SEAMAIL

Newsletter from Software Engineers Association

Volume 6, Number 12 1991

目　次

編集部から		1
ソフトウェア･プロセスに関する調査 (第 1 次集計結果)	野村敏次･岸田孝一	2
I. 会社の概要とソフトウェア･プロセス		3
II. プロジェクトの特性とソフトウェア･プロセスの実態		8
III. ソフトウェア･プロセスの問題点		16
IV. 個人的な背景，教育，経験		19
我社におけるプロセス成熟度の調査と考察について (独立系を中心とした日本企業プロセス成熟度調査とその後	佐藤雄一	22
ソフトウェア特許およびユーザインタフェイス著作権に関する LPF からのアピールについての会員アンケート回答集計	編集部	37
Call for Papers & Participation		
第 10 回夏のプログラッミング･ワークショップ		42
The 15th International Conference on Software Engineering		44
The 3rd Annual Symposium of the IAKE		45

ソフトウェア技術者協会
Software Engineers Asociation

ソフトウェア技術者協会(SEA)は，ソフトウェアハウス，コンピュータメーカ，計算センタ，エンドユーザ，大学，研究所など，それぞれ異なった環境に置かれているソフトウェア技術者または研究者が，そうした社会組織の壁を越えて，各自の経験や技術を自由に交流しあうための「場」として，1985年12月に設立されました．

その主な活動は，機関誌SEAMAILの発行，支部および研究分科会の運営，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，および内外の関係諸団体との交流です．発足当初約200人にすぎなかった会員数もその後飛躍的に増加し，現在，北は北海道から南は沖縄まで，1000余名を越えるメンバーを擁するにいたりました．法人賛助会員も約60社を数えます．支部は，東京以外に，関西，横浜，長野，名古屋，九州の各地区で設立されており，その他の地域でも設立準備をしています．分科会は，東京，関西，名古屋で，それぞれいくつかが活動しており，その他の支部でも，月例会やフォーラムが定期的に開催されています．

「現在のソフトウェア界における最大の課題は，技術移転の促進である」といわれています．これまでわが国には，そのための適切な社会的メカニズムが欠けていたように思われます．SEAは，そうした欠落を補うべく，これからますます活発な活動を展開して行きたいと考えています．いままで日本にはなかったこの新しいプロフェッショナル・ソサイエティの発展のために，ぜひとも，あなたのお力を貸してください．

代表幹事： 熊谷章

常任幹事： 市川寛　落水浩一郎　岸田孝一　中野秀男　深瀬弘恭　松原友夫　山崎朝昭

幹事： 飯沢恒　筏井美枝子　上原憲二　臼井義美　奥山充　片山卓也　亀田繁　窪田芳夫　小林俊明
杉田義明　田中一夫　玉井哲雄　鳥居宏次　中谷多哉子　中山照章　野村敏次　野村行憲　浜野善吉
林香　平尾一浩　藤野晃延　二木厚吉　堀江進　本位田真一　盛田政敏　山崎利治　渡邊雄一

会計監事： 辻淳二　吉村成弘

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)：野々下幸治
教育分科会(SIGEDU)：杉田義明　中園順三
ネットワーク分科会(SIGNET)：小林俊明
調査分科会(SIGSURVEY)：岸田孝一　野村敏次

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：藤野晃延　北條正顯　野中哲　松下和隆
長野支部：市川寛　佐藤千明　堀内広水
名古屋支部：筏井美枝子　岩田康　鈴木智　平田淳史
九州支部：平尾一浩　藤本良子

賛助会員会社： ＮＴＴソフトウェア研究所　ＮＴＴ九州技術開発センタ
ＰＦＵ　ＳＲＡ　アスキー　エイ・エス・ティ　エスケーディ
オムロンソフトウェア　カシオ計算機　キャノン新川崎事業所
クレオ　さくらケーシーエス　コグノスジャパン　サン・ビルド印刷　システムラボムラタ
ジェーエムエーシステムズ　ジャストシステム　セイホーソフトウェア
セントラル・コンピュータ・サービス　ソフトウェアコントロール　ダイキン工業　テクノバ
テックシステムズ　ニコンシステム　ニッセイコンピュータ　ヒラタ・ソフトウェア・テクノロジー
ファコム・ハイタック　フクダ電子　ムラタシステム　リコーシステム開発
リバテイーシステム　安川電機　古河インフォメーション・テクノロジー
構造計画研究所　三菱電機セミコンダクタソフトウェア
三菱電機メカトロニクスソフトウェア　三菱電機関西コンピュータシステム
三菱電機東部コンピュータシステム　三菱電機北伊丹製作所　松下ソフトリサーチ　新日鉄情報通信システム
新日本製鉄エレクトロニクス研究所　池上通信機　中央システム　辻システム計画事務所
東芝アドバンストシステム　東電ソフトウェア　東北コンピュータ・サービス　ＳＲＡ東北　日本ＮＣＤ
日本データスキル　日本ユニシス・ソフトウェア　日本情報システムサービス
日本電気ソフトウェア　日立エンジニアリング　富士ゼロックス情報システム　富士写真フィルム
富士通　富士通エフ・アイ・ピー　富士通ビー・エス・シー　クレオアールアンドディ　ＳＲＡ北海道　（以上61社）

ＳＥＡＭＡＩＬ　Ｖｏｌ．６，Ｎｏ．１２　１９９２年６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）
〒１６０　東京都新宿区四谷３－１２　丸正ビル５Ｆ
TEL: 03-3356-1077　FAX: 03-3356-1072
印刷所　サンビルド印刷株式会社　〒１６２　東京都新宿区築地町８番地
定価　５００円（禁無断転載）

編集部から

☆

桜の花が散らないうちにと考えていたSeamail Vol.6の最終号ですが，ISFSE & ICSEツアーやら，ソフトウェア･シンポジウム開催の準備やらにとりまぎれている間に，いつのまにか梅雨に入ってしまいました．

☆☆

この号には，昨年夏に行ったソフトウェア･プロセス実態調査の第1次集計結果を収録しました．この結果も，じつは昨年の秋には出ていて，野村さんが北京のBICSで発表したりしていました．編集部としては，もう少し突っ込んだクロスの分析などをしてからのレポートを考えていたのですが，SIG/SURVEYのボランティアの方々が忙しく，なかなか分析がまとまりそうにないので，とりあえず単純集計結果だけをお届けします．

この数字を御覧になって興味を持たれた方は，今からでも結構です．SIG/SURVEYに参加･御協力ください．次号には，欧米での調査結果の単純集計を報告します．比較して見るとなかなか面白そうです．

☆☆☆

昨年暮れに名古屋の南山大学で開かれたワークショップに参加された佐藤雄一さんから，当日のプレゼンテーションに手を加えた力作のレポートが届きました．やはり，ソフトウェア･プロセスの実態についての，興味深い分析です．上の調査結果とあわせてお読みください．

☆☆☆☆

前々号に載せたLPF（プログラミングの自由のための連盟）からの2つのアピールについては，会員各位の声を募集しましたが，思ったより少なく，26件しか回答がかえって来ませんでした．

しかし，外部ではいろいろ反響を呼んだようです．たとえば，日経ンピュータ(4月6日号)では，編集長の松崎さんがコラムで好意的に取り上げてくださいました．それが目にとまったからでしょうか，特許庁のお役人やコンピュータ･ソフトウェア著作権協会の事務局から，バックナンバー購入の申込みがありました．

日経コンピュータでは，その後6月15日号で，ソフトウェア知的財産権の特集を組んでいます．

☆☆☆☆☆

さて，次号はいよいよVol.7です．先日の総会で決まった新幹事名簿，予･決算の数字などをお知らせします．発行は7月上旬には必ず!!!

☆☆☆☆☆☆

お詫びと訂正

SEAMAIL Vol 6-12号の

佐藤雄一さん(P22 -)の文章に挿入違いがありました．

お詫びして下記のように訂正いたします．

正しい順序: P32 -> P34 -> P35 -> P33 -> P36

編集部

ソフトウェア･プロセスに関する調査
第１次集計結果

調査分科会 (SIG/SURVEY)

野村 敏次 (日本電子計算)　岸田孝一 (SRA)

調査のねらい

SEA では，昨年の夏，アメリカの非営利調査研究機関 Rocky Mountain Institute of Software Engineering (略称 rMISE. 代表: William Riddrle 氏) と協同で，表記のアンケート調査を行った．この調査は，現時点におけるソフトウェア･プロセス（すなわち，ソフトウェア開発･保守工程）の実態を明らかにし，さまざまな環境条件の違いにもとづく状況比較を可能にするとともに，これからのプロセスの改善を考えるさいの基礎を確立することを目的としている．

日本における調査は，SEA 会員を中心に約 3000 通のアンケートを発送し，500 通弱の回答を得た．rMISE では，ACM/SIGSOFT のメンバーを中心に，欧米で同様なアンケート調査を実施し，やはり数百通の回答を得ている．

結果の集計･分析は現在まだ進行中であり，最終結果が出るにはしばらく時間がかかりそうであるが，とりあえずここでは，日本における調査結果の単純集計だけを報告する．なお，結果表中の数字はすべて % である．

全体の構成

調査は 4 つの部分から成っている：

I. 会社の概要とソフトウェア･プロセス：

回答者が所属する会社の特性とそこでのソフトウェア･プロセスに関する方針やガイドライン，および実際の活動について．

II. プロジェクトの概要とソフトウェア･プロセス：

回答者が現在携わっている（または最近まで携わっていた）プロジェクトの特性とそこでのソフトウェア･プロセスに関する方針やガイドライン，および実際の活動について．

III. ソフトウェア･プロセスの問題点と改善案：

ソフトウェア･プロセスまわりの諸問題に関する回答者の個人的見解，およびさまざまな改善案に対する意見について．

IV. 各個人の背景／教育／経験とソフトウェア･プロセスの実践：

回答者自身の特性および自分が実践しているソフトウェア･プロセスについて．

ソフトウェア技術者の生涯教育の実践やニーズもまた，今回の調査における１つの関心事である．この問題に関する質問はアンケート全体に散りばめられている．

I. 会社の概要とソフトウェア･プロセス

ここでは，まず，あなたの会社について，またソフトウェア･プロセスに関する会社としての方針やガイドライン，および実際の活動についてお訊ねします．これらの質問は，あなたの会社全体におけるプロセスの実態を明らかにすることを意図しています．

以下の質問は，あなたの所属する「部門」に関するものです．ここで，「部門」ということばは，あなたがいま働いている組織上の「大きな単位」(major organizational unit) を意味するものと解釈してください．

大企業の場合，そうした組織上の単位は，一般に「部」とか「事業部」とか呼ばれることが多いでしょう．中規模の会社では，「部」，「課」，「グループ」など，さまざまな名前で呼ばれます．小さな会社の場合は，そうした組織区分がないことが多いので，「部門」を「会社全体」と解釈していただいてかまいません．

Q-1. あなたの会社およびあなたの部門が提供する主要なプロダクトまたはサービスは？（それぞれの欄で，該当する答えを１つだけ選ぶ）

(1) 会社の主要なプロダクトまたはサービス ▽	(2) 部門の主要なプロダクトまたはサービス ▽	
15.2	6.6	ハードウェアおよびコンピュータ･システムの製造･販売
0.4	1.2	コンピュータの端末装置（ディスク･ドライブ，ターミナル，etc）の製造･販売
0.8	1.8	CAE/CAD/CAM システムの製造･販売
12.8	16.4	システム･ソフトウェアの受託･開発･販売
39.7	41.7	アプリケーション･ソフトウェアの受託･開発･販売
1.4	11.4	自社内向けのシステムおよびアプリケーション･ソフトウェア開発
0.2	0.6	マネジメント･インフォメーション･システムの開発
3.2	0.8	公益事業（電力，通信，etc.）
2.6	1.8	OA 機器の製造･販売
1.2	1.0	通信システムおよび装置の製造･販売
3.6	1.2	工業用制御システム，装置，ロボットの製造･販売
2.2	1.0	電子測器，自動検査装置，設計･検査装置の製造･販売
0.8	1.0	電子医療機器の製造･販売
0.8	1.0	航空，海洋，宇宙または軍事システムの開発
1.0	1.6	官公庁（国または地方自治体）
-	0.2	自動車その他の地上交通手段の製造･販売
7.6	3.2	大衆向け電機製品または消費財の製造･販売
0.2	-	集積回路またはセミコンダクタの製造･販売
0.2	-	電子部品（ボード，モジュール，電源，etc）の製造･販売
0.6	3.4	独立のまたはアカデミックな研究開発
1.2	1.6	技術，経営または特定分野のコンサルティング
0.2	0.2	医療，健康診断
0.2	-	法律，会計
0.6	1.4	教育，訓練

Q-2. あなたの部門には，ソフトウェア技術者がおおよそ何人いますか？（各行ごとに該当する答えを１つだけ選ぶ）

	10 人以下	11 ～ 50 人	51 ～ 100 人	101 ～ 500 人	500 人以上
部門全体	10.6	40.5	22.6	21.4	4.6
一番小さなプロジェクト	99.0	0.4	-	0.2	-
平均的なプロジェクト	75.2	23.8	0.6	-	0.2
一番大きなプロジェクト	35.7	48.9	10.0	4.0	0.8

Q-3. あなたの部門のソフトウェア･プロジェクトの構成はどうなっていますか？（該当する答すべてにチェック）

ライフサイクルのフェイズ別 (ex. 要求定義担当/全体設計担当/etc)	41.1
ソフト･プロダクトの機能別 (ex.UI 担当/データ管理担当/TR 処理担当，etc)	48.1
特定の出荷プロダクト別 (ex. 第 1 リリース担当/第 2 リリース担当/etc)	36.1

Q-4. あなたの部門では，ソフトウェア･プロダクトの保守や機能強化はどんな風にされていますか？（該当する答を 1 つだけ選ぶ）

[a] ふつうは開発を担当したのと同じスタッフが行なう	61.3
[b] ふつうは開発を担当したのと違うスタッフが行なう	3.8
[c] ときには同じスタッフが，またときには違うスタッフが行なう	32.9

[a] と答えた方は Q-6 へ．その他の方は次の Q-5 へ．

Q-5. あなたの部門における新規開発担当者と保守（機能強化）担当者の比率は？（該当する答を 1 つだけ選ぶ）

開発 5 以上　対　保守 1	30.6
開発 2 ～ 5　対　保守 1	41.5
開発 1　対　保守 1	12.0
開発 1　対　保守 2 ～ 5	12.6
開発 1　対　保守 5 以上	2.2

Q-6. あなたの部門のソフトウェア･プロジェクトの管理者のうちで，ソフトウェアに関連した経歴または経験を持っている人は，およそ何％ぐらいですか？（該当する答えを 1 つだけ選ぶ）

10% 以下	11 ～ 25%	26 ～ 50%	50 ～ 75%	75% 以上
8.4	5.6	6.8	10.8	65.7

Q-7. あなたの部門には，ソフトウェアに関連するさまざまな意志決定を改善するための社内講習会やセミナーがありますか？（該当する答すべてにチェック）

管理者向け	プロジェクト･マネージャ向け	技術リーダ向け	技術者向け
34.1	39.3	45.1	66.3

Q-8. あなたの部門が提供しているプロダクト（またはサービス）のユーザとあなたの部門とのインタフェイスまたはコミュニケイションは，うまく行っていると思いますか？（該当する答を 1 つだけ選ぶ）

大変よい	まあまあよい	あまりよくない	大変よくない	ユーザとの接触はない
4.6	64.3	18.4	2.0	7.8

Q-9. あなたの部門では，ソフトウェア開発の際に作られるさまざまな成果物（たとえば，仕様書，設計書，コード，プロジェクト計画，テスト仕様，etc.）を，プロジェクト間でどの程度再利用していますか？（該当する答を 1 つだけ選ぶ）

0 ～ 10%	11 ～ 20%	21 ～ 40%	41 ～ 60%	61 ～ 80%	81 ～ 100%
27.3	31.3	20.4	12.6	3.8	1.6

Q-10. あなたの部門では，標準的なソフトウェア･プロセスが明確かつ詳細に定義されていますか？

Yes	45.9	No	51.3

No と答えた方は Q-15 へ．Yes の方は次の Q-11 へ．

Q-11. そのプロセス定義は，それぞれのプロジェクトの事情に合わせてカスタマイズできるようになっていますか？

Yes	72.5	No	25.8

Q-12. そのプロセス定義を各プロジェクトに徹底するためのメカニズムとしてどんなものがありますか？（該当する答すべてにチェック）

ソフトウェア･プロセス定義ドキュメント	90.4
社内トレーニング･コース	36.2
定期的セミナー	15.7
第３者が行なうトレーニング･コース	8.7
ビデオ教材	7.0
ニュースレター	8.3

Q-13. あなたの部門のソフトウェア･プロセス定義では，プロセスが持つさまざまな側面のうち，どれとどれが取り上げられていますか？（該当する答すべてにチェック）

主要な開発フェイズの区分	86.0
あるフェイズで行うべきタスク	71.6
各フェイズまたはタスクの目的／目標	65.5
各フェイズまたはタスクの開始条件	36.2
各フェイズまたはタスクの終了条件	39.7
各フェイズまたはタスクが必要とする作業成果物	72.5
各フェイズまたはタスクが作り出すべき作業成果物	85.6
各フェイズまたはタスクにおいて人間が演ずべき役割	24.5
各フェイズまたはタスクのスケジューリング/コスト見積手順	35.8
各フェイズまたはタスクにおいて利用すべき開発技法	34.5
各フェイズまたはタスクにおいて利用すべきツール	29.7
各フェイズまたはタスクにおける報告義務	48.5

Q-14. それぞれのプロジェクトで実際に行われている開発プロセスが，部門であらかじめ定義された標準プロセスに合致しているかどうかを，どうやって検証していますか？（該当する答すべてにチェック）

プロジェクト内部の人間による定期的監査	65.9
プロジェクト外部の人間による定期的監査	31.0

Q-15. あなたの会社には，ソフトウェア･プロセス･グループ（すなわち現場の各部門で使われているソフトウェア･プロセスの定義･モニター･評価･改善に責任を持つ独立した組織単位）がありますか？

Yes	26.9	No	69.3

No と答えた方は Q-21 へ．Yes の方は次の Q-16 へ．

Q-16. そうしたソフトウェア･プロセス･グループは，会社内にいくつありますか？（該当する答を１つだけ選ぶ）

１つだけ	２～３グループ	４～５グループ	５グループ以上
67.9	14.2	3.0	11.9

Q-17. あなたの一番身近にあるソフトウェア･プロセス･グループの大きさは？（該当する答えを１つだけ選ぶ）

５人以下	６～15人	16～50人	50人以上
39.6	44.8	9.7	0.7

Q-18. そのグループの大部分のメンバーは専任ですか？

Yes	69.4	No	29.1

Q-19. そのグループは，およそ何年ぐらい前からありますか？（該当する答えを 1 つだけ選ぶ）

1 年未満	1 ～ 2 年	3 ～ 5 年	5 年以上
14.2	20.9	30.6	32.1

Q-20. そのグループは，プロセスの定義や改善勧告を各プロジェクトに伝えるのに，どんな手段を用いていますか？（該当する答すべてにチェック）

ガイドブックまたは解説マニュアルを発行する	75.4
各プロジェクトのプロセス定義を支援する	25.4
プロセスに関連した教育や訓練を行なう	41.8

Q-21. あなたの部門で現在動いているプロジェクトの中で，プロセスの評価（アセスメント）はどの程度実施されていますか？（該当する答を 1 つだけ選ぶ）

まったく実施されていない	33.5
プロジェクトのうち 1/4 以下でしか実施されていない	30.5
1/4 ないし 1/2 のプロジェクトで実施されている	10.8
半分以上のプロジェクトで実施されている	19.4

