員状況(平成元年7月18日現在)

正会員	1666名	< 身	女	分布>
賛助会員		男	=	1580
		女	=	86

### 正会員の勤務地および居住地域分布

3 19

2

56

2

10

21

226

4

37

2

6

0

1

0

1

2

5

0

18

6

0

1

1

5

1

1

4

34

19

8

45

3

4

11

32

168

14

67

2

6

0

0

1

1

5

0

20

6

33

0

1

5

1

貝の動物地やみび沿江地域力和			5 正地域力和	< 年 節 分
		勤務地域	居住地域	20以下 = 0
北海道	=	9	8	$20_{24} = 37$
青森	=	0	0	$25_{29} = 318$
	=	1	1	30_34 = 454
宮城	=	6	5	
	=	10	10	
福島	=	1	1	$40_{-}44 = 285_{-}$
秋田	=	1	1	$45_{-}49 = 117$
新潟	=	7	7	50_54 = 32
栃木	=	6	6	55_59 = 19
群馬	=	5	4	60以上 = 16
	=	12	14	
				<血液型分布>
177	=	21	122	
1 1	=	19	100	A型 = 644
NCN.	=	882	529	〇型 = 484
神奈川		135	298	B型 = 363
山梨	=	1	1'	AB型= 175
長野	=	36	37	
富山	=	3	3	TANKA FIA MATA
垣土	=	4	3	替肋牵骨牵补名

### 賛助会員会社名

ジェーエムエーシステムズ、日本システム、リコーシステ ム開発, SRA, 辻システム計画事務所, PFU, 近畿日 本ツーリスト,東電ソフトウェア,セントラル・コンピュ ータ・サービス, 日本能率コンサルタント, 構造計画研究 所,ケーシーエス,サンビルド印刷,日本システムサイエ ンス,富士ゼロックス情報システム, IN情報センター, 富士通、エムテイシー、クレオ、情報システムサービス、 インターナショナル・データ・リサーチ、日立ビジネス機 器、アトラス情報サービス、三菱電機セミコンダクトソフ トウェア, 伊藤忠エレクトロニクス, 三菱電機, 富士通B SC, 千代田製作所, 村田技研, ニコンシステム, 日本ビ ジネスデータプロセシングセンター, 阪神計算センター, 昭和電工コンピュータサービス, エヌ・ティ・ティ・シス テム、日本情報システムサービス、ヒラタ・ソフトウェア ・テクノロジー、日本エム・アイ・シー、日本電気ソフト ウェア、ソニー、コマス、三菱電機コントロールソフトウ ェア, インターフィールド・システムズ, ジャストシステ ム,マイクロリサーチアソシエイツ,松下ソフトリサーチ, シャープ、東洋エンジニアリング、日本ユニシス・ソフト ウェア, 清鳳オフィスメーション, 新日鉄情報通信システ ム,テックシステムズ,熊本電子計装工業,NTT九州技 術開発センタ, 立石ソフトウェア, 三菱電機関西コンピュ ータシステム、日立エンジニアリング、スインク、テスク



ソフトウェア技術者協会

〒102 東京都千代田区隼町2-12 藤和半蔵門コープビル505 TEL.03-234-9455 FAX.03-234-9454