SEAMAIL

Newsletter from Software Engineers Association

Volume 9, Number 1 May, 1994

目　次

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，800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40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，東北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．

ＳＥＡＭＡＩＬ　Ｖｏｌ．９，Ｎｏ．１　　１９９４年５月９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　１，０００円（禁無断転載）

編集部から

☆

Vol.9, No.1 です．この号は期せずして，昨年秋からこの春にかけて行われたいくつかのイベントのレポート特集になりました．

☆☆

大場さんが報告してくださったIWSED は，奈良先端大の鳥居先生と Maryland 大の Basili 先生とが共同でここ数年続けてきた国際ワークショップから発展したものです．

☆☆☆

次に掲載したのは，今年 11 月に予定されている第 2 回の準備を兼ねて行なわれた 3 月の国内ワークショップのアンケート質問用紙．「結果は?」とお尋ねになりたい方も多いでしょうが，ワークショップ主催者の鳥居先生の方針は "Give-and-Take".「情報が欲しい人は自分でも何か情報を持っていらっしゃい．他人の話を聞くだけの人はお断り!」ということですので，秋に 2nd IWSED に出てみようかなとお考えの方はこれを参考にそろそろ準備してください．

☆☆☆☆

ISO 関連の標準化活動でアクティブに活動されている松原さんにオーガナイズしていただいた 3 月 Forum は満員の盛況でした．当日使われた OHP 資料をもとに，この新しいプロセス･アセスメント規格の概要を，編集部でまとめてみました．

☆☆☆☆☆

日本 PCTE ユーザ･グループ会員の塩谷さんには，そのキックオフ･ミーティングの報告にあわせて，この新しいレポジトリ標準をめぐる最近の動向をまとめていただきました．

☆☆☆☆☆☆

大阪の横山さんからは，SEA 関西分科会での討論結果の報告をいただきました．それぞれの支部や分科会での討論也発表を，必ずだれかがこうしてレポートしてくださると，編集部としても，またそうした会合に出られない会員の方々にとってもうれしいと思うのですが....

☆☆☆☆☆☆☆

さて，大場さんからのもう 1 つの原稿は，これまでこの誌上で論じてこられたリストラ問題を御自身で体験されたことの貴重なレポートです．こうした本音の文章は Seamail でなければ読めない! そういった原稿を他の会員の方々もぜひ，お寄せください．

☆☆☆☆☆☆☆☆

IWSED: 「ソフトウェア工学データの収集に関する国際ワークショップ」の報告

大場 充
(ソフトウェア工学研究所)

昨年10月18日から20日の3日間，京都府と奈良県の県境にある「けいはんな都ホテル」において，「ソフトウェア工学データの収集に関する国際ワークショップ」(International Workshop on Software Engineering Data: Defect Data) が開催された．本報告は，その国際ワークショップでの内外参加者による議論を簡単にまとめたものである．ソフトウェア・プロジェクトにおける品質面の定量的管理に関心を抱いておられる方々の参考になれば幸いである．

1. IWSED とは

今回のワークショップは，奈良先端科学技術大学院大学の鳥居宏次教授と米国メリーランド大学のビクター・バシリ教授が共同で組織・主催したものである．ワークショップの目的は，「ソフトウェア開発・保守において『実際に収集されているデータ』の理解とその利用に関する知識の交換」であった．

その背景には，ソフトウェアの品質に関する数値データが，日本国内でも，また欧米でも，さまざまな形で公表されているにもかかわらず，実際に国内または国際的な企業間での比較や実態の把握がまったく進んでいない状況に対する研究者たちの失望感があった．つまり，数値自体は学術的には何の意味もなく，それらが実際にプロジェクトのどんなフェイズでどんなふうに収集され，利用されているのかという「文脈」(コンテクスト) こそが重要なのではないか，という問題意識である．

ワークショップでは，このような「データ収集・分析の文脈を探る」という目的を達成するために，参加予定者に対して，事前に，収集データとその利用法に関するアンケートを配布し，そのアンケートへの回答を参加の条件とし，「クローズド・ワークショップ」の形式をとった．