Q-22. あなたの部門のプロジェクトの中で，これから 12ヶ月以内にプロセスの評価（アセスメント）を実施することを予定しているものはどの位ありますか？（該当する答を 1 つだけ選ぶ）

まったく実施する予定はない	30.9
プロジェクトのうち 1/4 以下で実施する予定である	28.9
1/4 ないし 1/2 のプロジェクトで実施する予定である	11.4
半分以上のプロジェクトで実施する予定である	21.0

Q-23. あなたの部門では，ソフトウェア･プロセスの評価を支援するために，どんな技法が使われていますか？（該当する答すべてにチェック）

CMU/SEI のプロセス成熟度検査手法	0.6
その他の外部で開発されたソフトウェア･プロセス評価／調査手法	7.4
社内で開発されたソフトウェア･プロセス評価／調査手法	20.2
各種の基準･方針･手続きに関する文書のレビュー	38.9
顧客／ユーザへのインタビュー	22.2
類似プロジェクトとの比較分析	15.0
上級管理者，管理者，技術者，品質管理担当者へのインタビュー	14.8
品質データの収集と分析	30.7
生産性データの収集と分析	33.9
コスト･データの収集と分析	31.9
スケジュール･データの収集と分析	34.7
プロジェクト内のソフトウェア技術者の作業の観察	26.3
プロジェクト･ミーティングへの出席	35.5
作業成果物のレビュー／検査ミーティングへの出席	39.9
プロジェクト完了後のレビュー･ミーティングへの出席	25.5
支援ツールや環境のレビュー	13.2

Q-24. あなたの部門では？（該当する答すべてにチェック）

プロジェクト管理の標準/方針/ガイドラインを説明した文書がある	42.1
会社または業界の標準に準拠したソフトプロセスや開発技法がある	46.5
すべての作業成果物に対して品質保証が必要とされている	20.2
作業成果物の品質保証は開発チームと独立したグループで行われる	17.0
作業成果物をリリースしてよいかを判定する明確な基準がある	13.4
営業上のプレッシャーでプロジェクトのスケジュールが影響を受ける	58.1
明確に定義されたソフトウェア･プロセスが存在する	26.7
明確に定義されたプロセス･モニタリング計画が存在する	3.0
明確に定義されたソフトウェア･プロセス評価計画が存在する	7.0
明確に定義されたソフトウェア･プロセス改善計画が存在する	2.4
プロセスの定義/監視/評価/改善に責任を持つ中心的なグループがある	11.8
プロセスの評価は，プロジェクトの一部分として実施されている	35.9
プロジェクトが完了するごとに，必ず事後のレビューが行われている	23.6
そのレビューにおいては，プロセス上の問題点の分析が必ず行われる	13.2

Q-25. あなたの部門では？（該当する答すべてにチェック）

CASE 技術の導入に関するはっきりした計画がある	11.0
新技術の導入に責任を持つ特定のグループが設けられている	21.2
新技術の重要性や導入メリットの分析を担当する人間がいる	40.1
新技術の導入によるビジネスチャンスを経営者や管理者に理解させる	11.4
ツールや技法を選択する場合,必要な比較/評価情報のアクセス手段	11.8

Q-26. あなたの部門では？（該当する答すべてにチェック）

教育訓練に関して中心的な役割を果たすグループがある	46.7
技術者全員が毎年自分で継続的な教育計画を立てて，それを実行	23.0
すべての職務について，必要とされる知識や技能が，定められている	14.5
新人技術者のための訓練プログラムが用意されている	67.1
新人技術者は全員，入社時にその訓練プログラムを受けなければならない	66.9
新人技術者は全員，入社時に見習い社員研修を受けなければならない	48.9
技術者の昇格には一定の教育訓練が必要だという制度が確立されている	21.4
技術職から管理職への異動には一定の教育訓練が必要だという制度が確立	21.6
管理者の昇格には一定の教育訓練が必要だという制度が確立されている	16.0
管理者や技術者が大学や大学院で教育を受ける場合，費用を会社が負担	18.4
管理者や技術者が社内での教育や訓練のコースに参加することを奨励	58.9
管理者や技術者に，社外のセミナーや講習会に参加することを奨励	69.7
特定のプロジェクトの必要に応じて，特別な講習会やセミナーを実施	34.7

II. プロジェクトの特性とソフトウェア･プロセスの実際

このセクションでの質問は，あなたが現在働いている（または最近まで働いていた）ソフトウェア･プロジェクトに関するものです．そのプロジェクトでのプロセスに関するポリシー，ガイドライン，および実際の活動についてお訊ねします．これらの質問は，現実のソフトウェア･プロジェクトにおけるソフトウェア･プロセスの実態を明らかにすることを意図しています．

[1] もし，あなたが現在１つまたは複数のプロジェクトに関係している場合は，その中のどれか１つを選び，そのプロジェクトについて答えてください．

[2] あるいは，現在は特定のプロジェクトに関係していないが，過去５年間に１つまたは複数のプロジェクトに参加した経験をお持ちであれば，その中のどれか１つを選び，そのプロジェクトについて答えてください．

[3] 上のどちらでもないが，現在進行中のいくつかのプロジェクトの内情をよく知っているというのであれば，その中のどれか１つを選び，そのプロジェクトについて答えてください．

もし，上のいずれの立場でもないという場合は，Section-III の Q-55 へジャンプしてください．

Q-27. あなたは，このセクションの質問に対して，どのような立場で回答されますか？（該当する答を１つだけ選ぶ）

現在従事しているプロジェクトの１つについて答える	58.3
以前に従事し，１～３年前に終了したプロジェクトについて答える	18.8
以前に従事し，４～５年前に終了したプロジェクトについて答える	4.4
自分が従事してはいないが，よく知っている現在進行中のプロジェクト	11.3

Q-28. そのプロジェクトは？

いま働いている会社のプロジェクトである	89.8
以前働いていた会社のプロジェクトである	2.8

以下の質問では，あなたが現在そのプロジェクトに参加しているという状況を想定しています．もし，あなたが念頭においているプロジェクトが，過去のものであったり，実際に従事していないものである場合には，仮にいまそのプロジェクトに参加しているつもりになって答えてください．

Q-29. そのプロジェクトで働いているソフトウェア技術者の人数は？（該当する答を１つだけ選ぶ）

10 人以下	11 ～ 25 人	26 ～ 50 人	51 ～ 100 人	101 ～ 500 人	500 人以上
50.1	26.1	9.9	5.2	1.8	0.6

Q-30. そのプロジェクトの開始から終了までの期間は？（該当する答を１つだけ選ぶ）

0.5 年以下	0.5 ～ 1 年	1 ～ 2 年	2 ～ 5 年	5 年以上
0.6	20.4	32.5	29.7	4.2

Q-31. そのプロジェクトで作られるソフトウェアの用途は？（該当する答を１つだけ選ぶ）

自社内で使用するためのもの	12.6
特定の顧客向けのカスタマイズ･ソフトウェア	52.1
製品として販売するもの	17.4
非ソフトウェア製品の中に組み込まれるもの	9.6

Q-32. そのプロジェクトで作られるソフトウェアの種類は？（該当する答を 1 つだけ選ぶ）

非手続き的なもの（たとえばスプレッドシートまたは DB クエリーなど）	1.8
バッチ型の DB アプリケーション	4.2
その他のバッチ型アプリケーション	4.2
対話型の DB アプリケーションまたはトランザクション処理	32.7
その他の対話型アプリケーション	13.0
科学技術計算または数学的計算	1.2
OS またはその他のシステム･ソフトウェア	5.2
ソフトウェア･ツール（群）または環境	9.0
コミュニケーション／通信制御	5.2
プロセス制御	4.0
埋め込み型またはリアルタイム	7.0
グラフィクス，アニメーションまたはイメージ処理	2.6
ロボットまたはメカニカル･オートメーション	1.2
人工知能	0.6

Q-33. そのプロジェクトで作られるソフトウェアの性格は？（該当する答を 1 つだけ選ぶ）

研究開発を目的としたプロトタイプ	4.4
新しいソフトウェア･システムの実験的プロトタイプ	7.8
新しいソフトウェア･システムの最初のバージョン	45.1
既存のソフトウェア･システムに新しい機能を追加したバージョン	27.7
既存のソフトウェア･システムを機能面以外で改良したバージョン	7.6

Q-34. そのプロジェクトで作られるソフトウェアが稼働するマシンは？（該当する答を 1 つだけ選ぶ）

埋込マイクロプロセッサ	7.4
それ以外マイクロプロセッサ	3.6
大型汎用機	28.9
ミニコンピュータ	13.4
ワークステーション	26.9
パーソナル･コンピュータ	11.8

Q-35. そのプロジェクトで開発されるソフトウェアの目標サイズは，要求･設計･テスト仕様書などの**技術**ドキュメント（ただし，ソース･コードやプロジェクト計画書などは除く）のページ数でいえば，どの位のボリュームですか？（該当する答を 1 つだけ選ぶ）

100p 以下	101 ～ 400p	401 ～ 800p	801 ～ 1200p	1200p 以上
6.8	21.0	19.6	10.0	33.1

Q-36. そのプロジェクトで開発されるソフトウェアの目標サイズは，コメントを含まないソース･コードの行数（KSLOC すなわち 1 千行単位）でいえば，どの位のボリュームですか？（該当する答を 1 つだけ選ぶ）

30K 以下	31 ～ 100K	101 ～ 200K	201 ～ 300K	300K 以上
18.2	28.7	16.4	7.0	19.6

Q-37. あなたがそのプロジェクトで開発作業に利用しているハードウェア設備は？（該当する答すべてにチェック） Q-43

専用のコンピュータ	ターミナル	ワークステーション	パソコン
32.3	39.1	45.3	40.3

Q-38. あなたが開発作業に利用している主なハードウェア設備はどこにありますか？（該当する答を 1 つだけ選ぶ）

自分のデスクにあり，1 人で使っている	25.5
自分のデスクにあるが，複数人で一緒に使っている	7.6
自分のデスクからは離れた場所にある	58.5

Q-39. その開発用ハードウェアの利用形態は？（該当する答すべてにチェック）

スタンド･アローンで使っている	32.5
専用のメインフレームあるいはミニコンに直接つながっている	22.6
プロジェクト支援だけでなく，他の活動と共用の MF/ミニコンに直接接続	19.8
MF，ミニコン，あるいは FS へのアクセスを提供するような N/W に接続	35.3
MF，ミニコン，あるいは FS へのアクセスができないような N/W に接続	4.4

(MF ＝メインフレーム，FS ＝ファイルサーバー，N/W ＝ネットワーク)

Q-40. そのプロジェクトで使っているハードウェアから，外部のネットワーク（Telenet，CSNet，junet，またはあなたの会社の世界規模の広域ネットワーク）にアクセスできますか？

Yes	28.5	No	62.5

Q-41. そのプロジェクトで用いている開発技法の特徴は？（該当する答すべてにチェック）

アドホック	8.2
標準的	41.3
フォーマル	5.8
セミ･フォーマル	5.6
インフォーマル	8.2
プロトタイピングの概念にもとづく	15.8
構造化分析／設計の概念にもとづく	42.1

Q-44

Q-42. そのプロジェクトにおけるマネジメント技法の特徴は？（該当する答すべてにチェック）

アドホック	10.0
標準的	33.1
フォーマル	6.2
セミ･フォーマル	4.6
インフォーマル	7.0
ワークブレイクダウン	10.2
ウォータフォール	21.6
インクリメンタルまたはスパイラル	5.4
スケジュール重視	52.1
コスト重視	20.8
定量的	10.6

Q-4

Q-43. そのプロジェクトでは，次のそれぞれの活動に対して，体系的なきちんと文書化された手順，開発技法および（または）ツールを，どの程度利用していますか？

活動	存在しない	使っていない	部分的に利用	全面的に利用
プロジェクト管理	44.5	4.0	18.0	23.4
要求定義	43.1	5.4	16.8	25.1
全体設計	31.9	5.0	22.6	30.9
詳細設計	25.9	2.8	20.8	40.7
プロトタイピング	65.5	3.2	8.6	9.2
各工程間のトレース	57.9	5.0	10.6	12.6
コーディング	26.9	2.4	17.5	42.9
単体テスト	30.1	2.4	23.2	33.5
システムテスト	33.1	1.8	20.2	33.5
回帰テスト	65.7	1.8	8.2	8.6
レビュー／検査	34.5	3.8	21.6	29.1
品質保証	46.9	3.6	16.0	20.8
システム･インテグレーション	64.7	0.6	9.4	9.8
トラブル報告	27.3	3.8	16.2	40.9
メインテナンス活動	42.9	3.2	13.2	27.1
フィールド･サポート活動	59.7	2.6	10.6	11.8
コンフィギュレーション管理	58.3	2.4	8.6	14.2
変更管理	36.9	3.8	16.2	31.1
プロセス定義	53.7	2.4	11.6	17.2
プロセス･モニタリング	62.9	2.0	9.8	9.4

Q-44. そのプロジェクトでは，次にあげるコストまたはスケジュール見積技術のうち，どれを利用していますか？（該当する答すべてにチェック）

パブリックドメインの見積ツール（たとえば，COCOMO など）	2.2
市販の見積ツール（たとえば，SPQR20, CHECKPOINT, SLIM など）	0.4
プロジェクト･マネジャの経験にもとづくインフォーマルな分析	67.7
プロジェクトの規模にもとづく見積	50.1
過去のプロジェクトのコスト／スケジュール･データにもとづく見積	53.9
ファンクション･ポイント分析に基づく見積	6.4
顧客によるレビュー	19.0
複数の情報源を用いた見積（デルファイ法）	3.4

Q-45. そのプロジェクトでは，次にあげるリスクの分析／回避技術のうち，どれを利用していますか？（該当する答すべてにチェック）

プロトタイピング	22.8
顧客レビュー	49.1
要員の訓練	31.3
経費/利益分析	15.6
再利用	34.5
要求の凍結	38.7
チーム結成演習	1.6

Q-46. そのプロジェクトでは，メンバ間のコミュニケーションや協調を支援するために，どのようなメカニズムが使われていますか？（該当する答すべてにチェック）

電子メイル	16.8
電子掲示板	5.0
オンライン･ノート･システム	1.6
ニュースレター	11.6
定例ミーティング	77.2
電話会議	2.8
テレビ会議	1.6
飲み会やパーティ	43.1

Q-47. プロセスの品質を測り一定の水準を維持するための活動（たとえば，プロセス･レビューやプロセス･アセスメント）に対して，プロジェクトのリソースを何パーセントぐらい割り当てていますか？（該当する答を１つだけ選ぶ）

0%	1 ～ 5%	6 ～ 10%	11 ～ 25%	26 ～ 50%	51 ～ 75%	75% ～
12.4	26.5	26.3	13.4	5.8	1.6	1.2

Q-48. プロダクトの品質を測り一定の水準を維持するための活動（たとえば，テスティングやコード･レビュー）に対して，プロジェクトのリソースを何パーセントぐらい割り当てていますか？（該当する答を１つだけ選ぶ）

0%	1 ～ 5%	6 ～ 10%	11 ～ 25%	26 ～ 50%	51 ～ 75%	75% ～
6.0	13.2	24.8	25.1	13.6	2.0	1.8

Q-49. そのプロジェクトにおける正式なレビュー（すなわち，はっきりと定められた手順にしたがって行われるレビュー）の特徴は？（該当する答すべてにチェック）

いかなる成果物に対しても正式なレビューは行われない	13.0
すべての成果物に対して正式なレビューが行われる	11.6
すべてではないが，一部の成果物に対して，正式なレビューが行われる	58.5
どの成果物が正式なレビューを受けるべきかを決める一定の基準がある	15.0
成果物がいつ正式なレビューを受けるべきかを決める一定の基準がある	14.0
正式なレビューのガイドラインとして一連の質問表が用意されている	9.8

Q-50. そのプロジェクトでは，各種の開発成果物に対してどのようなレビューや検査を行っていますか？

(1) プロセス定義書	形態	行わない	39.3
		定められた手順にしたがって正式に	16.6
		アドホックすなわち非公式に	27.3
	回数	最初のバージョンができた時１度だけ	25.7
		新しいバージョンができる度に継続して	24.8
	範囲	成果物全体ではなく部分的に	29.5
		すべてのバージョンの全体を対象として完全に	20.6
	頻度	作成／修正の間しばしば（少なくとも週１回）	7.2
		作成／修正の間ときどき（週１回以下）	40.7

(2) プロジェクト計画書	形態	行わない	17.6
		定められた手順にしたがって正式に	35.1
		アドホックすなわち非公式に	32.7
	回数	最初のバージョンができた時1度だけ	36.1
		新しいバージョンができる度に継続して	34.5
	範囲	成果物全体ではなく部分的に	34.5
		すべてのバージョンの全体を対象として完全に	35.1
	頻度	作成／修正の間しばしば（少なくとも週1回）	8.4
		作成／修正の間ときどき（週1回以下）	55.9

(3) 要求仕様書	形態	行わない	12.0
		定められた手順にしたがって正式に	32.9
		アドホックすなわち非公式に	40.3
	回数	最初のバージョンができた時1度だけ	27.1
		新しいバージョンができる度に継続して	48.3
	範囲	成果物全体ではなく部分的に	38.5
		すべてのバージョンの全体を対象として完全に	36.1
	頻度	作成／修正の間しばしば（少なくとも週1回）	12.2
		作成／修正の間ときどき（週1回以下）	58.5

(4) 設計仕様書	形態	行わない	9.6
		定められた手順にしたがって正式に	38.1
		アドホックすなわち非公式に	38.5
	回数	最初のバージョンができた時1度だけ	25.5
		新しいバージョンができる度に継続して	52.7
	範囲	成果物全体ではなく部分的に	46.7
		すべてのバージョンの全体を対象として完全に	31.3
	頻度	作成／修正の間しばしば（少なくとも週1回）	16.0
		作成／修正の間ときどき（週1回以下）	58.3

(5) ソースコード	形態	行わない	27.7
		定められた手順にしたがって正式に	17.2
		アドホックすなわち非公式に	41.5
	回数	最初のバージョンができた時1度だけ	26.1
		新しいバージョンができる度に継続して	36.7
	範囲	成果物全体ではなく部分的に	46.9
		すべてのバージョンの全体を対象として完全に	15.8
	頻度	作成／修正の間しばしば（少なくとも週1回）	13.2
		作成／修正の間ときどき（週1回以下）	48.3

(6) テスト計画書	形態	行わない	19.0
		定められた手順にしたがって正式に	28.9
		アドホックすなわち非公式に	37.9
	回数	最初のバージョンができた時1度だけ	30.7
		新しいバージョンができる度に継続して	38.3
	範囲	成果物全体ではなく部分的に	42.9
		すべてのバージョンの全体を対象として完全に	26.5
	頻度	作成／修正の間しばしば（少なくとも週1回）	11.6
		作成／修正の間ときどき（週1回以下）	54.3