アンケート案の作成には，バシリ教授，鳥居教授，ロンバッハ教授 (ドイツ)，ポーター助教授 (メリーランド大学)，松本助教授 (奈良先端大学院)，大場などのメンバーが参画した．また，国内では，これまでソフトウェア信頼性シンポジウムを主催してきた高信頼性ソフトウェア研究会のメンバーにも，アンケートの質問項目の検討に参加していただいた．

日本が開催国であるという好条件と，鳥居教授の積極的な働きかけにより，ワークショップ参加者の約2/3は日本の企業からの参加であった．海外からの主要な参加者には，バシリ教授を中心とした海外の大学の研究者が多かった．その中で，米国からはAT&Tのボッタ博士 (ベル研究所) とモトローラのステングライン氏 (移動電話インフラストラクチャ・グループ) の2氏が，ドイツからダイムラー・ベンツのヘーゲル女史 (社内情報システム・サービス) が参加された．

日本側の参加者は，日立 (3名)，日本IBM(2名)，NEC(1名)，東芝 (1名)，三菱電気 (1名)，日本ユニシス (1名)，NTT(1名)，オムロン (2名)，松下通信 (1名)，シャープ (1名)，ENICOM(2名)，電力中研 (1名)，SRA(1名)，さくらKCS(1名)，奈良先端大 (2名)，大阪大 (2名)，南山大 (1名) などであった．

外国からの参加者は，他にアメリカから3名，ドイツから2名，イタリアから3名，オーストラリアから2名であった．その中には，ドイツのカイザースラウテルン大学のロンバッハ教授 (英語読みはロンバック) や，オーストラリアのニュー・サウス・ウェールズ大学のジェフリー教授などの，ソフトウェア・プロセスやメトリクスの著名な研究者の顔があった．

会場の立地条件から，昼はホテルでの会議，夜はビール・ワイン・ウィスキーを飲みながらの会議室での雑談を余儀なくされた．2日目の夜は，夕食会の後も，雑談に花が咲いた．特に，メリーランド大学から来たバシリ教授とポーター助教授の2人は，日本人参加者の間を飛び回りながら，ジョーク交じりで日本の文化について議論していた．日頃，世界の第一線の研究者と接する機会の少ない日本の参加者にとって，かれらと直接対話できるまたとないチャンスであった．

会議のプログラム概要は，第１日目が，いくつかの組織によるアンケートへの回答の要約と，背景の説明に関する発表および質疑応答，第２日目が，分野別グループによるグループ討議，最終日は，各グループの代表者によるグループ討議の報告とまとめの議論であった．

ワークショップ２日目のグループ討議については，各グループに配属された日本人リーダに各分科会での議論の報告をお願いした．ご多忙中にもかかわらずご快諾いただいたことに感謝する．量的な制約もあり，著者の判断で原文を編集し，かなり短くしたことをおわびする．

日本人リーダによるグループ討論のまとめと，外国人リーダによる討論の総括をよく比較すれば理解できることであるが，日本人と外国人の認識にはかなりのギャップがある．外国人リーダのことばのはしはしには，「会議中に日本人参加者から聞いた日本の現状に関する認識」と，「会議の前に自分たちが抱いていた日本の現状に関する理解」との格差に対する驚きが隠されている．

端的な表現ではないかも知れないが，「日本の現状はこんなものか」といったような一種の失望感があったことは否めない．つまり，「期待値」が異常に高かったのである．かれらが聞いていたのは，「日本のデータ収集とその活用は組織的であり，自分たちのそれよりずっと進んだものだ」とする一種の「虚像」であった．

外国人参加者の何人かが期待していたことの１つは，「米国のソフトウェア産業を追い越した日本のソフトウェア産業の実情」ではなかったかと思われる．しかし，実際に会って話を聞いてみると，「程度の差こそあれ，日本も同じ問題で悩んでいる」ことを確認し，「決して日本が米国を追い越したわけではないこと」を知らされたのではなかろうか．

外国の一流の研究者たちに日本の実態を知ってもらったことは，それなりの成果だと評価できる．ただ，「虚像」と「実像」の格差に幻惑され，逆に，かれらが今度は日本の技術力を『過小評価』するのではないか，との危惧がないわけではない．ロンバッハ教授の総括にそのようなニュアンスを感じたのは，筆者のみではなかったと思う．

このような「異文化交流」は，ある程度の蓄積ができて初めて結果が出てくるものである．その意味で，継続が重要である．その過程では，両者とも欲求不満状態に陥ることが，何度となくあるだろう．今回，期待したほどの成果がなかったからといって，「無意味なことだ」と結論づけるのは早計である．むしろ，日本がリーダーシップを握って，積極的に働きかけ続けて行かなければならない．

最後になったが，今回のワークショップを主催された鳥居･バシリ両教授の献身的な努力と熱意に対し，参加者全員を代表して，心から謝意を表する．遠路はるばる，この会議に参加するために来日された海外参加者各位にも，感謝する．地理的には遠くはないが，さまざまな困難を克服してデータをまとめ，アンケートに回答し，多忙な時間を裂いて積極的に参加してくださった国内参加者全員にも感謝したい．

2. 第１日

ワークショップ第１日目，会議のオープニングと，外国からの参加者２名，日本からの参加者７名によるアンケート回答の要約と背景に関する発表が行われた．

2.1. オープニング

奈良先端大の鳥居教授が，主催者を代表して会議の開催を宣言し，ワークショップ全体の議事予定とノン･ディスクロージャ･アグリーメントについて，簡単に説明した．

その後，鳥居教授はアンケート結果の要約について説明した．説明資料には，組織の名称などは除かれており，統計的な分布だけが解るようになっていた．説明資料には，アンケートの重要な質問に関する回答の分布が示されていた．説明後，鳥居教授は参加者に対して，説明資料を参加者全員に配布することに関し，問題があるかどうかについての意見を聴取し，資料を配布することの合意を得た．

その後，バシリ教授が登壇しワークショップの目的について再度説明した．それらは，「どんなデータが収集されているかを知ること」，「データの意味するものを理解すること」であった．バシリ教授は特に，「データの文脈を理解することの大切さ」について強調した．

この「データの文脈」に関して，ニュー･サウス･ウェールズ大学のジェフリー教授は，「何をソフトウェア･エラーと見做すかは，国や文化によって違ってい

る」と指摘し,「ソフトウェア·エラーとは何か」についても議論すべきであると提案した.

最後にバシリ教授は,ワークショップの結果として新しいアンケートを作成したらどうかと提案した.

2.2 モトローラ

米国モトローラ社のリチャード·ステングライン氏は,移動電話インフラストラクチャ·グループにおけるエラー·データ収集について説明した.

同組織におけるエラー除去のプロセスは,インスペクション,単体テスト,機能テスト,サブシステム·テスト,システム·テスト,フィールド·テストからなっている.

同氏の所属する品質保証は,比較的新しい組織で,各開発組織からは独立している.この組織が設立される以前は,テスト部門(検査部門)が品質保証の責任ももっていた.現在の品質保証は,プロセス保証,プロセス監査,とリリース管理を主たる業務としている.また,同組織は小規模ではあるが,プロセス·エンジニアリングとメトリクスを専門とするグループをもっている.

モトローラ社では,ソフトウェア·プロセス·ビジョンと呼ばれる枠組を定義している.それは,SEI のCMM, IEEE1074 − 1991 規格, CIG プロセス仕様, 各組織への標準プロセスの展開, 自己評価(セルフアセスメント), 継続的なプロセス改善などの要素から構成される.特に,CMM と IEEE1074 が重要な要素となっている.

モトローラ社における用語の定義は,次の97なものである.「フォールト」とは「エラー」または「欠陥」をいう.「エラー」とは,開発工程におけるインスペクションで発見される問題をいう.「欠陥」とは,テスト担当者または製品のユーザによって発見される問題をいう.「故障」(現象)のデータは収集されていない.

あるプロジェクトでは,「フォールト」の97%は開発工程で取り除かれていた.フォールトの総数は,約1,000個であった.製品の大きさは,全体で約1,000KLOC,そのうち新規開発分は約100KLOCであった.