Q-51. プロジェクトが進んでいく過程で，下にリストアップした成果物の中でどれとどれが作られますか？（該当する答すべてにチェック）

営業用資料	29.5	バージョン説明書	16.2
プロジェクト管理計画書	41.7	ドキュメントレビュー報告書	22.2
リスク分析書	2.6	テスト計画書	60.1
コスト見積書	47.9	テスト･ケース	48.3
工数見積書	69.9	テスト･ドライバー	25.9
規模見積書	44.7	テスト報告書	63.7
スケジュール表	84.0	要求/設計/コードのレビュー報告書	31.9
プロジェクト状況報告書	56.3	システム統合化計画書	7.8
品質関連の計画書	16.2	導入報告書	13.4
品質関連の報告書	22.8	障害報告書	54.7
生産性関連の計画書	15.0	障害報告分析書	28.1
生産性関連の報告書	18.2	変更提案書	26.1
システム要求仕様書	60.7	変更報告書	35.3
ハード/ソフトのトレードオフ分析書	7.8	プロジェクト･ライブラリ	22.8
ソフトウェア要求仕様書	59.1	再利用のための部品	21.6
ソフトウェア設計書	73.5	手続きマニュアル	23.0
ソース･コード	76.6	訓練資料	13.2
ユーザ･マニュアル	62.1	プロセス定義書	16.0
技術報告書や論文	19.4	プロセス評価報告書	7.2

Q-52. そのプロジェクトでは，下記の作業を支援する自動化ツールを，どの程度使っていますか？

作業	利用できない	全然使わない	あまり使わない	よく使う
プロジェクト管理	49.1	20.0	6.8	6.2
要求定義	54.9	17.6	5.4	3.4
全体設計	54.7	18.8	5.2	3.0
詳細設計	46.5	18.8	7.4	9.2
プロトタイピング	50.5	15.0	7.6	7.0
各工程間のトレース	57.9	15.4	4.0	2.4
コーディング	39.7	16.0	10.4	17.8
単体テスト	40.7	15.6	10.8	15.8
システムテスト	49.9	15.0	9.6	7.0
回帰テスト	54.9	14.0	7.0	2.6
レビュー／検査	59.1	14.6	5.0	2.0
品質保証	53.7	15.6	6.6	4.6
システム･インテグレーション	58.3	13.6	4.4	2.6
トラブル報告	56.1	14.6	3.8	6.0
メインテナンス	51.9	14.4	7.0	6.0
フィールド･サポート	58.9	11.8	3.8	3.4
コンフィギュレーション管理	55.5	14.0	4.2	4.8
変更管理	47.7	16.6	8.2	7.4
プロセス定義	59.9	13.8	3.4	2.0
プロセス･モニタリング	59.1	14.6	3.2	1.2

Q-53. そのプロジェクトで使用している自動化支援ツールの入手先は？

	すでにあった	外部から購入	独自に開発	社内から入手
プロジェクト管理	5.4	3.8	4.0	7.2
要求定義	4.0	5.6	2.8	4.4
全体設計	4.2	4.6	3.0	4.4
詳細設計	7.0	5.0	3.6	7.8
プロトタイピング	6.2	5.6	3.8	3.0
各工程間のトレース	4.0	2.2	3.0	2.8
コーディング	13.8	5.6	5.2	7.4
単体テスト	13.0	4.6	7.0	6.0
システムテスト	8.4	2.4	6.2	5.0
回帰テスト	5.2	1.2	3.4	3.2
レビュー／検査	4.6	1.0	2.6	2.8
品質保証	5.2	0.8	3.0	5.8
システム･インテグレーション	4.6	0.8	2.6	3.2
トラブル報告	5.0	1.0	4.2	4.0
メインテナンス	5.8	1.8	4.4	3.8
フィールド･サポート	4.0	0.8	2.6	3.2
コンフィギュレーション管理	5.6	1.2	3.0	3.2
変更管理	8.6	2.6	5.0	4.2
プロセス定義	4.0	0.8	1.4	2.4
プロセス･モニタリング	3.2	0.8	1.4	2.6

Q-54. そのプロジェクトでは，次のことはどの程度行われていますか？

	まったくしない	ときには行なう	頻繁に行なう
仕様の再利用	18.6	45.9	23.8
設計の再利用	15.8	46.7	25.5
ソース･コードの再利用	8.4	36.9	42.5
テスト計画の再利用	36.5	32.3	17.2
ソフトウェア設計時にユーザの利用事例設定	46.9	27.3	10.8
プロトタイプの開発	40.3	32.7	13.4
テストケース設計時にユーザの利用事例設定	42.7	31.3	11.0
ブラックボックス（機能）テスト法の利用	41.7	22.8	19.8
ホワイトボックス（ロジック）テスト法の利用	40.9	27.5	15.8
フォーマル仕様の作成	33.5	29.1	21.2
インフォーマルな仕様の作成	26.9	38.7	17.8
既存ソフトウェアのリバースエンジニアリング	55.1	23.0	5.6
すべての開発成果物に対する品質保証	37.3	29.7	16.8
開発成果物に対するコンフィギュレーション管理	42.5	26.1	13.6
ソフトウェアの変更には，必ず正式な承認手続き	29.7	30.7	24.8
各種管理指標の収集と分析	43.1	28.1	12.8
各種生産性指標の収集と分析	40.7	29.9	14.0
各種品質指標の収集と分析	42.7	28.9	13.2
各種工程指標の収集と分析	42.7	27.3	14.4
ソフトウェア･エラーの類別	31.3	30.7	23.2
エラーの根本原因の追跡と分析	21.6	26.7	37.5
エラーの原因を除去するための開発工程等の改善	36.5	36.1	12.6
プログラム行数の継続的モニタリング	45.7	25.1	14.2
ドキュメント･ページ数の継続的モニタリング	59.7	18.4	6.0
ファンクションポイント情報の継続的モニタリング	64.1	13.6	4.2
プロジェクト遂行中エラーの継続的モニタリング	36.5	30.7	16.6
出荷／納品後発見のエラーの継続的モニタリング	30.5	29.3	23.6
プロジェクトで用いる開発工程の定義，改善	36.5	36.3	11.8
作業成果物の間のトレース可能性の維持	49.9	24.6	8.2
将来の変更やシステムの進化に対する明確な考慮	27.7	42.9	15.0

III. ソフトウェア･プロセスの問題点

ここでは，ソフトウェア･プロセスに関するさまざまな問題点の要因とその解決策の有効性について，あなた自身が個人的にどのように感じているかをお訊ねします．これらの質問は，ソフトウェア開発･保守の支援に関する現場のニーズを的確に把握し，将来の技術開発の方向を正しく導くことを意図したものです．

Q-55. ソフトウェア･プロジェクトにおいて発生する主要な問題は，開発活動のどの側面に起因していると思いますか？ あなた自身の経験にもとづいて，次の20項目の中から，もっとも問題を含んでいると思われるものを3つだけ選んでください．

プロジェクト管理	57.1	レビュー／検査	32.5
要求定義	63.1	品質保証	14.8
全体設計	28.9	システムインテグレーション	6.0
詳細設計	12.2	トラブル管理	7.6
プロトタイピング	4.8	メインテナンス	10.6
各工程間のトレース	5.0	フィールド･サポート	3.8
コーディング	4.6	コンフィギュレーション管理	2.2
単体テスト	5.2	変更管理	18.0
システムテスト	12.4	プロセス定義	3.6
回帰テスト	0.8	プロセス･モニタリング	1.8

Q-56. ソフトウェア･プロジェクトにおいて発生する問題のうちで，もっとも重大な問題は何ですか？ また，もっとも些細な問題は何ですか？ あなた自身の経験から，次の27項目中，それぞれ5つずつ選んでください．

(1) もっとも重大な問題（5つ選ぶ）▽	(2) もっとも些細な問題（5つ選ぶ）▽	
4.2	54.1	個人の作業スペースが十分でないこと．
45.5	5.6	プロジェクト開始時の計画が適切でないこと．
50.3	3.6	コスト／スケジュール／作業量の見積り現実的でないこと．
10.0	6.0	必要に応じてプロジェクト計画を修正することができないこと．
64.9	5.0	要求仕様があいまいであること．
36.5	11.8	要求仕様が変更されること．
48.3	11.8	能力のある技術者が不足していること．
45.3	8.0	能力のあるプロジェクト管理者が不足していること．
12.8	23.4	体系的な規律正しいアプローチを徹底できないこと．
17.6	14.2	要員に対する訓練が不十分なこと．
12.0	32.5	適切な支援ツールがないこと．
1.2	37.1	ツールのレスポンス･タイムが十分でないこと．
4.4	22.6	ターゲット･マシンを十分に利用できないこと．
5.4	33.1	支援ツールが統合化されていないこと．
7.2	29.5	研究開発の水準と現場での実践とのギャップが大きいこと．
10.4	10.2	ソフトウェア･プロセスがきちんと定義されていないこと．
3.0	13.6	ソフトウェア･プロセスのモニタリングが不十分であること．
2.8	11.4	ソフトウェア･プロセスの評価がなされていないこと．
6.8	21.6	端末，パソコン，ワークステーションが十分に利用できないこと．
23.2	5.8	技術レビューや検査が不十分であること．
25.7	5.6	テストや検証が不十分であること．
11.8	5.0	変更管理が適切に行われていないこと．
11.2	28.9	ドキュメント作成の支援が不十分であること．
1.4	37.9	ワークステーションがネットワーク化されていないこと．
15.4	5.4	品質保証や品質管理が適切に行われていないこと．
12.8	6.2	開発の段階でメインテナンスのことが考慮されていないこと．
3.2	10.4	開発の段階でリリース後の変更が考慮されていないこと．

Q-57. あなたの会社で次に挙げるような変更·改善がなされた場合，高品質のソフトウェアを納期通りに予算内で作成するということに関して，どれだけの効果があると思いますか？

変更·改善の効果は？	大きい	ふつう	小さい	ない
物理的な作業環境の改善	23.6	41.4	28.9	3.2
能力の高いスタッフのプロジェクトへの配員	81.4	12.8	3.4	-
ワークステーション·ベースの環境の利用	16.6	47.7	26.3	5.6
ワークステーションのネットワーク化	15.8	43.3	29.5	7.8
管理者教育の改善	30.1	40.9	21.2	5.0
技術リーダ教育の改善	38.9	41.9	14.8	1.6
新人教育の改善	21.2	41.7	29.5	4.2
一般の技術者教育の改善	25.7	47.3	20.6	2.8
技術的な手順，方法論あるいは技法の改善	30.9	45.3	19.0	1.4
管理上の手順，方法論あるいは技法の改善	27.7	43.3	22.8	3.0
コスト見積りの改善	25.9	38.5	26.7	5.0
工数見積りの改善	35.7	39.1	19.0	2.6
スケジューリングの改善	35.7	40.7	17.2	2.6
ソフトウェア·プロセス定義の改善	13.2	42.5	36.7	3.0
プロセスの進捗状況モニタリングの改善	11.4	44.1	36.3	3.8
ソフトウェア·プロセス評価の回数の増加	6.2	35.5	44.5	8.8
技術レビュー／検査の回数の増加	23.8	40.7	26.1	5.8
プロジェクト管理ツールの機能向上	12.2	40.7	36.9	6.0
要求定義ツールの機能向上	27.3	35.7	29.1	4.0
設計支援ツールの機能向上	30.5	40.7	22.6	2.6
コーディング支援ツールの機能向上	21.6	41.9	29.3	2.8
テスティング支援ツールの機能向上	34.5	41.7	18.8	1.4
SI支援ツールの機能向上	14.4	39.1	36.5	5.6
ドキュメント作成支援ツールの機能向上	28.3	40.7	25.5	2.0
ツール間の統合性の改善	18.4	41.7	30.3	5.4

Q-58. 次にあげる開発活動のうち，よりよい自動化支援ツールの導入によって大きな効果がもたらされるものはどれですか？（あなた自身の経験にもとづいて，該当するものすべてにチェックしてください）

プロジェクト管理	37.3	レビュー／検査	26.3
要求定義	34.5	品質保証	21.2
全体設計	30.1	システム·インテグレーション	6.2
詳細設計	38.3	トラブル管理	22.2
プロトタイピング	38.5	メインテナンス	27.5
各工程間のトレース	16.4	フィールド·サポート	6.0
コーディング	43.7	コンフィギュレーション管理	15.2
単体テスト	47.9	変更管理	37.9
システムテスト	41.7	プロセス定義	7.6
回帰テスト	12.4	プロセス·モニタリング	9.2

Q-59. ソフトウェア技術者に対する生涯教育のアプローチとして，あなたはどんなやり方がよいと思いますか？

	大変よい	ふつう	あまり	よくない
社内セミナー	26.5	57.9	11.6	2.0
社外セミナー	37.5	49.5	9.6	1.6
ビデオテープ	2.8	27.7	46.9	20.4
対話型のビデオディスク	9.8	42.9	34.5	9.6
自己学習	22.6	45.1	22.6	7.8
インフォーマルなOJT	29.7	46.9	14.6	6.0

Q-60. さまざまな開発スキルを改善するために訓練手段としては，あなたはどんなやり方がよいと思いますか？

	大変よい	ふつう	あまり	よくない
自己学習	22.6	44.7	24.2	6.2
プレゼンテーションと講義	24.8	55.1	16.2	1.0
実習	59.5	33.3	4.4	0.6
ケーススタディ	48.5	41.1	7.8	-
オンラインのヘルプ	8.4	41.5	29.5	3.0
オンラインのチュートリアル	5.6	41.7	29.9	3.6

Q-61. さまざまなツールや技法についての情報を得るためには，あなたはどんなやり方がよいと思いますか？

	大変よい	ふつう	あまり	よくない
試用してみる	70.1	12.0	1.8	0.6
ユーザ向けドキュメントを読む	5.2	49.9	25.9	3.2
技術ドキュメントを読む	8.6	46.1	26.1	3.4
ベンダの技術スタッフと話し合う	26.9	42.5	13.0	1.2
既存のユーザと話し合う	46.3	30.3	7.0	0.2
講習会を受講する	12.8	51.1	19.2	1.4

Q-62. 以下にリストアップしたツールや技法あるいは概念の，現在の時点における知名度／普及度，および，将来それらが開発の生産性や品質に対する効果について，あなたはどのようにお考えですか？

	知名度／普及度				将来の効果			
ツール／技法／概念	高い	普通	低い	ゼロ	高い	普通	低い	なし
高水準の手続き言語	44.1	23.6	11.8	0.4	28.5	38.1	10.6	0.6
アプリケーション向き言語	21.6	35.5	22.4	0.4	19.0	42.7	16.0	0.2
アプリケーション･ジェネレータ	8.2	30.3	37.5	3.4	27.5	32.3	16.4	1.6
オブジェクト指向プログラミング	29.9	23.6	24.6	2.2	47.5	23.4	7.2	0.2
構造化分析／設計	47.9	22.0	10.0	0.2	29.7	40.1	8.6	-
視覚的プログラミング	8.6	20.2	37.9	11.4	32.3	26.5	16.0	1.8
実行可能な仕様記述	5.0	17.6	42.5	12.8	22.2	30.9	21.4	2.0
フォーマル仕様記述	5.8	24.2	38.1	9.6	8.2	33.9	30.9	2.0
エキスパート･システム	28.7	27.9	20.2	3.2	18.4	37.9	20.2	1.6
ハイパーテキスト／ハイパーメディア	22.4	25.7	26.7	4.0	28.7	31.5	15.0	1.4
自動プログラム合成	8.2	23.6	39.5	7.6	24.8	27.1	23.2	1.6
自動検証	4.0	16.8	45.7	12.2	26.5	28.7	19.2	2.2
ラピッド･プロトタイピング	5.6	23.4	36.5	11.4	21.0	36.3	15.6	2.6
プロセス･モデリング	1.8	20.0	42.7	12.8	10.8	37.9	23.4	2.8
プロセス評価	2.0	20.8	42.5	11.8	10.2	40.9	22.0	2.0
プロジェクト管理上ツール･セット	4.0	25.3	42.3	6.6	18.0	37.5	19.0	1.6
再利用ライブラリ	13.0	35.7	27.9	3.0	33.1	34.3	10.4	0.2
再利用ライブラリのブラウジングツール	5.6	17.4	44.9	9.6	25.9	33.1	14.2	1.6
リバース･エンジニアリング	9.2	23.2	36.7	9.4	17.2	31.9	23.0	3.6
テスト生成ツール	8.2	27.3	39.9	4.0	25.3	37.5	14.0	1.0
ルールベースのプログラミング	5.6	17.4	44.3	10.6	6.8	34.5	31.1	2.4
CASE ツール	36.7	27.3	14.8	1.2	36.9	33.9	7.4	0.6
ワークステーション･ベースの環境	31.7	32.5	14.6	0.8	37.9	34.9	4.8	0.6
グラフィックな設計記述法	9.2	27.1	36.9	5.8	28.7	33.9	13.4	0.8
設計分析ツール	6.0	28.7	39.9	4.2	22.2	37.1	16.6	1.0
ドキュメンテーション･テンプレート	7.2	28.5	37.7	4.6	15.4	36.3	22.8	1.8
開発成果物のレビュー／検査	19.6	32.7	23.2	3.2	23.6	40.9	11.8	0.4

IV. 個人的な背景，教育，経験

このセクションでは，あなた自身の個人的な背景，教育，経験について，また，ソフトウェアの開発･保守の分野であなたがどのような実践をして来られたかについて，お訊ねします．これらの質問は，前のセクションにおけるあなたの回答を正確に理解するためのものです．

Q-63. あなたの最終学歴は？（該当する答を 1 つだけ選ぶ）

高卒	学士	修士	博士
14.6	58.9	9.4	1.0

Q-64. あなたの勉強した分野は？（該当する答を 1 つだけ選ぶ）

計算機工学	4.0	物理学	7.4
システム工学	3.2	数学	4.8
ソフトウェア工学	6.8	応用数学	1.6
電気工学	16.2	コンピュータサイエンス	2.6
宇宙工学	0.4	その他の理論科学	2.4
原子核工学	0.4	データ処理	1.4
人間工学	0.2	オペレーションズ･リサーチ	1.0
機械工学	2.4	経営学または管理工学	5.8
その他の工学	8.0	心理学	0.4
応用物理	3.4	その他の文科系学科	10.0

Q-65. あなたは，会社に入る前または入社後の生涯教育の過程で，次の技術分野のうちどれを学びましたか？（該当する答すべてにチェック）

離散数学	7.0	ソフトウェア工学の有益性	18.0	プログラミング方法論等	38.1
フォーマル仕様技術	10.2	検査／検証技術	14.6	プロトタイピング	16.2
構造化分析／設計	48.9	テスティング	20.2	ソフトウェアの再利用	22.8
ソフトウェアメトリクス	9.8	品質保証	17.8	グラフィクス	12.8
プロジェクト管理	41.9	ソフトウェア工学方法論	24.8	人工知能	13.4
コスト見積り	25.9	要求定義方法論	17.6	分散コンピューティング	13.0
コンフィギュレーション管理	6.4	ソフトウェア設計方法論	40.7	データベース技術	30.9

Q-66. あなたは，学会その他の専門家団体（たとえば，SEA，情報処理学会，ソフトウェア科学会，ACM，IEEE など）に所属していますか？

Yes	38.9	No	44.5

Q-67. あなたは，仕事上の専門的なワーキング･グループ活動に参加していますか？

	Yes	No
社内のワーキング･グループに参加	34.5	49.3
社外のワーキング･グループに参加	16.8	67.1

Q-68. あなたは，仕事に関連のある雑誌，ニューズレターその他の定期刊行物をよく読みますか？（該当する答を１つだけ選ぶ）

仕事の一環として読んでいる	61.3
特別な課題，問題がある時に読む	20.2
ほとんど読まない	3.2

Q-69. あなたは，専門的な会議やワークショップに１年間に何回ぐらい参加しますか？（該当する答を１つだけ選ぶ）

１回以下	１回	２～３回	４回以上
32.3	12.8	26.9	13.0

Q-70. あなたは，ソフトウェアの開発や保守の仕事に何年くらい携わっていますか？（該当する答を１つだけ選ぶ）

～２年	３～４年	５～10年	10年～
2.8	7.4	31.7	43.1

Q-71. あなたは，現在ソフトウェア開発や保守の現場の第一線で仕事をしていますか？ それとも現場の第一線を離れてから何年かたちますか？

まだ現場にいる	離れて２年以内	離れて３～４年	離れて５～10年	離れて10年以上
57.7	8.4	8.6	7.2	2.4

Q-72. あなたは，ソフトウェア開発や保守の仕事で，どのようなな職務を経験しましたか？

職　務	経験なし	いくらか	十分ある
全社レベルの政策や戦略の立案	57.5	20.2	5.8
契約のモニタリング	64.5	14.0	5.0
全社レベルの経営情報システム等	60.9	16.8	5.8
品質や生産性の分析と改善	36.3	37.9	9.8
プロジェクト管理	16.8	37.7	29.7
技術グループの管理	27.1	34.9	22.0
システム運用管理	33.3	34.5	15.8
コンピュータオペレーション	27.9	32.1	23.4
プロセスの定義/評価/改良	31.3	38.1	13.8
システム要求定義	13.4	37.9	32.3
ソフトウェア要求定義	7.4	36.1	40.1
ソフトウェア設計	2.6	21.2	60.3
コーディング	2.6	18.6	63.3
ドキュメント作成	1.8	19.6	62.5
テスティング	4.6	21.8	57.7
品質保証	24.8	38.3	20.6
システムインテグレーション	37.3	33.3	12.8
コンフィギュレーション管理	39.9	31.7	10.2
ソフトウェアのインストール	14.2	39.9	29.7
ソフトウェア保守	10.2	33.9	40.1
訓練計画の策定／実施	28.9	42.1	12.4
ツールまたは環境の開発/保守	27.9	37.3	18.0
営業／販売	63.5	16.2	3.6
調査／研究開発	33.9	38.1	11.8
プロセスの定義	45.5	31.3	5.8
プロセスの評価	47.5	29.5	5.4
プロセスの改良	48.1	28.9	5.4

Q-73. あなたの現在の職務は？（該当する答を１つだけ選ぶ）

経営管理	9.0
プロジェクト管理	18.0
技術グループの管理	11.4
プロセスの記述	-
プロセスの評価	-
プロセスの改良	-
プロセスの定義／モニタリング／評価／改良	0.8
自社のソフトウェアプロダクトの開発／保守	7.2
自社製品に組み込られるソフトの開発／保守	8.6
特定の顧客向けのソフトウェアの開発／保守	12.4
ソフトウェア･ツールや環境の開発／保守	5.8
システム運用管理	1.2
コンピュータ･オペレーション	-
ドキュメント作成	0.2
テスティング	-
品質保証	0.2
ソフトウェアのインストール	0.2
訓練計画の策定／実施	1.0
フィールド･サポート	0.6
営業／販売	0.4
調査／研究開発	6.4

Q-74. 上の質問でお答えになった職務についてからの年数は？（該当する答を１つだけ選ぶ）

～１年	１～２年	３～５年	５年～
8.2	17.8	30.7	27.3

以上

我社におけるプロセス成熟度の調査と考察について
(独立系を中心とした日本企業プロセス成熟度調査とその後)

佐藤 雄一 (y-satoh@sra.co.jp)
株式会社 SRA 技術推進部

1991 年 12 月 21 日に南山大学において「情報管理学科ソフトウェア工学ワークショップ」が SEA の協賛を得て行なわれました。私は『日本(独立系)企業におけるプロセス成熟度の調査と比較』と題して発表をさせて頂きましたが、この時発表内容に対し聴衆の方々から質問やコメント等かなり反応がありました。会場には現場の方が沢山いらっしゃり、現場の成熟度モデルに関する関心の高さを思わせました。そこで多少手を加えて SEAMAIL で発表させて頂こうと思い立ちました。よろしくお願いします。(前半部分既に聞いた事のある人は後ろの方だけ読んで下さい)。

1. はじめに

ソフトウェア開発プロセスの信頼性は、ソフトウェア以外の生産業種と比べかなり低いという指摘が文献 [1] など各所でなされている。昨今プロセス成熟度モデルが話題となり、開発プロセスの向上 (とくに信頼性の向上) に寄与すると期待されている。当社では 1990 年秋に、日本(独立系)企業におけるプロセス成熟度の実態調査を、成熟度アンケートを用いて行なった。その後も折りに触れて、社内での討論を行ないその結果を整理してきた。今回の発表では、前半にアンケート調査の結果を述べ、後半には社内での討論や成熟度に対する見方・状況等を報告する。以下では、2 章にプロセス成熟度モデルの概要を説明し、3 章で 1990 年秋のアンケート調査の結果をまとめる。4 章では成熟度レベル向上の指針について考察する。5 章では、アンケートで測り切れていないプロセス側面についての、社内での調査や討論結果を参考までにまとめている。6 章では成熟度モデル自体の位置付けや状況についての試論をまとめた。

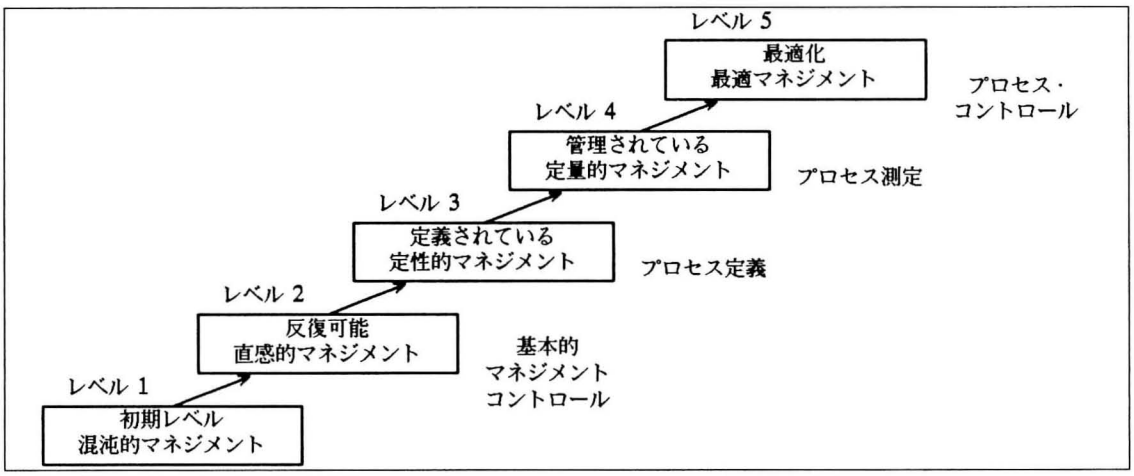


図 2-1 プロセス成熟度モデルの 5 つの段階

2. プロセス成熟度モデルについて

SEI (Software Engineering Institute) のプロセス管理に関する主張には、以下のようなポイントがある ([2][3][4][10] 他)。

○ プロセスは定義・管理・測定、そして段階的な改良が出来るものである。開発組織は**各工程を安定して繰り返せるようになって**初めてプロセスの改善に本腰を入れて取り組む事が出来る。

○ ツールや技術レベルの向上も大切だが**プロセスの向上が伴わないと個々の改善は大きな効果をもたらさない。**

○ 開発組織は一つ一つ抱える事情が違う。**各々の組織に適合した改善のポイント**を指し示す事が、プロセス向上戦略を立てる上

で大変に重要である。

○ (個人プロセスだけでなく) チームとしてのプロセス管理は、**プロジェクトの規模が大きくなると飛躍的に重要性を増す。**

○ 組織全体がプロセス向上の恩恵を受けたければ**組織ぐるみでのプロセス改善**の必要性がある。

ソフトウェア開発組織がどのようにして信頼性と安定性のあるプロセスを築き上げるか、という過程をモデル化したものが SEI の 5 段階プロセス成熟度モデルであり、その段階を測るメジャーに相当するものが成熟度アンケートである (文献 [4] 等参照)。このモデルの目的は、ソフトウェア開発組織の状況をプロセスの観点から分析し、その組織に合ったプロセス改善の為の目標を設定する事である。

図 2-1 に成熟度の 5 段階の特徴を示す。

(1) レベル 1

"初期" レベル、または "混沌的マネジメント" のレベル。プロジェクトの動向が正しく把握されていないので、安定したプロセス (工程や段取りなど) の改善が不可能な状態にある。プロジェクトの成果は一定せず、行きあたりばったりである。このレベルではまずスケジュールやコストの正確な予測を行なう努力が必要である。またコンフィグレーション管理をしっかり行なう必要がある。

(2) レベル 2

"反復可能" レベル、または "直感的マネジメント" のレベル。コスト、スケジュール、仕様変更、契約などに関して経験的直感に基き管理を行なうことができる。行きあたりばったりではなく、毎回同じような開発プロセスを辿ることができる。しかし管理は一部の人の経験に頼っており、その人と共にノウハウが消え去る恐れがある。また新たな課題に直面した時レベル 1 に戻る可能性が大きい。

(3) レベル 3

"定義可能" レベル、または "定性的マネジメント" のレベル。プロジェクトに関しより多くの経験的知識が組織で共有され、プロセスを比較し改善することができる。危機に際しても冷静さを失わず過去の経験から最善の策を取ることができる。CASE その他の先端的技術も容易に取り入れてその恩恵にあずかることができる。但し使用される計測データはまだ定性的なものが多い。

(4) レベル 4

"管理可能" レベル、または "定量的マネジメント" のレベル。"レベル 3" で得た知識と統制に加えレベル 4 の組織では、数値的 (定量的) にプロセス (工程・段取りなど) の比較・改善を行なうことができる。このレベルにおいて、もっとも大きな進歩が可能である。しかし統制されたプロセスを維持するためには、マネジメントの強い支援と人々の訓練・教育など不断の努力を要する。

(5) レベル 5

"最適化" レベル、または "最適マネジメント" のレベル。測定されたデータを使って、組織は各々のプロジェクト毎に最適なプロセスを設計できる。また、その最適化 (補正) を行なうことができる。危機があればそれを予測し、事前に取り除く。工程に弱点があれば、それを早く見つけ事前に補強する。"レベル 5" ではあらかじめ危機が起こらないようにプロセスをアレンジする、

3. 日本 (独立系) 企業におけるプロセス成熟度の調査と比較

SRA ソフトウェア工学研究所では 1990 年 10 月に日本のソフトウェア開発業界 (主に独立系及びユーザ系) における「ソフトウェアプロセス成熟度」のアンケート調査を行なった [7]。これは成熟度研究を始めるに当たりまず、成熟度の観点から我々日本のソフトウェア業界がどう見えるか、位置付けられるかを知りたかった為である。結果として、"レベル" という見方で言うと調査対象 10 社が全て "レベル 1" に相当し、差異は見られなかった。しかし結果の分析過程で、企業や業務別などの特徴が統計に良く現われていた。この調査 (統計) に現われた業務別の特徴などを中心に報告する。

3.1 調査内容

3.1.1 成熟度アンケートの構成

本調査では SEI の成熟度アンケートを翻訳して、調査に使用した。アンケートは、デモグラフィーと呼ばれるバックグラウンド情報（例えば所属する会社の規模や回答者の役職など）質問欄と、本体であるプロセス成熟度質問欄の 2 つから構成されている。プロセス成熟度質問欄は全 85 問あり、全て Yes-No 選択方式である。表 3-1 に質問の例を挙げる。これらの 12 問は全 85 問中でも特に「レベル 2 キークエスチョン」と呼ばれるもので、成熟度がレベル 2 以上であるかどうかの判断に用いられるものである。

表 3-1 レベル 2 キークエスチョン

質問番号	質 問 文
1.1.3	ソフトウエア品質保証の担当者は，その開発プロジェクトの管理者とは別系統の，管理者レベルへの報告ルートを持っていますか
1.1.6	それぞれのプロジェクトに（それがソフトウェア開発を含む場合），ソフトウエア構成管理の担当者が配員されていますか
2.1.3	契約に先立って，それぞれのソフトウェア開発計画に対するマネジメント・レビューを行なうための正式な手続きが，きちんと使われていますか
2.1.14	ソフトウエアの規模（サイズ）を見積もるために，きちんとした手続きが使われていますか
2.1.15	開発スケジュールを立案するために，きちんとした手続きが使われていますか
2.1.16	開発コストを見積もるために，きちんとした手続きが使われていますか
2.2.2	ソフトウエア構成管理の対象となる各要素に対して，ソフトウエア・サイズのデータが時系列的に維持されていますか
2.2.4	コーディングおよびテストのエラーに関する統計データが収集されていますか
2.4.1	トップ・マネジメントがソフトウエア開発プロジェクトの進捗状況を定期的に監視するためのメカニズムが存在しますか
2.4.7	ソフトウェア開発の第一線マネージャは，開発のスケジューリングやコスト見積りを決定する権限を持っていますか
2.4.9	要求仕様に対する変更を管理するために，何らかのメカニズムが働いていますか
2.4.17	コードに対する変更（だれが，どのような状況下でなら，変更を行なってよいか？）を管理するために，何らかのメカニズムが働いていますか

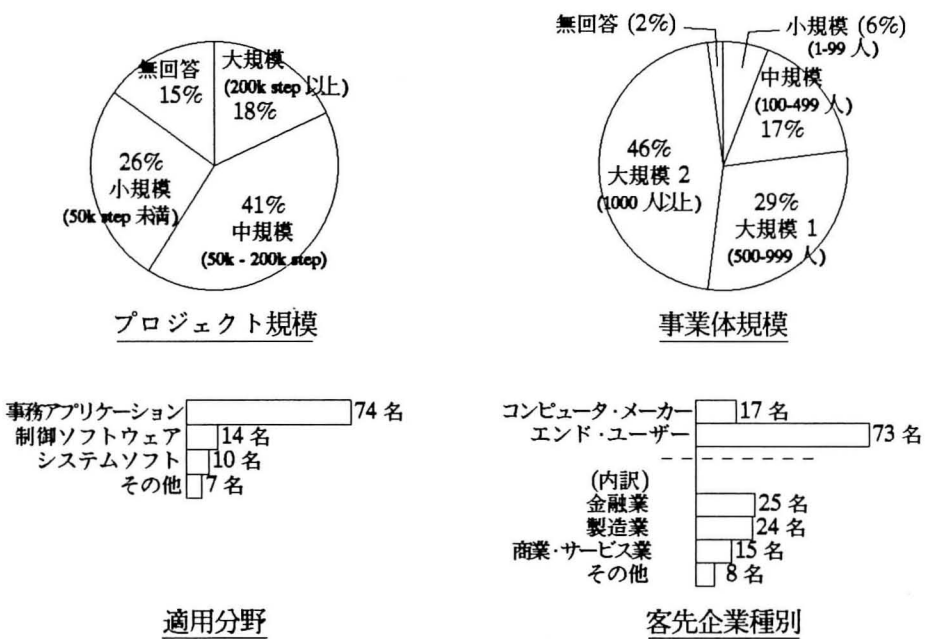


図 3-1 アンケート回答者のプロフィール

3.1.2 アンケートの実施経過

アンケートは 1990 年 10 月初頭から配布を開始し、同月中旬に回収した。アンケートにはソフトウェア開発企業 10 社が協力し、106 人の回答が回収された。10 社は殆どが独立系であったが、その中には、特定の企業の業務を中心とする実質 "ユーザ系" も数社含まれる。

3.1.3 アンケート回答者のプロフィール

デモグラフィー回答結果をもとに回答者の、職種その他のプロフィールの一部を図 3-1 に紹介する。

3.2 結果及び比較

3.2.1 SEI のサーベイとの比較

弊社のアンケートとほぼ同時期に、SEI も日本でプロセス成熟度の調査を行なっている [5]。SEI の調査対象は殆どがコンピュータ製造業とその系列企業群と推察され、SEI と弊社の調査対象は補間的関係にあると考えられる。SEI の調査によると一部のソフトウェア・ファクトリーを除き、調査した組織の殆んどが "レベル 1" である。弊社の調査では対象企業 10 社は全てレベル 1 に相当した (*)。

質問 No	質問概略内容	比較内容
1.1.3	SQA 及び独立した報告ルート	
1.1.6	構成管理担当者の存在	
2.1.3	契約前の開発レビュー手続き	
2.1.14	サイズ見積り手続き	
2.1.15	スケジュール立案手続き	
2.1.16	コスト見積り手続き	
2.2.2	サイズデータの時系列記録	
2.2.4	テストエラー統計データ収集	
2.4.1	トップの定期的監視メカニズム	
2.4.7	第一線マネージャの見積り権限	
2.4.9	要求変更管理メカニズム	
2.4.17	コード変更条件管理	

上: 日本, 下: 米国　　標本数: 106 ／ 96

図 3-2 全体結果及び米国でのデータとの比較

3.2.2 日米の傾向比較

以下では 85 問の成熟度質問回答結果のうち、「レベル 2 へのキークエスチョン」と呼ばれる 12 問への回答結果のみ扱う。これらは調査対象組織がレベル 2 以上であるか判断する為のもので、レベル 1 企業にとって最も重要と言える。

回答者全員のレベル 2 へのキークエスチョンへの、Yes 回答の割合 (以後 Yes の割合と書く) を図 3-2 に示す。この表では米国での調査結果 ([4] Workshop Assessment Data) も示している。米国でのデータは DoD 関連企業 (優良企業) が主である。まず質問 1.1.3 及び 1.1.6 を見る。米国での Yes の割合が高い一つの理由は、質問内容が DoD の下請け条件と重なる為と考えられる。一方日本では SQA (ソフトウェア品質保証グループ (**)) の設立に消極的である。

次いで Yes の割合が低いのは質問 2.2.2 である (19%)。コードサイズのトレースは、プロセスを乱す要因の早期発見の為に重要である。質問 2.2.4 とともに、システマティックなデータ採集がプロジェクト管理にあまり利用されていない事を示唆している。

質問 2.1.14 から 2.1.16 はマネージャーの見積り手続きについてであるが、いわゆる "勘" に頼るあまり、手続き化して主観による偏りを排除する努力が、殆どなされていない事を示唆している (***)。

(*): これはアンケート結果から機械的に引き出した結論で、正確にはプロジェクト毎と企業組織の "レベル" は区別しなければならない。各企業単位の集計結果はどれもレベル 2 から程遠く、実質的にどの回答者の属するプロジェクトもレベル 1 に属すると考えられる。
(**): SEI の言う SQA のまず最初の目的は、各プロジェクトが定められたプロセスに則って仕事を進めているかチェックする事である。
(***): ここで言う見積り手続きは、数値化した方法だけでなく Wideband Delphi Technique ([3] pp.92) など、マネージャーの直感をより "客観的に" 利用しようとする手法を含んでいる。

質問 2.4.7 では日米で Yes の割合が逆転している。これは日本における「現場」の権限の高さを指し示しているものと言える。

3.2.3 業務形態による特徴

回答結果をデモグラフィー質問の結果に基づいて色々な業務形態に分けて比較した。

質問No	質問概略内容	比　較　内　容
1.1.3	SQA 及び独立した報告ルート	
1.1.6	構成管理担当者の存在	
2.1.3	契約前の開発レビュー手続き	
2.1.14	サイズ見積り手続き	
2.1.15	スケジュール立案手続き	
2.1.16	コスト見積り手続き	
2.2.2	サイズデータの時系列記録	
2.2.4	テストエラー統計データ収集	
2.4.1	トップの定期的監視メカニズム	
2.4.7	第一線マネージャの見積り権限	
2.4.9	要求変更管理メカニズム	
2.4.17	コード変更条件管理	

上: 小規模, 中: 中規模, 下: 大規模　　標本数: 28 ／ 43 ／ 19

図 3-3 プロジェクト規模による比較

(1) プロジェクト規模による比較

回答結果をプロジェクトの規模 (予想ソースコードステップ数) によって分けたものを図 3-3 に示す。プロジェクトのサイズが大きいほど、プロセスに注意を払っている事が、回答結果に顕著に現われている。これは大規模なプロジェクトほど経済的影響が大きい為であり、またプロセス管理に人的·時間的資源を回す余裕がある為である。とりわけ以下の主題において Yes の割合が高い (括弧内は質問番号)。

- ·開発計画のレビュー (2.1.3)
- ·サイズの見積り (2.1.14)
- ·スケジュール立案手続き (2.1.15)
- ·コスト見積り手続き (2.1.16)
- ·開発中のソフトウェア·サイズのトレース (2.2.2)
- ·要求仕様の変更 (2.4.9)

これらの項目は経費に注意を払いながらプロジェクトを安定運営する事と深く関わる項目ばかりである。

第一線マネージャーのコスト·スケジュール見積り権限 (2.4.7) の Yes の割合も大規模プロジェクトの方が高い。正確なスケジュールやサイズ見積りの為には、同種のプロジェクトの実経験が豊富な現場サイドの関与が不可欠である。

質問No	質問概略内容	比　較　内　容
1.1.3	SQA 及び独立した報告ルート	
1.1.6	構成管理担当者の存在	
2.1.3	契約前の開発レビュー手続き	
2.1.14	サイズ見積り手続き	
2.1.15	スケジュール立案手続き	
2.1.16	コスト見積り手続き	
2.2.2	サイズデータの時系列記録	
2.2.4	テストエラー統計データ収集	
2.4.1	トップの定期的監視メカニズム	
2.4.7	第一線マネージャの見積り権限	
2.4.9	要求変更管理メカニズム	
2.4.17	コード変更条件管理	

上: 事務アプリ開発, 下: 制御アプリ開発　　標本数: 74 ／ 14

図 3-4 開発対象分野による比較

(2) 開発対象分野別による比較

図 3-4 に開発対象分野が事務アプリケーションである場合と、制御ソフトウェアの場合に分けた結果を示す。事務アプリケーション群で Yes の割合の高い質問は、プロダクトの品質に直接関連した質問ではなく "納期通りに終らせる" ためのファクターである。一般に事務アプリケーションを対象とするプロジェクトでは、客先から、品質についてよりもコスト的なパフォーマンスを要求される。また、事務アプリケーションを対象とするプロジェクトは一般に制御分野のプロジェクトより規模が大きく、プロジェクト規模による比較との相関がある。

一方制御アプリケーション群で Yes の割合が高い項目は、プロダクトの品質及びコンフィグレーションに関連するものである。一般に制御アプリケーションには信頼性が要求され、信頼を維持するためにはバージョン管理などのコンフィグレーション管理が大切である。質問 2.4.7 もこの群で高いが、これは制御アプリケーションの方が一般に複雑で変化に富み、開発のスケジュールやコストの決定が熟練者（現場の技術者寄りのマネージャー）に任されているためと推測される。

一方、質問 2.4.9 は事務アプリケーション群で Yes の割合が高い。規模の大きい事務アプリケーション程、経済的理由から変更が良く管理されていると言える。

(3) 元請け・下請けの区別による比較

回答結果を、プロジェクトが元請けであるか下請けであるかによって分けると、わずかではあるが総じて下請けの Yes 回答の割合が高い事がわかった。下請け企業の場合、契約先もソフト開発企業であるから、いわばプロに監視される立場にある。この為に下請け側での Yes の回答の割合が高くなったと思われる。元請け企業の場合、契約先はエンドユーザーでソフトウェアに詳しくない場合があり、関心も異なる。一方 "プロ" の場合はどうチェックすれば良いか知っている。

また質問 2.1.5 (下請け業者のプロセス監視制度の有無を問うもの) への元請け群の Yes の割合が 54% と高く、下請け業者への監視を裏付けている。

(4) 役職による比較

役職による比較はこれまで述べてきた比較と異なり、同一プロジェクトの中で役職による回答の差（＝現状への認識の差）がある、ということを示唆している。回答結果を、マネージャー職／プロジェクト・リーダー職／それ以外のエンジニア職、の 3 群に分けてみた。すると、レベル 2 へのキークエスチョンではリーダー職群は 6 つの質問に関して Yes の割合が他の 2 群より低かった (質問 1.1.3, 1.1.6, 2.1.14, 2.1.15, 2.1.16, 2.4.9)。レベル 3, 4, 5 へのキークエスチョンにおいても、そのような質問項目がそれぞれ 7 質問 (全 13 問)、6 質問 (全 12 問)、4 質問 (全 4 問) あった。リーダー職の回答は他の 2 群より "No" の回答が多く、傾向としてプロセスに対し "厳しい" 見方をしている。

プロジェクト・リーダーはプロジェクトの直接的な管理者であり、また多くの組織において現場の下級技術者とマネージャーの殆ど唯一のパイプ役である。職務柄、もっとも詳細な作業から開発全体のプロセスまで、作業状況をもっとも広く知っていると考えられる。リーダー群はもっとも正確にプロジェクトの状況を答えているようである。

(5) その他の傾向

成熟度質問回答全問にわたって、総当たり戦で相関を調べた。その結果、質問 1.2.1 への回答 (ワークステーション、端末の普及率を示す) と他の多くの質問回答の間に負の相関がある事が分かった。しかし同時に、質問 1.2.1 回答とプロジェクト規模の間にも負の相関傾向があった。これは小規模なプロジェクトほど機械化されていて、かつ成熟度質問の結果を見る限りではプロセスに注意が払われていない、という事を示唆している。

その他には、あまり顕著な相関を示すものがなく、あっても類似した質問 (相関があって当然のもの) であった。

3.3 調査のまとめ

本調査の結果と考察を、幾つかのポイントにまとめて示す。

(1) プロセス認識およびプロセス意識

質問に対し Yes, No の切り分けがはっきりしている組織 (どの質問結果も比較的に 0% か 100% に近い組織) と、そうでないところがあった。切り分けがはっきりした組織 (プロジェクト) ほど、構成員のプロセスに対する見解が一致して明快であると

考えられる。このファクターは「プロセス認識」と呼べるだろう。プロセス認識が高ければ「自分達のプロセスを (プロセスとして) 意識する度合」(プロセス意識) も亦高いと想像出来る。

一般に「現場は保守的」と言われるが、プロセス意識が高い程、なぜプロセスを改善した方が仕事がやりやすくなるのか理解しやすいと考えられている。しかし、今回の調査から推し測ると、現場では一般にこれらの意識はあまり高くないと見られる。

(2) 各企業のプロセス管理活動はあまりなされていない (レベル1)

今回調査では独立系企業が主で、成熟度レベルは全社とも 1 であった。アンケート結果を端的に一言で表すと、精密に管理されたプロセスが "ない" 事が明らかにされた、と言える。プロセスの信頼性が低いので改善すれば効果が大きいと見られる反面、開発当事者の間で「プロセス意識」が乏しいので (当事者が「改善した方が良い」とあまり思っていないので) 改善には多大の努力を要するだろう。

(3) 会社、業務形態、職種間の差異

業務形態、職種間毎に差異をとって調べると、結果の多くは、これまで経験的·定性的に現場で言われてきた特徴に対応する事が分かった。また、対応しないものは、示唆的であった。

また、会社毎に統計をとると回答パターンが非常に違った。荒っぽい言い方をすると、この差異は日米比較で示したパターンの差と同じ位大きい。個々の会社のパターンは、会社の成り立ちを良く表しているように思われた (本レポートでは詳細を割愛している)。

(4) 分業の未分化について

3.2.2 で述べたように日本では SQA の設立に消極的である。ここで質問 1.1.3, 1.1.6 (SQA、構成担当者の設置についての質問) 以外の、特別の役割やセクションの設置に関する質問についても調べてみた。その様な 4 つの質問 1.1.7, 1.1.4, 1.1.5, 2.1.7 では、上述の質問 (1.1.3, 1.1.6) と同様 Yes の割合が 15%, 18%, 6%, 11% と非常に低かった。ここで、質問 2.4.7 (第一線マネージャの権限) で日本の方が Yes の割合が高い事を考え合わせると、米国では良かれ悪しかれ分業が進んでいて、日本では相対的に未分化である事を示唆しているように思われる。

もっとも、この差は日米差ではなく、開発の性格的違いによるものと見た方が良いかも知れない。というのは、本調査では殆どの回答者がカスタマー·ソフト開発 (商業的一品開発) に従事している。しかもその半数以上は事務系アプリケーションに従事している。米国データの方は、殆どが軍事を含む制御系アプリケーションに属すと見られる (少なくとも事務系アプリケーションは殆ど含まれていない、と言われている)。この様に、開発と一言で言っても、内容·性格が違えばプロセス向上の条件·手段等が異なってくると思われる。

(5) 回答間の相関について

当初この調査を始めた際に、期待としてクロス分析 (質問間の相関分析) によりプロセス管理の「背後にある力学」のようなものを探れるのではないか、と考えた。結果としては、「自分専用の端末·ワークステーションがあるか」との質問に対し多くの質問との間に負の相関があったものの、その他の結果には見るべきものが無かった。

これは、開発現場毎に事情が異なっているので、もっと個別に当たらないと一般的な法則性はつかめない、という事を示唆しているように思える。実際、会社間の回答パターンには非常に開きがあった。もう一つの指摘としては、レベル 1 は "fail スコア" であり、レベル2や3で指摘されるような組織間の (プロセス管理に関する) 共通性が無いのであろう、という事である。回答が変化に富みしかも相関が殆どないと言う事は、比喩的には一種の "混沌" であったとも言える。

(6) 今後に向けて

本調査の延長線上に考えられる一つの目標として、より具体的にレベル向上の条件を明らかにして行く事が挙げられる。その為には以下の点に注意する必要がある。

○「効果的な目標は何か?」を突き止める為には、個別の現場の実態に即してプロセスを分析する必要がある。まずプロセス向上の "障害要因" を突きとめる必要がある。

○ 現場の声を無視してレベル向上はあり得ない。現場のキーマン達の声にまず耳を傾ける必要がある。

これらの点に注意しながら、1991 年度から現場のプロセス調査を開始し、続行中である (但しこの調査は直接成熟度モデルを枠組として行なっているものではない)。

4. 成熟度レベル向上の指針

ここでは、現時点で考えられるレベル向上への指針をまとめると共に、指針への考察を行なう。

4.1 なぜレベルは向上させた方が良いか？

話しの順番が逆になってしまったが、現場では「なぜレベルを向上させた方がいいのか？」という質問をかなり耳にする。そこで、それについて先に筆者なりにまとめておきたい。

○ マネージャー·サイドから

現状として、現に生産性やプロセス信頼度のばらつきに悩んでいる。顧客との約束が守れないばかりでなく、採算も圧迫される。成熟度モデルの見方からするとこれは行き当たりばったりのプロセス管理の弊害である。レベル向上に向けての努力とは着手できる (着手してもっとも効果の上がる) 所から手をつけてプロセス管理を向上させて行こうとする事である。ちなみに成熟度レベルとは組織毎に優劣をつけるものではない。組織毎に合った改善方法を捜す為の道具としてレベル判定を行なうのである。

今年度 (1991 年度) は、ソフトウェア業界全体の採算が悪化しつつある。景気の後退している今こそ改革にふさわしい。

○ 開発当事者サイドから

プロセス向上とは一言でいって「見通しのよいプロセス運営 (プロジェクト運営) をしよう」という事に他ならない。この為には例えば設計やウォークスルー (レビュー) などの前段階に力を入れて後段階でのトラブルを潰しておく必要がある。

現場の声としては、しかし、前倒し的にレビュー (ウォークスルー) しなくてもデバッグでつぶして行く方が簡単だという声が多い。特にさほど信頼性要求の高くない画面構成などやさしい処理の場合に言われる様である。しかし筆社が観察した範囲で言うと品質に関して見通しを立てていないプロジェクトの人達程レビュー (ウォークスルー) を敬遠している様に思われる。後でつけが回っている。第一のポイントは見通しなのである。また、弊社では 4 年前から現場の作業中のストレスの調査を実施しているが、テスト期間中のストレスは非常に高い事が分かっている。先が見えにくく抑鬱感が増している為と見られる。フォールト除去に関しても、後になってからあわてない計画性のある「ストレスの少ない」工程を実現した方が開発者として得ではなかろうか (その為には前段階での適切なプロセスを学ぶ必要があるのは言うまでもないが)。つまり第二のポイントはストレスである。ちなみにレビューでのストレスを論じる人がいるが、それは仲間内のレビューに慣れていないという点を問題にしている為である。

レビュー (ウォークスルー) だけでなく成熟度モデルで言うあらゆる制度化·役割分割·指針等は品質保証、見通し、プロセス·トラブル·シューティング、(そして暗にストレス除去) 等を実現させる道具となっている。

○ 組織の文化として

成熟度モデルは、『テスト結果をもっとプロセスに反映させよう』などの個々のプロセス改善をうたうものではなく、それらを互いに構成した統合的ビジョンを提出した点に新しさがある。この結果として、成熟度モデルは暗に組織文化に対する視点を持つようになった。

私見ではあるが、先の見えるプロジェクト運営はメンタルヘルスに貢献する。プロセスの可視性が高まって知識が共有化されると、作業に統一感が与えられ、またチームの構成員という自覚を生み出す。作業の統一感、メンタルヘルス、チーム構成員の自覚 (誇り) は企業活動に生活の場としての重みを増し、プロセスは「職場の文化」という形をとって定着する。つまりレベル向上の努力とは独自の文化の創成でもある。CI を持ち出すまでもなく、これは企業にとっても個人にとっても非常に意義のある事である。

4.2 レベル２キークエスチョンへの回答から得た指針

レベル２キークエスチョンへの回答結果から、「どうしたら、これをクリアするプロセスになるか」を社内のマネージャー達と討論した。その結果をまとめたものが以下のレベル向上ポイントである。

○ 1. プロセス意識

出発点として、プロセスの現状に対する現場の意識が高くなければならない。

○ 2. SQA (ソフトウェア品質保証グループ)

SQA を設立し、各プロジェクトが決められたプロセス基準を守っているか第 3 者的にチェックする必要がある。

○ 3. テスト・エラーの記録

カスタム・メードの受託を主とする独立系会社では決まった記録方式を確立しにくい。データ収集のポイントとしては、

1) エラーのデータ採りを機械化する。

2) ある時点でテストチームを編成し必ずエラーを記録させる。

3) テスト以前のフェーズで効果的なレビューを行ないエラーを減らしておく。

などが考えられる。

○ 4. ソフトウェア・サイズのトレース

サイズの監視は危険なプロジェクトの検出の為に非常に重要である。

○ 5. マネージャーのトレーニング

見積り知識は複雑で経験を積んだマネージャーの頭の中にしかない。それを有効に役立てる為には複数マネージャーの手順化されたレビューを導入する事が考えられる。それを含めプロセス中心の考え方を徹底する為にマネージャーのトレーニングが不可欠である。

以上であるが、実際には上の項目をいきなり実行しようとしても無理がある (少なくとも当社の場合)。最初はコストをかけてでも実験するという場合には良い課題になるかも知れないが、そもそもこのまとめ方には「受験対策的発想」という面がある。その意味で、もっと現場の実態に即したアプローチがあって然るべきであろう。

4.3 文献から得られる示唆

今度は文献からどんな示唆が得られるか見てみよう。"進む方向を見定める" (次の目標を設定する) 為には、"地図の上で現在地点を確認する" (現在状況を見定める) 必要がある。文献 [3] には、レベル 1 という「現在状況」について示唆的なポイントが幾つかある。

(1) 低い成熟度の組織の特徴とその問題点

レベル 1 組織はプロセス向上の "出発点" であり、以下のような特徴を持つとされている。

○ 場当たり的な方針決定、そして解決方法。
○ 問題点・課題を理解している上級管理者があまりいない。危機的状況を先送りする。
○ 仕事を計画する手順があっても、正しく使われているかどうか確かめる仕組みがない。
○ 問題が生じた時、手順を放棄して "コーディングとテスト" に逆戻りする。
○ 成熟したプロセスから得られる利益を知らない。自分の行動の結幕を理解出来ない。
○ インスペクション、テストエラー分析等がムダに見える。
○ スピードと本当の前進の区別が付かない。

(2) 突破口 (The Way Out)

成熟度モデルの主張によれば、上述の点を改善するには組織ぐるみで改善を実行しなければならず、そのためにはマネジメントサイドの強固な意志が必要とされる。SEI がかつて遭遇した事のあるレベル 1 の組織では、以下の 4 点のうちいずれかが弱かった [3]。従って、レベル 1 組織は以下の点に特に注意する必要がある。

○ プロジェクト管理 (プロジェクト・プランニング)
- プロジェクト管理の基本は、分担に応じてしっかり成果物を上げる事である。
- その為には、適切な準備、明確な責任の所在と公知、実行へ専念させる事、が必要である。
- 出発点は、作業量見積、最適スケジュール、資源の見込みを付ける事、である。

○ 上級マネージャによる管理 (プロジェクト・マネジメント)
- 契約前の開発計画レビュー・承認が必要である。

・四半期毎にプロセスの遵守、品質活動、スケジュールの追跡を行なうべきである。
・ここでは費用の傾向やプロジェクト毎の品質・生産性をレビューする。
これをしっかり行なわないと、過度の約束や費用の高騰を招きプロセスが不揃いになる。

○ 品質保証グループ (SQA: ソフトウェア・クォリティ・アシュアランス)
・これは、予定された方法で開発作業が実際行なわれている事を、管理的に保証するものである。
・これを効果的に進める為に、SQA は管理者への独自の報告経路を持つべきである。
・充分な人的資源 (一般に開発組織の 3 − 6 %) を確保する必要がある。

○ 変更管理 (コンフィグレーション・マネジメント)
・これは技術面での安定の為だけでなく、費用のコントロールの為にも重要である。
・要求変更の多くは延期したりあとから組み込んでもよいものである。
・設計やコードの変更。
これをしっかり行なわないと、秩序だった設計・作成・テストが不可能になる。品質計画が有名無実化する。

以上の内容は、とても説得力があるけれども、ここから更に個別に応じたブレイク・ダウンが必要と言える。

(3) 改革事例

プロセス成熟度モデルに沿った改革事例がいくつか報告されている。文献 [9] はレベル 2 から 3 になった Hughes Aircraft 社での事例である。文献 [10] は改革に着手した Texas Instruments 社での状況報告である。

4.4 その他の留意事項及び考察

その他の、レベル向上に関する考察をまとめている。

(1) カスタマー・ソフト開発とレベル向上の関係: 我々にとっての成熟とは何か

今回の調査対象は殆どが独立系ソフトウェアハウスである。こでのカスタマー・ソフトウェア開発には、次のような問題がある。

・顧客毎・開発毎にプロセスが全然違ってしまう。
・「数の効果」を生まない分だけ、プロジェクト毎に信頼性やプロセスの仕組みにかけられる費用が限られている。

そのような中での「我々にとっての成熟 (の意味) とは何か？」という問いかけは、非常に重要であるように思われる。

そもそも、企業毎に成熟の意味は非常に違っている筈である。成熟度向上を手掛けた米国企業と、日本の企業には幾つもの差異があると指摘されている。今回の発表においても聴衆側から幾つか指摘がなされた。例えば MIS 系の開発は、軍用ソフト開発などに比べ遙かにやさしい。日本では大学教育にしっかりした SE 教育カリキュラムがなく「しろうとが組んでいる」。ある種の「人現管理」は優れているものの、そもそも「仁徳天皇陵のもっこ担ぎと法隆寺の宮大工」くらいに、開発システムも違うし働いている人も違う。またその他に、同じ日本でも製造業系と独立系では実態が違う。製造業系では例えば、OS のファクトリーなどで非常にレベルの高い組織が指摘されているし、レベル２、３など中間的組織も指摘されている [5]。独立系企業は努力が足りないという事だろうか？。

そもそも独立系の現場は、現行の成熟度モデルには乗りにくい点があるのかも知れない。米国でも MIS 系 (商業・事務系) の開発企業に成熟度モデルが試された事はまだないと聞く。そう考えると上述の 2 点が気にかかる。我々 (独立系企業) にとって「繰り返せるプロセス」とは何だろう。顧客毎にあまりにプロセスが違う場合、具体的に SQA をどのように設定する事が考えられるだろうか。各々のプロセスが異なる中で、共通して支援出来る事には何があるのだろうか (社内留保の有効利用方法として何が提案出来るか)。

顧客毎にプロセスが違ってくる、という点に関して更に関連事項を以下に挙げる。

○ 独立系は特に顧客獲得競争の弊害が大きい
ネームバリュー・全体規模が小さいしハードを持たない分だけ顧客獲得競争が激しい。立場関係が背景にあり、なし崩し的な交渉が行なわれる事もかなりある。顧客に対する「きめ細かいサービス」が身上であるが、それは一歩間違えばなし崩し的にプロセスが解体される危険をはらんでいる。

○ トップ·プロセス構造を顧客との関係が支配する

本年度社内で行なっているプロセス実態調査の結果として、トップ·プロセスの構造を左右する要因として顧客との関係(立場の構図)が浮かび上がった。第一に、立場の強弱の組み合わせがプロセスの動的構造を決める。第二に、交渉担当者は、顧客との交渉(の成行き)に合わせてプロセスをリアルタイムで柔軟に変更して行く必要がある [8]。

○ 制度·分業化によるプロセスの切り分けが「遅れている」

日本側(独立系企業)データと米国データを比較すると、日本側(独立系)では制度·分業化によるプロセスの切り分けがあまりなされていない、という点が最も目についた。これは顧客毎にプロセスを変え、顧客との交渉でプロセスをリアルタイムに補正してゆかねばならない事と非常に関連があるように見える。

またこの他に「派遣形態」で作業するという事があり、注意を要する。派遣に近い作業形態では、成員の成熟度として相手方組織のものを論じる事になってしまう。本稿ではこの状況は取り上げていないが、独立系企業の組織としての統一性が保ちにくくなる要因であり大変重要な問題である。

(2)「生みの苦しみ」

これも聴衆側からご指摘を受けたポイントだが、例え成熟度が増せばストレスが減るとしても、レベルを上がろうとするそのプロセスにおいてはストレスが高まる恐れが多分にある。特にレベル1から2へ上がろうとする時は、プロセス意識が低い為に、なぜやり方を変えなければならないのか分からない。「やらせる」側と「やらされる」側の分化が生じて、ここにストレスが生まれる。

ある組織において、コードレビューを定着させようとした時、最初はかなり大変だった、という例が紹介された。但しある程度成果が出始めると開発者側(やらされていた側)から「やらせて欲しい」と言うようになったという事である。効果が体験されるまでが大変と言える。この辺りの事情はポテンシャル·エネルギーの比喩が適切なのだろうか。

5. アンケートで測り切れていないプロセス側面

アンケートでは測り切れていないプロセス側面についての、社内での調査や討論結果をまとめている。

(1) アンケートに表れなかった要因の考察

以下の点は、成熟度アンケートで測り切れていないと思われるプロセスの側面(アスペクト)である。

○ WS系マシン環境整備プロセスの巧拙がアンケートに反映されていない。

○ 小規模プロジェクトにおける個人プロセスの評価が抜けている

プロジェクトが小規模になるほど、グループとしてのプロセス管理が減り、個人プロセスによってプロジェクトの成否が定まるようになる。現に小規模プロジェクトを多く抱える部所ではそちらの方が気になる。

○ プロセスのダイナミックな側面

·ネゴシエーションにおける交渉担当者のビヘービア

前章で述べたように顧客によりプロセスが形成される側面が強いので、この交渉の巧拙は致命的に効いてくる場合がある。

·プロジェクト内部の細かなチーム·フォーメーションの巧拙

ここで上述事項に対する成熟度モデルの立場も示しておく。

○ 上記項目をよく検討すると、全てアンケートという形式自体の制約によって調べられなくなった項目と言える。画一的な質問だけでは分からない。例えばマシン環境構築の巧拙は、ツールの種類だけ調べても駄目で、経験豊富な人が実際に見ない事には判断出来ない。

○ SEIのアセスメントでは、その様なアンケートの足りない部分を現場でのヒアリングなどで補っている(特に対象現場での優秀なキーマンの分析能力が重要)。

○ 上述の側面に対し"メタ"な測り方というものは存在する。例えばあるプロセス能力を直接評価する代わりに、遂行能力のある人を組織として持って管理しているかどうかを評価する。

しかし、これらの点を考慮に入れても尚、以下の指摘をしておきたい。

○ マシン環境整備に関して SEI は、「レベルが上がれば使いこなせる」と言うが、現場としては「使いこなそうと努力している内に (何年もかかって) プロセスを身につける」という側面が強い。それに顧客の種類が多く、独立系企業にとってそれらを柔軟に使いこなせるスキルを持つ事は重要である。成熟度文献で指摘されている「マシン」には上流工程で使う高度なものが含まれているのかもしれないが、低い成熟度での、具体的ニーズに即したマシンの利用技術の重要性も、もっと強調して欲しい。

○ 顧客との交渉に関連する話題は文献 [3] では、レベル５になってやっと登場する (18 章。17 章にも若干関連事項がある)。しかし我々の認識では、ここレベル１でそれは考慮する必要があるし、むしろ当社にとっては今もっとも必要な事の一つである。契約や顧客との関係について真剣に考えれば、そこから逆に品質やプロセス管理の大切さに立ち帰って反省する事にもなる。

表 5-1 UNIX WS コミュニティと汎用機コミュニティのプロセスの差異

	UNIX WS コミュニティ (主に制御系)	汎用機コミュニティ (主に金融系)
成熟度質問回答パターン	0% が殆ど + 幾つかの項目で高い Yes 割合 (技術導入)	全体的にスコアが良く見えるが中間的スコアが多い。
	"分からない" のパーセント少ない	"分からない" のパーセント多い
技術導入	技術導入に関心高い (高いスコア)	技術導入にそれほど関心高くない
技術の浸透	基礎技術が属人的でなく組織で共有されている	技術が属人的な場合が多い
プロジェクト規模	小	大
構成員について	構成員が自分達の prject を良く知っている	構成員に prject をよく知らない人いる (長期で途中から入る人が多い為)
文化	白黒がはっきりしている	曖昧
プロセス	顧客から独立したプロセス (一括→社内開発, 但し部分開発も多い)	・外部のプロセスの影響下にある (受託しても常駐が多い) ・蓄積された部分もある (再利用可能なプロセス関連記述)
社会的背景	UNIX WS 系: 利益率高い	下請け比率高く "薄利多売"
プロセスの社内的評価	マシン環境をうまく取り込んだプロセスに対し評価。プロセス管理は個人ベースに傾く	"過去の遺産" をうまく利用するものの、独自のプロセスポリシーに欠ける。集団ベースのプロセス管理

(2) マシン環境によるプロセスの差異 (一つの例)

弊社の、主に制御系を扱い UNIX WS を業務でよく使うセクションと、主に事務系を扱い手作業の多いセクションで成熟度アンケートの回答を比較した。このグループの、プロセスに関連する傾向を、アンケートの傾向と共に 表 5-1 にまとめている。前者では、質問毎に回答が 0% か又は高得点かに分かれる傾向があり、全体としてはスコアが低かった。しかし技術導入に関する項目ではスコアが高かった。後者は、全体的にスコアが良かったものの、中間的な得点が多く "分からない" の回答が多かった。前者はマシンを巧みに取り入れたプロセスの構築で社内的に評価が高いが、プロジェクトの平均規模が小さく、個人のプロセスが重要視される傾向にあった。この様なプロセスは、アンケートで評価されなかったと言える。この事から一見スコアが高くても曖昧な回答は評価出来ないのではないか、またマシンを使った開発環境及びその構築プロセスに関しては成熟度質問票の評価項目が不充分なのではないか、との結論に達した。

(3) 関連する文献

文献 [6] には成熟度モデルに対する幾つかの批判のポイントが挙げられている。中心的なものはアンケート調査という手段の限界の指摘と、レベル 4、5 で中心的となるレベル改善のパラダイムに対する批判である (その他この文献では成熟度アンケートが米国において企業のランク付けのスタンダード化しそうな風潮を強く非難している)。その中でやはりマシン環境に関する質問が欠如している事を指摘し、これはマシン環境に関してアンケート調査という方法では本当の開発組織の持つ力量や問題点が分からないのでそれと知りつつ抜かれたのではないか、と推測している。この筆者 (の一人) に上述のグループ間の比較を述べた所 All or nothing パターンになるという所に非常に興味を示してくれた。というのは、彼らに言わせると「それが普通」だからであ

る.

6. おわりに

(本章は、SEAMAIL 投稿の為に書き加えたものです)

本報告の前半は成熟度質問票に基づくアンケート調査の解析であった。これには、成熟度を離れてもそれなりの実態調査としての面白さ (資料価値) があると思う。反面、レベル 1 しか調査に現れなかった為、成熟度の段階を踏まえた比較研究は出来なかった。その後レベル 2 への道を模索したものの、状況把握が一般的すぎる事や独立系ソフトウェア·ハウスの独自の問題の位置付けが難しい事などの為に、現状として総論賛成『各論不明瞭』の状況が続いている。『何を最初にやれば一番うまくゆくか ? 』という所でどうも悩みがある。「テストの記録をつけて分析すればいい」と言い放つのは簡単であるが、現実にどれ程の労力でどれだけの成果が得られる (変化が起こる) か予測出来ずにものを言っても何にもならない。逆に "ここが肝ドコロだ" と分かるなら、例え一見強引に見えるとしてもはっきり『まずこれが大切』と言い得るのである。本報告以降の私共の調査研究の示唆する所では、利益の計上の仕方のような会社全体のシステムから最初に手をつけるべきではないか ? と思わせる "ふし" があるのだが (当然会社によって違う部分だが..) これはまた今後の調査という事で本稿の範囲を外れる。

ここで、成熟度自体に対する筆者の見方を試論としてまとめさせて頂きたい。そもそも成熟度モデルとは何だろうか ? これを考える上で以下のような整理が必要ではないかと思う。

組織論的成熟度 (モデル) ---- (1)

↕ (2)

計測学的成熟度 (モデル)

組織論的成熟度 (モデル) とは、もともと W.S.Humphrey 等が考案したモデルのアイデアそのものを指す。成熟度モデルは、個々のプロセス改善要項を組み合わせて統合的ビジョンを提出した点に新しさがある。原著による成熟度の記述の背後には開発組織というものに対する有機的統一体としての理解が感じられ、これが "開発組織論" として成熟度モデルを読む時の面白さを作っている。しかし言うまでもなく、この意味での記述体系は工学となる以前の日常的言語の上に築き上げられている。一方、成熟度モデルはアンケートを軸に工学的領域とつながろうとしている。これが計測学的成熟度 (モデル) である。ここで問題は 2 つに分かれる。

(1): そもそもの組織論的成熟度のアイデアとしての正しさ·一般性。
(2): 計測学的モデルと (1) のアイデアが対応しているかどうか。

(2) は (1) の正しさ·一般性があってこそ論じる事が可能になるであろう。いまここで (1) と (2) に "決着" を付けようというのではない。しかし関連する事項を並べて、読者の方々に独自の判断を促す事は出来ると思う。

- o アンケート自体の "測る道具" としての "精度" については 5 章で述べたように使った我々としても、また公にも、批判がある。
- o しかし一方 SEI がこれまでのデータを調べた所、レベル 2, 3 の分け方に関しては統計的に "internal consistency" があったと文献 [11] に載っている (Cronbach's alpha に言及している事から群としての独立性という意味だと思うのですが、お詳しい方がいたらご教示下さい) 。これ自体は組織論的成熟度モデルとアンケートが対応している証左にならないが、少なくとも "何かあるな" と思ってもいいらしいという事は分かる。
- o レベル分けに関しても批判があるものの主にレベル 4, 5 に関連するものが多く、レベル 2, 3 (の分け方) に関しては特に強い批判は出ていないように見受けられる (レベル 4, 5 に関するものとしては前掲の Boillinger の批判 [6] の他に例えば Caper Jones の "プロセスが重過ぎる" という批判がある) 。
- o (1) の組織論的成熟度モデルは、意味合いの差はあるものの実践的なソフトウェア·エンジニアリングの世界では広く受け入れられている様に思う。特にもっとも総論的な部分、すなわち「プロセス管理を改善して行くと幾つかの決まった特徴的な段階を経る事になる」、「従い、組織の現状を知って適合する改善ポイントを見つける事が重要」、「組織ぐるみでのプロセスの向上が必要」、「開発規模が大きいほど組織プロセス管理が効いてくる」などの点に関してはほぼ間違いなく認められていると言える。昨年秋に米国電気メーカーのソフト部門スタッフと会う機会に恵まれたが、成熟度という概念自体 (組織論的成熟度の方) は完全に定着しているように見受けられた。

o 組織論的成熟度は工学的 (或は数理的) 記述でないから、その意味で不可避的にあいまいさを持っている。理解に属人性がある。SEI もそれを承知しているからコンサルテーションや成熟度評価のスタッフの訓練には手間をかける。

成熟度モデルが最も成功した例として SEI のコンサルテーションがあるが、ここで現場から参加したキーマン達の意見は非常に重要とされる。アンケート結果は『あたりをつける』為の目安であり、本当に重要な改善ポイントを見つける作業はキーマンを含むスタッフ達の直感を大切にした (それ故不可避的に属人的様相を持つ) ものである。この事は何を意味するだろうか？『コンサルテーションとはそもそも属人的なものだ』と言ってしまったら身もフタもない。組織論的成熟度の理解がよほど属人的なのか？現場理解が属人的なのか？組織論的成熟度には言い足らない、従って直感的に補完しなければならない部分が多いという事だろうか？また計測学的成熟度 (アンケート) が部分的にしか当てにされていない事は、(2) の問題に通じる。

o SEI としても上述に関連してレベル記述を精緻化 (あるいはもっと分かり易く) しようとしている [12] [13]。

そもそも筆者が '90 頃文献 [3] [4] などを読んでいた時、アセスメント実施方法や関連プロセス技術は詳しいのに、肝心のレベルの記述や考え方が充分に応用できずコンサルテーションに想像のつかない部分があり、一種エソテリック (秘教的) なものであるかの思いに囚われた事がある。DoD 関連技術だったせいで、情報として完全に外部に流通していない部分があって、これが一種エソテリックな雰囲気を醸し出した。これが、成熟度モデルの是非をめぐって "世間を騒がせた" 原因の一端ではないかと思うのだがいかがだろうか。その意味で文献 [12] [13] の方向性はエソテリックな雰囲気を一掃し、属人性を少なくし、アンケートの構成に手懸りを与えるものである。組織論的成熟度の記述自体が変化しつつあると言える。但し精緻化は一般性を損なう恐れを常に持つ。この整理が、もともと直感が持っていた重要なものを落としてしまわない事を願う。

ここで幾つかの批判に意見を述べさせて頂く。筆者の回りではそもそも明確に立場を持った批判らしい批判を聞く事は少ない。『レベルの高い方が優秀だというのが気に食わない』とか『レベルなど上げなくて良いと思うから関心がない』などという意見があるが、これは結局レベルが低くても良いと言っている事になる。これは立場として考える時、誤解しているか成熟度の考え方自身を信じていないかどちらかである。最初からプロセスに関して何も問題がないと思う場合は別として、問題を感じていてかつレベルが低くて良いと言っているなら、それは「私は苦しんでいる。そしてこの苦しみのままでいい」と言っている事になる。これは「もっと楽になる方法などない (成熟度そのものへの不信)」と思っているか、あるいは状況設定そのものを理解していないかどちらかである。不信があれば不信として表明すべきである。また、「回りも苦しんでいるから別にいい」は、自分が苦しんでいる以上理由にならない筈である。尤も「楽になる為にとても苦労するならいい (レベルを上げるのが大変過ぎる)」という諦めはあり得るかも知れないが...

また『文化が違うから適応出来ないだろう』という人も居るが、(筆者自身この報告書の中でそれに近い事を書いたにも関わらず) この意見から、かなり『ノイズ』をフルい落さなければならないと思う。成熟度モデルで導入を勧める制度・役割分割・指針等 (成熟度アンケート及び文献 [3] で推奨されるもの) を見てみると、運用レベルでは多少の差が出るとしても本質的効果にはさほど差異がないと思われる。例えば荒っぽく言って人はなぜプロセスを書いておかなければならないのだろうか。もっとも素朴な答えは「人は忘れ易い」からである。人が忘れ易いのは洋の東西とは何の関係もない。このように文化の差異以前の「行動学的レベル」の人間の性質がプロセスと制度のあり方に最も関与していると思われる。「テスト結果を次に生かす」とか「コンフィグレーションの担当をはっきりさせる」事の重要性には洋の東西で何の違いもない (尤もパッケージ開発・カスタマイゼーションとカスタム・ソフト開発でコンフィグレーションの内容が違うという事はあるとしても)。また、『組織はレベル 1 からレベル 3 に直接移る事は出来ない。なぜなら繰り返せる (レベル 2) という事を経験せずにプロセスを定義しようとするから、納期のプレッシャーで管理側が押し潰されてしまうだろう』 [12] という言明を見て、洋の東西で事情が変わると思えるだろうか。確かにもっと微妙な問題もある。それらは今後も調査等を通じて事情を明らかにして行かなければならないだろう。例えば SQA であるが、それが出来にくい (或は出来易い) 風土がどうやって生じるのか調べる事は重要である。しかし SQA の本質的機能や必要性は同じではないか。製造業では当り前の組織である。SQA については米国でも非常に誤解が多いと SEI の D.Kitson からコメントを頂いた事がある。

以上、ざっと成熟度を取り巻く論点をとりまとめて、いや寄せ集めてみたが、最後に何点か更に補足の必要を感じている。

o 筆者の属する独立系開発企業では成熟度モデルに対しある種のとまどいが感じられている。上述の文化論の続きになるが例えば独立系企業でとりわけ SQA 機能が実現しにくくなる事情はありそうである。というのは顧客との "距離が短かすぎる" と SQA 的な社内の品質管理以前にプロセス管理が顧客オリエンテッドになる (という傾向がある)。『顧客サービスは独立系の存在理由』とまで言う人がいる。上述の筆者の意見に沿って言うと、何らかの「行動学的レベル」又はそれに準ずる重大な条件の違いがあるのかも知れない。そう言い切れるかどうかは、今後もっと詳しく調べる必要があると思う。

o 文献 [12] [13] にしても、そこで使われている推論の一つ一つを見ると、推論の根拠を更に明らかに (精密に) する必要があると思われる。用いられているある種の推論は、慣れた社会学者から見ると非常にナイーブに見えるという気がする。本来、組織論的にあるいは集団社会学的にシステマティックに状況を解析し、より条件を明確にした推論を展開すべきである。それが却って理論をがんじがらめにするか、より鋭敏にするかは、創造性に富んだ理論化手法の適用次第である。尤も、社会学自身にしても理論化手法に関して長年の論争がある (例えば客観主義的社会学とシュッツの流れを汲むもの) 現在、成熟度モデルにそこまで求めるのは状況的に複雑過ぎるかも知れない。ただ根本的な事として、成熟度が描こうとした世界は既にそういう社会学的世界と繋がっているという認識が本来あるべきである。人間の振舞いについての推論は人間の振舞いを研究する学問の仕方で記述すれば良い。工学的精緻化が必要なら更にその後で行なうべきであって、その前にやってもナイーブな結果、あるいはひとりよがりの結果に終わり易いのではないか。これは私個人の無知への批判でもある (文献 [12] [13] を編集した B.Curtis は、心理学に詳く上述についてもある程度承知している人だと考えられる)。

結びの言葉として、いま如何にして成熟度の本質的アイデアをより発展させるべきか、色々な意味において試されていると言える。

参 考 文 献

[1] Druffel,L.E.,Lt.Col.USAF,et al, Report of the DOD Joint Service Task Force on Software Problems, Department of Defnese, July 1982.

[2] Watts S.Humphrey, "Statistically Managing the Software Process", CHANGE: NEW DIRECTIONS FOR STATISTICS AND COMPUTING, Vol.2, No.2, 1989.

[3] Watts S.Humphrey, "Managing the Software Process", Software Engineering Institute, Addison-Wesley, 1989.
「ソフトウェアプロセス成熟度の改善」, 日本電気ソフトウェアプロセス研究会訳, 日科技連, 1991

[4] W S.Humphrey, D H.Kitson, T C.Kasse, "The State of Software Engineering Practice: A Preliminary Report", Software Engineering Institute, CMU Technical Report, CMU/SEI-89-01, Feb, 1989.

[5] Humphrey, W.S., Kitson, D.H., Gale, J., "A Comparison of U.S. and Japanese Software Process Maturity," Software Engineering Institute, the proceedings of the 13th ICSE.

[6] Bollinger, Clement McGowan, "A Critical Look at Software Capability Evaluations," IEEE Software, July, 1991.

[7]「ソフトウェア開発に関するプロセス・モデルについての調査研究」、情報サービス産業協会、２－Ｊ００７、平成２年度

[8]「ソフトウェア開発に関するプロセス・モデルについての調査研究」、情報サービス産業協会、３－Ｊ００５、平成３年度

[9] Watts Humphrey, Terry Snyder, Ronald Wills, "Software Process Improvement of Hughes Aircraft," IEEE Software, July, 1991.

[10] Watts Humphrey, "Hardware and Software", Texas Instruments Technical Journal, Vol.8, No.3, May-June, 1991.

[11] Watts Humphrey, Bill Curtis, "Comment on 'A Critical Look'," IEEE Software, July, 1991.

[12] M C.Paulk, B.Curtis, M B.Chrissis, et al, "Capability Maturity Model for Software," CMU/SEI-91-TR-24, August, 1991.

[13] C V.Weber, M C.Paulk, C J.Wise, J V.Withey, et al, "Key Practices of the Capability Maturity Model"," CMU/SEI-91-TR-25, August, 1991.

ソフトウェア特許 および ユーザインタフェイス著作権 に関する LPF からのアピール についての会員アンケート回答集計

SEAMAIl Vol.6, No.6-7 に掲載された League for Programming Freedom からのアピールに関して，編集部から会員の方々にアンケートをお願いしました．回答を寄せられた人数は思ったより少なく 26 人でした．この問題に対する関心がそれほど高くないのか，それとも問題が問題だけに回答しにくかったのか，理由はよくわかりません．ともかく，以下に結果をまとめておきます．

Q1. あなたは，これらのアピールに示された LPF の意見に賛同しますか？反対ですか？

(1) ソフトウェア特許権問題について

□全面的に賛同する	14	**********
□部分的に賛同する	9	*********
□賛同しない	2	**
□どちらともいえない	1	*

(2) ユーザ･インタフェイス著作権問題について

□全面的に賛同する	23	**********
□部分的に賛同する	2	**
□賛同しない	0	
□どちらともいえない	1	*

Q2. あなた自身またはあなたが働いている会社が，過去または現在，ソフトウェアの法的権利に関するトラブルに巻き込まれたことがありますか？

□YES	6	******
□NO	11	**********
□知らない	9	*********

Q3. 日本におけるソフトウェア特許やユーザ･インタフェイス著作権問題が，法律上どのように取り扱われているか，御存じですか？

□よく知っている	2	**
□いくらか知っている	13	**********
□ほとんど知らない	7	*******
□全然知らない	4	****

Q4. SEA は LPF からのこれらのアピールに答えて何らかの活動を開始すべきでしょうか？

□そうすべきだ	16	**********
□もう少し状況を検討してから	8	********
□SEA はそんな活動はすべきでない	2	**

Q5. その他, 御意見またはコメントがあればお書きください.

回答 No.1

２つのアピールには全面的に賛成です．特に,「ソフトウェアの特許を廃止する１つの方法」として,「特許の対象領域からソフトウェアを除外する」という提案は有意義かつ効果的ではないかと思います．

Xerox, Lotus, Apple などが訴訟を起こしたという事実は新聞などによって報道されていたので知っていましたが, その背景については情けないことにほとんど知りませんでした．アメリカがこのような状態になってしまっているのでは，日本もそのうち「右にならえ」してしまうのでしょうか? 早速，状況を調べてみなくてはと思っています．

状況をほとんど知らずに,「活動を開始すべきだ」などというのは無責任かも知れませんが，あえていわせて貰いました．

回答 No.2

SEA の組織の位置付けに理解が足らないので，LPF の呼びかけに応えて活動すべきだとの答をかけなかった（法人会員には発言権がなく，純粋に技術者個々人の協会であるなら,「そうすべき」と回答するところ)．

このアピールに対して個人ではできることは，限られている．SEA ぐらいの組織が「ロビーイスト」(?) として活動するぐらいでなければ，影響力を行使できないだろう（ただし，効くかどうかは別)．

もし，SEA が，社会に対して働きかけをする組織であるなら，知的所有権だけではなく，Unicode などにも，もしかしたら行動をおこすべきなのかもしれない．

回答 No.3

以前にも同様な記事を見たが何もできなかった．何か協力したい.:-)

回答 No.4

特になし．

回答 No.5

現在の特許制度は産業の発達を目的とした産業立法であり，現在でさえ部分的に，産業の発達を阻害するような部分が存在する（特にアメリカに顕著)．もし人間の思考を具現化するソフトウェアに特許制度が適応されたら，明らかに，不合理ばかりでなく，産業の発達を明らかに阻害するのは間違いない．

また，ソフトウェアに特許制度を適応しなくても，特許制度の目的の１つである発明を奨励する機能は，ソフトウェアに関して失わないのは，過去の歴史からいっても，自明である．

LPF の活動に賛同しています．国内でもかような運動があれば，微力ですが支援したいと思っています．

回答 No.6

小生は，自分の属する部署で開発したソフトウェアの頒布と使用権利保護について仕事上携わっております．開発したソフトウェアの特許抵触については少しだけ経験があります．

そうした立場から，以下に感想を述べさせていただきます（個人的な意見は一番最後に述べます)．

(1) ソフトウェア特許のパターン

(1-1) 第 1 のパターン

各企業ともノルマで特許提出を押しつける．各個人は焼け糞でソフトウェア特許がすごく一般的なものであるにも拘らず出してくる．特許として登録されてしまう．最初から特許権利を主張するつもりもないそういうソフトウェア特許が溢れてくる．米国のようなドミノ倒しのような特許訴訟の下地ができてしまう．

(1-2) 第 2 のパターン

ソフトウェアの特許で確実に請求可能なものは，グラフィックに関するものである．特許で請求しようとするものがいる．しかも非常に単純なアイデアを．計算機上で実現したからといって，そんなことは誰もが考える．

(1-3) 第 3 のパターン

進んだアルゴリズムを開発する．使用に関して特許権を請求したいと願う．アルゴリズムを特許として申請する．

(2) ソフトウェア特許において気のつくところ

- 外から見て特許に抵触しているとわかるようなソフトの特許は特許でない．
- 外見からではソフトの特許は抵触しているかどうか判定のしようがない．わからない．開発者に利益が回収できない．
- 自己申告制度だとあまりに正直者が損をしすぎる．
- いつ訴訟を起こされるかわからない状況では開発者に対して意欲が存続できない．

だから，ソフトウェアは特許に関して完全フリーなものとして，存在させてあくまで著作権による保護をおこなう．

しかし，それでは，ほんとうに（コトバ通りの意味で「ほんとうに」）進んだアルゴリズムを開発した機関や個人は，これではあまりに浮かばれない．逆に一介の技術者としてはすごい発明をしてイッパツ当てたいという開発意欲も同時に存在しているとも考えるべきである．

特許は発明に関する詳細を登録することが目的で作られている．発展を促すためには，計算機上でいろいろなもっとも有効なソフトウェアを使用可能な環境が必要があり，ソフトウェアの内部の実現について明らかににする必要がある．

(3) 私の意見 (あくまで 1 人のソフトウェア技術者として)

アルゴリズムを開発する側もそれを使用してソフトウェアを販売する側も開発意欲を損なわないためには，次のようなことを立法化すべきだと思われます．

1. 外見から使用していることがわかるような誰もが思いつくソフトウェアに関する特許は特許として認められない．
2. 誰もが思いつかないようなアルゴリズムに関してのみ特許として認められる．この場合，アルゴリズム特許料は，累進課税のようにそれをソフトウェアとして実現し，かつ有償で販売される時だけその販売価格によって累進的に決定される．特許成立時にある程度の目安価格を決定しておくことがもっとも望ましい．
3. ソフトウェアを商品として有償で販売するためには，必ずそのソースコードを関係機関に著作権登録する必要があり，この時何のアルゴリズムを使用したかについて正直に白状させる．これを法制化する．ここで正直に白状しなかった場合，後で判明した時に，莫大な罰金が請求されるものとする．

そして，あらかじめ，決めておかれた累進的なアルゴリズムに関する特許料を支払う．開発したソフトウェアに対して無償で配布する場合に限ってのみ完全にアルゴリズム特許からは解放されるものとする．

すぐには到底実現できそうではありませんが，これが今の考えです．ユーザインターフェースの著作権に対してはあくまでフリーであるべきだと考えます．

回答 No.7

LPF 参加方法の詳細が知りたい．あるいは SEA で取りまとめていただけないか？（各個人が個別に LPF に問い合わせると，LPF が大変なのではないかと思う）

回答 No.8

ハードウェアメーカで働くソフトウェア技術者は，同じような悩みをみな持っていると思います．年何件というノルマがあって，無理に出願しているのが現状です．技術者のパワーの使い道として本当にもったいないと思います．私も社内的には活動して行きたいと思います．

回答 No.9

(1) 日本の現状を知る方法または状況そのものを知りたい．

(2) 過去に CAD システムの開発を IEEE の CG&A に発表したところエバンス & サザーランド社から，ウインドウ上でのクリッピング処理につき自分たちの特許を使っていないか否かの問い合わせの手紙が来たことがある．

回答 No.10

編集部からの巻頭メッセージの意味がよくわかりました．日本は元来ソフト軽視で，それが我々技術者の悩みであったわけですがアピールにある米国の状態に移行することがソフト重視になるとは絶対思えません．ソフト製品の不法コピー，コードの盗作等は厳しく制限されるべきですが，技法やインターフェースは人類の共用財産として大切に流布させるもので，特許で使用を制限し，特許料を徴収する対象ではないでしょう．

技法の特許が成立するなら学問·学習はまったく意味がなくなるか，特定の個人（企業）を潤すだけのものとなってしまうでしょう．以前，クイックソートの技法の特許化という話しがあった記憶がありますし，建設業界で周知の施工法に特許が認められ，回避策を探った経験があります．ソフトウェアに限らず特許の現状についてはまったく無知なのでもっと深刻な事態になっているかもしれませんが，たとえば TRON キーボードなどは法的にどうなっているのか，ご教示下さい．

日本も「特許で儲ける」という，ビジネスチャンスを探しまわっている野心的企業家または個人にウイルスのごとく蔓延しそうな魅力的アイデアは広まる可能性があり，運動を開始しても早すぎることはないと思われます．憎むべきは，悪意の特許商売ですが，対策には限度が予想されるため，「現在，特許化を見送り，これから皆で検討しよう」という主張には全面的に賛成します（ただし，Mac を買うな！という対策には応じ兼ねますが）．

回答 No.11

Q2 で YES と答えたが，それは次の理由による：すなわち，IBM の日立訴訟により，間接的に

(1) M シリーズで使っていた TSS エディタを別のものと替えさせられた（多分，元のエディタが IBM のものと似ているためと思われる）．当然，ユーザ·インターフェイスはまったく違ったものになってしまった．

(2) データベースプロダクトの変更と元のプロダクトのエンハンス中止を通告された．

回答 No.12

現在の状況はきわめて問題があると思われるが，まだ議論が十分なされているとは思えない．また，SEA はその性格上あらゆる主張に対して中立であるべきだ．ただし，これらの主張を積極的に公表するのであれば，よいと思う．

回答 No.13

● ソフトウェアは特許であるという理論が飛躍している．

5 万ラインを 2 人年で作れる優秀なプログラマがざらにいるわけがない．そんな優秀なプログラマが 2 人年 10 万ドル未満で雇えるわけがない．並のプログラマでも人件費は 2 人年 10 万ドルは楽に越える．しかも 5 万ラインなら 5 人年かかる．

ソフトウェアは簡単に複写できるからこそ，ハード

ウェア以上に複写を制限すべきであるという逆の論理も成立するはずだ.

- 論理の飛躍を除くと，この人はハードウェアを含む特許制度全体について，重箱の隅をつつくような批判をしていることになる．営々として続いてきた特許制度の関係者が見たら笑止千万な幼稚な議論である.
- ストールマン氏はおそらく人件費10万ドル未満で2人年で5万ラインを作るような優秀で安給料で満足している修行僧のような人なのだろう．そんな特殊な生活をしている人が，自分の判断基準でビジネスの世界を論じるのはおかしい．クジラを食料にしていない人が，捕鯨禁止を押しつけるようなものだ.
- この人は権利保持者の側の立場をまったく考察していない．ソフトウェア特許は既に大量に出願されているが，大きな問題は起きていない．小さなトラブルは当然起きるがハードウェア特許でも起きているのであって，別に注目すべきことではない.
- このような捕鯨禁止論者にも似た狂信者の活動に，SEAが賛同することはやめた方がよいでしょう．日本のソフトウェアハウスはほとんど特許活動を行なっていない．クジラを食べたことのない人達なので，このような捕鯨禁止論に簡単にだまされる恐れが強い．今回のアンケートでも，Q1で*賛成する*，Q3で*ほとんど知らない*と答える人が相当多いのではなかろうか．それでSEAの方向を決められたら困る．困るというより恥ずかしい.

回答 No.14

音楽の場合，編曲にも著作権がある筈だが，ユーザ･インタフェイスの場合はどうなるのか？

回答 No.15

1．ソフトウェアの特許について

ソフトウェアの定義がよくわからない．何かあいまいなものを感じる．このままで反対，賛成を表明し，行動を起こすのは誤りだと思う．ソフトウェア，ハードウェアによらない特許制度そのものの矛盾ではないだろうか？ 特許制度の運用にも問題があると思う．本来の趣旨から離れて自社の利害の防衛という悪い面が出ているのではないか？

2．ユーザインタフェイスの著作権について

個々のコードには著作権を認めてほしい．しかしスタイル，形式は皆の共有物であって欲しい．「コード」と「スタイル，形式」をどう区別したらよいのかよくわからない.

回答 No.16

ソフトウェア特許問題に関しては，保護すべき優れた領域も存在すると思われます．しかし，防衛的な特許申請や包括的で，すでに一般的に周知のアルゴリズムに対する特許は，害多くして益少なしでしょう.

LOOK & FEELの問題と保護すべきユーザ･インタフェイスの境界はハッキリしていません．しかし，使用者の立場からすれば，著作権問題を逃れるためにバラバラのインタフェイスを強制されるのは，かないません．害の多い知的所有権太りに手を貸すことはないと思います.

回答 No.17

私はこの分野は詳しくありませんが，基本的にこのアピールに賛成です．私の知る限り，現状のプログラム著作権の運用を見ると，「マーケットを独占したい．または，裁判に勝訴して使用料をせしめたい」などの後ろ向きな運用が目につくからです．もしかしたら本当に有益な特許として保護されるべき発明もあるのかも知れませんが，現状の特許申請手続きではミソもクソも一緒にされている部分が多いのではないかと思います.

特許審査手続きが整備される見込みのないまま，ソフトウエアに特許を認めることは好ましくありません．また，ユーザインタフェース･プロトコル･言語仕様などに著作権を認めることにも反対です.

回答 No.18

ライセンスに詳しい人から，「今アメリカで，著作権訴訟を起こされている某社がもし敗訴すれば，ウチで開発しているものの多くがそれに引っかかることになって，大変なことになる」という話を聞いたことがありましたが，このLPFのアピール文を読むまで，こんなオオゴトだとは思ってもいませんでした.

日本のソフトウエア技術者の多くは，私のように実状に疎いか，多分まだヨソゴトにしか思っていない人が多いように思います.

SEAとしては，まず真相を調査したのち，もし，それがソフトウエアの技術革新を阻むものであるならば，少なくとも，日本のソフトウエア技術者に真相を知らせると共に，SEAとしての立場を表明する必要があるように思います.

回答 No.19

LPFのアピールをよく読むと，LPFはソフトウェアの特許自体に反対しているのでなく，現在の運用方法が稚拙だから，ちゃんとした運用ができるようになるまで，一時的に凍結しようということのようだ．そういう意味では，このアピールに賛成してもよい.

しかし，このアピールには事実誤認ないしは事実を故意に曲解している箇所が多い．もう少し整理して，知的な財産を保護し，かつ技術の発展に寄与するにはどうすればよいか，前向きの議論をしたい.

たとえば，あたりまえのアルゴリズムやテクニックが特許化できないように，アルゴリズムやテクニックのエンサイクロペディアを作るとか，ソフトウェアの専門家を特許庁に送り込むとか....

回答 No.20

このアピールを読んで感じることは，特許法の問題点のうちで，特に審査の難しさあるいは審査の誤りを指摘しているに過ぎないように思われる．そうした特許法の弱点につけこんで，金儲けを企む企業が存在するからといって，特許法そのものを悪者にすることはできないと思う．ソフトウエアにも特許で保護する必要のあるものは存在する．特許法そのものよりも，むしろ審査する側に問題があると感じる．不勉強でよく分からないが，可能であれば審査の洩れや誤りを指摘し，特許の取消を求めるという行動を起こす方が，我々の取るべき方策としてはふさわしいと思う．

LOOK & FEEL やユーザ･インタフェイスについては，著作権問題を逃れるために，ユーザ･インタフェイスがメチャメチャになるのは願い下げである．私が Macintosh を好んで使うのは，ほとんどのソフトでユーザ･インタフェイスが共通しているというのが，その大きな理由です．ユーザ不在の利権争いは許しがい行為ではありますが....

回答 No.21

ソフトウェアの権利を著作権で保護することには問題が多く，むしろ特許の方がふさわしいという議論を読んだことがあり（たしか，Pamela Samuelson だったと思うが），説得力を感じたことがあるが，LPF のアピールを読んで，特許にも大いに問題があることがわかった．

実際，メーカでは特許申請を社員のノルマとしており，実に無駄な精力を費やしている．出願申請することに意味があるそうで，ほとんどは公示前にとり下げる．むだな話である．

回答 No.22

時間がなくて，十分に検討するほど読んだわけではありません．また，たとえ時間をかけて読んだところで，他の意見を聞いてみないとなんともしっくりきません．

なぜ，そんなに躊躇しているのかというと，先日楽器売場の楽譜のコーナを覗いたのですが，著名な JAZZ の演奏家の演奏を採譜したものが売られていて興味があって見たところ，５線譜に特殊な記号が漫然とならんでいるのです．実に味気ないものです．でも，それを理解できて演奏できる人にとっては十分な独創性が感じとれるのだと思うのです．では，その譜面がどれくらいの値段かといいますと，一般の書籍と比較して高くも安くもないという常識的な値段です．音楽は，自分で楽しむ際に，人前で演奏してお金をとらない限りは，著作権協会にお金を払うことなど不要だろうと思います．でも，CD の製造販売や放送局での演奏などに関しては，少なからぬ金額がしかるべきところに払われているのだろうと思います．

プログラムの範疇に限って言えば，苦労して自分の作ったアイディアをみだりに盗むことはされたくありませんが，何事も次第に金と権力に目がくらんだ組織による主義/主張に発展されてくるのは，法律の真意がゆがめられてくるようで嫌いです．

回答 No.23

ソフトウェアは，工業生産物としての側面と文化としての要素の両面をもっているが，これからは文化の基盤的な使用法が増加して行くものと考えられる．ソフトウェアが文化として認められて行くにしたがい，「何でもかんでも特許に！」という動きは自然になくなって行くだろう．自動車のハンドル，アクセル，クラッチ等の配置に関する有名な話のように，ソフトの使われ方が一般の大衆に明らかになってくれば，ソフトウェア特許は馬鹿みたいな話になると思う．

SEA はそれまで専門家を中心に啓蒙活動をする必要があると思う．

回答 No.24

私がプログラミングを職業として選んだ理由はお金儲けのためではなかった．若干の知的好奇心を満足させて，生活の糧を稼げればよいと考えていた．その考えは今でも変わらない．

しかし，プログラムを書いて大金を儲けることは不可能ではない．そのことはすでに何人かの人々によって実証されている．かれらがもともと金儲けを目的としてプログラマになったのか，途中でそうしたビジネス意識に目覚めたのかはわからないが，拝金主義がある種の宗教として定着している現代の社会では，それらの人々は英雄としてもてはやされ，その結果，最初から瞳の中に￥や＄のマークを輝かせたプログラマたちが後を追って登場し，その周囲には猿回しとしての企業家や銀行家あるいは法律家たちがあらわれる．利権に敏感な政治家や官僚たちも黙ってはおらず，知的所有権などという妖怪がさまよい出たりする今日この頃である．

知的活動としてのプログラミングに自由をという LPF の主張は，この職業の本質に根ざしたアピールとしては正しい．しかし，拝金主義に狂ったこの世紀末の社会の中では，おそらく大方の支持を得られないだろう．だからといって，座して敗北を待つのは潔よしとはしないが....

回答 No.25

ノーコメント．

回答 No.26

ノーコメント．

第10回　夏のプログラミング･ワークショップ

現場でのシステム設計を改革するテクノロジを求めて

― 設計における問題点とはなにか？ ―

参 加 者 募 集

主催：ソフトウェア技術者協会

夏のプログラミング･ワークショップは，リーダ役のスタッフのもとに２０代半ばの「若手」が数名のグループに分かれて集中討論／発表準備／成果発表／報告書作成などを実践的に行なうもので，企業，大学の壁を越えた積極的な活動姿勢を身につけるのに大変良い機会だと定評のあるものです．ＳＥＡの内部では「若手の会」と呼ばれるこのワークショップも，早いもので10回目になりました．例年は盛岡で行なわれることが多いのですが，今回は「１年おきくらいに西の方でも開催したい」と考え，暫定的に場所を愛知県は岡崎に移してみました．

今回のテーマは「設計」です．今回の目標の１つは共通の課題をグループ毎に異なった方法論(オブジェクト指向，ジャクソン法，構造化設計など)で解くことにあり，これを足がかりに設計作業に関する討論を行なっていきます．

ワークショップの前半(9/3)は，グループ毎の勉強と共通課題の回答に取り組むことになります．個々のグループは全く別の活動を行ない，実際にグループがどのような結論を導くかは，すべてグループのメンバに任されます．

さて，せっかく複数のグループが存在するのですから，ただ独立に活動したのでは意味がありません．そこで，他のグループが一体どのような討論を行なっているのかを知るための場として，期間途中に「中間発表」を設けます．今回は形式を変え，中間発表もグループ構成を変えた小グループ内で並行して行なうことにします．このため，参加者全員が最低１回の発表機会を得ることになります．

後半(9/4)は，中間発表を踏まえての軌道修正を中心に，「設計作業」に関連して抱いている各人の問題点，採り入れている技術なども合わせての討論を行ないます．

最終日には各グループの最終成果をグループ毎に発表します．そして，その後の質疑／全体討論結果などをふまえて，解散後に「グループ活動報告書」を作成するところまでがこのワークショップの作業です．参加者が自主的に活動することを望みます．

この４日間の活動を通じて，参加者全員が自分自身の現在のポジションを確認し，それぞれの会社の将来を担う中核技術者としてのビジョン/スタンスを確立することに役立てられれば何よりです．

例年どおり，意欲あふれる多くの若手エンジニアの方々の参加をお待ちしています．

開 催 要 領

期　　日： 1992年9月2日（水）から9月5日（土）まで（2日13:00開始，5日14:00解散）

場　　所： ホテル　レク･ワールド岡崎（〒444-21 愛知県岡崎市鴨田町所屋敷8-3; 0564-25-1311）

定　　員： 25名

参加資格： 年齢30歳以下，経験2年以上のソフトウェア技術者

費　　用： 55,000円（SEA正会員） 70,000円（SEA賛助会員） 80,000円（一般）
現地集合･現地解散とし，期間中の宿泊費（2日夕食から5日昼食まで9食付き）
および資料代を含みます．

実行委員長： 渡邉雄一（アスキー）

プログラム委員長： 塩谷和範（ＳＲＡ）

プログラム委員： 岡本隆一（さくらＫＣＳ），桜井修（池上通信機），
後藤浩（シスプラン），福良博史（ＪＭＡＳ），豊田英利（セス）他の予定

申込方法： 裏面の参加申込用紙の必要事項を記入の上，1992年7月10日（金）までに，郵便またはE-mailで，SEA事務局宛お送り下さい．(入力の手間を省くためにE-mail等の電子媒体を歓迎します．フロッピの場合は，MS/DOSもしくはMacintoshのテキストファイルに限定させて頂きます.)
なお，申込者多数の場合には，プログラム委員会でポジション･ステートメントの内容を審査の上，8月初旬までに参加者を選定させていただきますので，あらかじめ御了承ください．

申込み先： **〒160　東京都新宿区四谷3 - 12丸正ビル5F　ソフトウェア技術者協会**
Tel: 03-3356-1077　　Fax: 03-3356-1072
E-mail: wakate92@sea.or.jp (junet),　MHG01727(NIFTY),　shioya(mix)

*******　次ページにに申込用紙があります　*******

******* 締切り 1992年7月10日（金）厳守 *******

送付先：　　申込日付：　月　日

〒160　東京都新宿区四谷3-12丸正ビル5F　ソフトウェア技術者協会（Fax: 03-3356-1072）

第10回夏のプログラミング・ワークショップ
参加申込用紙

氏　名：＿＿＿＿＿＿＿＿＿＿＿＿（ふりがな）＿＿＿＿＿＿＿＿＿＿＿＿

種　別：☐ 正会員（No.　　）　☐ 賛助会員（No.　　）　☐ 一般

年　齢：＿＿＿＿（歳）　性　別：☐ 男　☐ 女　血液型：＿＿＿＿

特記事項：☐ タバコの煙にかなり弱い　☐ 酒が飲めない　その他：＿＿＿＿＿＿＿＿

会社名：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

部　門：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿　役　職：＿＿＿＿＿＿＿＿

住　所：（〒　　）＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

TEL:（　　）－（　　）－（　　）内線（　　）FAX:（　　）－（　　）－（　　）

ポジション・ステートメント：

(1) 現在までの主な仕事の内容について

(2) 現在利用している開発環境について

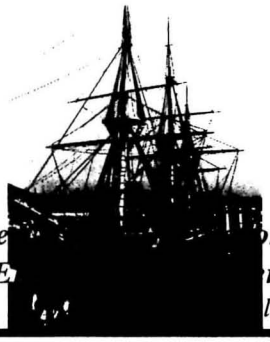
(3) これまでに現場でのシステム設計に採り入れてきた技術・技法などについて

(4) 以下のいずれかを選んで別紙（A4用紙に1～2枚；2000～4000文字）に述べて下さい．いずれの場合も，「何が」，「どのように」，「どうして」などを可能な範囲で具体的に述べるよう配慮して下さい．
特にb)の場合はその「改善の提案」も行なうようにして下さい．

a) 今までに経験した開発作業で，設計に感じた「ありがたみ」について
b) 今までに経験した開発作業で，設計に感じた「うらめしさ」について

CALL FOR PAPERS

15th International Conference on SOFTWARE ENGINEERING



The ... nal
Software E... nce
May 17-21, ... land

Organizing Committee

General Chairman:
Victor R. Basili
Department of Computer Science
University of Maryland
College Park, Maryland
20742 U.S.A.
(301) 405-2668,
basili@cs.umd.edu

Program Co-chairs:
Richard DeMillo
Department of Computer Science
Purdue University
West Lafayette, Indiana
47907 U.S.A
(317) 494-7823
rad@purdue.edu

Takuya Katayama
Department of Computer Science
Tokyo Institute of Technology
2-12-1 Ookayama,Meguro
Tokyo 152, Japan
011-81-3-3728-7663
katayama@cs.titech.ac.jp

Tools Co-chairs:
Michael Evangelist
Andersen Consulting
100 South Wacker Drive
Chicago, Illinois 60606
U.S.A.
(312) 507-6417
wme@andersen.com

Karen Huff
GTE Laboratories
40 Sylvan Road
Waltham, Massachusetts
02254 U.S.A.
(617) 466-4235
khuff@gte.com

Tutorial Co-chairs:
Clement McGowan
The MITRE Corporation
7525 Colshire Drive
McLean, Virginia 22102
U.S.A.
(703) 883-7099
mcgowan@mitre.org

Dieter Rombach
AG Software Engineering
Fachereich Informatik
Postfach 3049
D-6750 Kaiserslautern
Germany
011-49-631-205-2895
rombach@
informatik.uni-kl.de

Sponsored by
IEEE Computer Society Technical Committee on Software Engineering & ACM Special Interest Group on Software Engineering

ICSE is the Flagship Conference of the Software Engineering Community. The goal of ICSE is to serve the entire community with a balanced program of reviewed technical papers, panels, tutorials, tool demonstrations, posters, invited lectures, and workshops that bring together in one meeting the most current research results, the best new ideas about how to apply them, and the most comprehensive overviews of current technology and practice.

ICSE 15 will be held in conjunction with a number of more specialized workshops on topics ranging from software testing and analysis to configuration management. To foster increased participation by ICSE attendees, special workshop sessions will be introduced by mini-tutorials aimed at nonspecialists.

The ICSE 15 Program Committee solicits original technical papers, proposals for panel discussions, and posters. All contributions will be judged primarily on soundness, originality, and relevance to software engineering. The program committee anticipates that many submissions will describe ongoing work that will appear in more polished and complete form in scientific and technical journals, and will therefore place an emphasis on evaluating papers, posters, and panels on the basis of potential interest to ICSE attendees.

TECHNICAL PAPERS AND POSTERS may describe but are not limited to:

- Results of original theoretical or experimental research;
- Case studies;
- Large-scale data gathering and analysis;
- Industrial experience;
- New ideas for education, curriculum and training;
- Topical discussions of management, policy, and economic, or cultural issues.

PAPERS are limited to 6000 words with full page figures counted as 300 words and should include a short abstract, list of keywords, and lead author's address. Submitted papers should have been neither published nor submitted for publication elsewhere. Research papers should clearly indicate the relevance of the results to software engineering. Experimental papers should, in addition, contain clear descriptions of experimental methods used in obtaining the results. Experience and case study papers should discuss the relevance/applicability of the results to other environments and should not be used to advertise or promote specific commercial products or services. All papers should explain what is new and significant about the work presented and how it compares with related work. Accepted papers will be published in the conference proceedings. The program committee will select outstanding papers to be considered for publication in special issues of the IEEE Transactions on Software Engineering and ACM Transactions on Software Engineering and Methodology.

POSTERS will be limited to at most two 20 by 30 inch panels and should include enough material to allow attendees to determine the goals, methods and results of the work described. Poster sessions will be scheduled in public exhibition and demonstration areas throughout the conference meeting days. Posters will be considered in any of the areas described above, but may be most suitable for early phases of work-in-progress, student presentations, topics of highly specialized interest, new or controversial ideas, and work that has not yet attained a critical mass of results. Posters will be reviewed primarily on the basis of originality, relevance and potential interest to conference attendees. Poster presenters should submit one page abstracts describing the content and significance of the proposed poster.

PANEL PROPOSALS should include the title, a brief description of the panel goals, and probable panelists. The proposed panel chair should include a one-page biographical sketch. Panels will be selected according to their relevance to the goals of the conference and their potential interest to conference attendees.

DEADLINE FOR SUBMISSIONS: Eight (8) copies of technical papers, abstracts for posters, and panel proposals should be submitted to program co-chair Richard DeMillo by **September 15, 1992**. Authors and panel chairs will be notified of acceptance or rejection by **December 15, 1992.**

The TOOLS FAIR will allow exhibitors to demonstrate the latest technology in software engineering to an influential audience of technology decision makers. Technical presentations by the exhibitors will complement the demonstrations. Universities, research organizations, and commercial vendors are encouraged to submit proposals describing novel features of state-of-the-art tools and environments. Proposals must be submitted to Michael Evangelist by **February 1, 1993,** on the official application available from the Tools Fair co-chairs.

TUTORIAL PROPOSALS should address important software engineering topics in a comprehensive fashion. Proposals should include a detailed description of the topic, a table of contents for a full- or half-day tutorial, a characterization of the targeted audience, and the relevant experience of the tutor. Criteria for selecting proposals are: quality, relevance to attendees, and contribution to the overall conference program. Submit eight copies of tutorial proposals to the Tutorial Co-Chair, Clement McGowan, by **September 15, 1992.**

The LOCATION for ICSE 15 is the Baltimore harbor, which has become a model of urban downtown modernization and is a popular tourist attraction. It is within easy walking distance of the National Aquarium; the USS Constellation, oldest ship in the US Navy; the Maryland Science Center; the new Baltimore Orioles Park at Camden Yards, and the famous Harborplace shopping complex. Baltimore is within an hour's drive to historic Annapolis on the Chesapeake Bay, Washington, D.C., and numerous Revolutionary and Civil War-era landmarks.

Call For Papers and Participation

The Third Annual Symposium of the International Association of Knowledge Engineers (IAKE) November 16-18, 1992, Washington DC, USA

Symposium Chair: Dr. Elias Awad

"NEW GENERATION" KNOWLEDGE ENGINEERING
Serving the Knowledge-Aware Organization of the 1990s

Recognizing the growing role of Knowledge Engineering as it approaches the year 2000, this Symposium addresses KNOWLEDGE WORK and KBS DEVELOPMENT as two critical areas in building Knowledge-Based Systems. This 3-day event continues IAKE's tradition of emphasizing practical and advanced knowledge engineering applications and approaches to developing professionalism in knowledge engineering. The objective is to provide a forum for. and make available, the best practices and methodologies in this important area of technology.

This Symposium is designed around Workshops (technical, managerial, and education), Professional Panels, Tutorials, Exhibits, and Industrial Forums. A considerable number of quality refereed technical papers and industry-specific panels will form the Symposium Proceedings. Papers may be scientific, technical, or professional. They will be limited to 6,000 words and two pages of graphics. Panel Summaries are limited to 750 words and one graphic per panelist. English is the official language of the Symposium. Accepted topics include, but are not limited to:

KNOWLEDGE-BASED SYSTEM DEVELOPMENT

(Concentrates on technological trends, development techniques, knowledge-based systems)

Advanced/Emerging KBS techniques (neural nets, fuzzy systems, case-based reasoning, virtual reality, natural language) ♦ **Knowledge Analysis Methods** (acquisition, analysis of decision making, knowledge use in task environment, productivity analysis) ♦ **KBS Construction in the Domains** (medicine, manufacturing, scheduling, finance, operations research, etc.) ♦ **Embedding/Interfacing Techniques and Issues** (integration of KBS and data bases, front-end KBS in existing information systems) ♦ **KBS and Scheduling/Manufacturing** ♦ **Tool/System Evaluation** ♦ **Software Engineering and AI** ♦ **V&V of KBS** ♦ **Intelligent Systems--Today and Tomorrow**

KNOWLEDGE WORK

(Concentrates on knowledge: what it is, where it is, how it is used, how it is managed, and how it is communicated)

The Knowledge Engineer as Knowledge Worker ♦ **Role of Memory** ♦ **Reasoning** (beyond deduction to synthesis, metaphor, case-based, stories) ♦ **Applied Epistemology** (data, information, knowledge, wisdom) ♦ **Communication in Acquisition** (verbal, behavioral, group-based) ♦ **Maximizing Use of Existing KB Tools and Methodologies Based on Day-to-Day Operation** ♦ **Managerial Issues** ♦ **Knowledge as an Asset** ♦ **Justification for KBS Strategies** ♦ **Lessons Learned--Implications and Potential** ♦ **Who Owns Knowledge?**

THE PROFESSION

(Concentrates on skills, abilities, personality attributes, and career path of the knowledge engineer)

Preparing Knowledge Professionals for a Career ♦ **Career Opportunities and Career Pathing** ♦ **Ethical Issues** ♦ **Professional Standards** ♦ **Certification of the Knowledge Engineer** ♦ **Demographic Research** ♦ **Surviving the 1990's**

* *

Tutorial Proposals and Paper Abstracts are due by **April 1, 1992** (final papers due **June 15, 1992**), and should be submitted to: Julie Walker, IAKE, 11820 Parklawn Drive, Suite 302, Rockville, MD 20852 Tel: (301) 231-7826. For more information, contact:

Elias Awad, Symposium Chair (804) 924-3423
Mark Gembicki, Symposium Vice-Chair (301) 263-7616
Steve Oxman, Program Co-Chair (301) 266-1671
Quinton O'Neal, Program Co-Chair (408) 284-6347

Co-Sponsored by: Intelligent Computer Systems Research Institute, U. of Miami; US Air Force Logistics Command, AI Program Management Division; IITT International (France); and the Association de Ingenieros Del Conocimiento de Espana - AICE (Spain).
In Cooperation With: AAAI, ACM/SIGBIT, IEEE/TCES and ITESM

IAKE: Tel (301) 231-7826 • Fax (301) 770-4621 • CompuServe 71531,104 • IAKE@UC780.Bitnet
連絡先 〒107 東京都港区元赤坂1-7-10 ｸﾞﾗﾝﾄﾞﾒｿﾞﾝ元赤坂603 （財）国際ＡＩ財団
TEL. 03-3403-9805 FAX. 03-3403-9812

ソフトウェア技術者協会
〒160 東京都新宿区四谷3-12 丸正ビル5F
TEL.03-3356-1077　FAX.03-3356-1072