別の統計によると,インスペクションに時間をかければかけるほど,インスペクション中に発見されるエラーの総数は増加する.統計は,ソフトウェアのエラー密度と1行当たりのインスペクション時間との相関を分析したものである.

2.3 日本IBM

日本IBMの新谷氏は,同社のアプリケーション·ソフトウェア製品開発におけるプロセス,用語,ツール,プロジェクトで残される記録,データ分析法について説明した.

プロセスは,PPA(Programming Process Architecture)にもとづいて定義されているIBMの全社的標準プロセスをカストマイズしたものである.そのプロセスには,日本で開発されたツールや,日本で定義されたメトリクスなどを統合している.

同氏は,同組織で使われている「問題」という用語について説明した.「問題」とは,製品の「品質上の解決すべき問題」をいう.「問題」は,製品の「機能上の欠陥」である場合もあるが,それ以外の「改善を必要とする課題」(保守プロセスの問題や誤解を生みやすいマニュアルの記述など)である場合もある.

開発の各工程で報告される問題を管理するために,同組織ではQPID(Quality and Process Improvement Database)と呼ばれるツールを開発し,利用している.問題が発見されると,問題に関する記録が生成され,問題発見者や問題のタイプが記録される.問題解決の時点では,解決責任者,原因分析結果,問題が発生した原因などが記録される.

QPIDのデータベースに蓄積された問題データについては,信頼度成長曲線,工程別の単位規模当たりの発見問題数(問題／KLOC)などが分析される.また,問題状況報告書などもツールによって自動生成される.

同組織では,経験的に「単体テスト工程完了までに発見される問題の数」と「第三者組織による機能テストとシステム·テストの期間中に発見される問題の数」との比率を9対1にすることを目標としてきている.いくつかのプロジェクトでは,すでにこの目標を達成している.

最後に,開発チームがある開発フェーズから次のフェーズに移行できるかどうかを決定する場合に用いる品質確認のためのワークシートについて説明した.その後,IBMのチラレゲ博士によって開発された『直

交欠陥分類法』について簡単に説明し，同組織での直交欠陥分類法の適用の結果についても説明した．

2.4 シャープ

シャープの田中氏は，同社におけるソフトウェア開発プロジェクトで収集されたデータの分析について発表した．最初に，同社の概要を紹介した後，ソフトウェア開発工程についての説明があった．プロセス･モデルとしては，ウォーターフォール型のモデルである．

次に，テストと収集データについての説明があった．テストは，開発チーム自身によって実施される単体テストと，独立したテストチームが実施するシステム･テストからなっている．テスト工程で記録されるエラー･データは，残存バグ数の推定，製品出荷の判定，そして開発技術者の管理に利用されている．

バシリ教授から，「単体テストでもデータを記録し，残しているのか」との質問があった．これは，米国では一般に，開発作業者自身による単体テスト時のエラー発見記録は，公式な記録としては(信憑性が低いため)残されていないという背景から出た質問である．

これに対して田中氏は，「プログラミングを担当した作業者が，「デバッグの完了を宣言」した直後からデータを収集している」との答えがあった．バシリ教授の質問には，「本来デバッグ作業と本質的な違いのない単体テストを，より公式なものとする何か特別な方法が採られているのか」という別の質問が隠されていたのだが，それに対しては否定的な回答がなされた．

さらにバシリ教授は，モトローラとIBMの発表者に対しても同じ質問をした．ステングライン氏は，「単体テスト中でも，エラー記録が収集されているケースがないわけではない」ことを指摘した．また新谷氏は，「プロジェクトによって，単体テスト時のエラー･データの取り扱いは違っている」と述べた．

この後バシリ教授は，「単体テストを実施していない組織はあるか」と質問した．これに対しては，すべての参加者の所属している組織で，そのような例がないことが判明した．この質問の目的は，クリーンルーム･プロセスは「単体テストが諸悪の根源である」との仮定にもとづいており，単体テスト工程を取り除くことによってより高い品質のソフトウェアを開発しようとしていることがあり，産業界でもそのような試みがあるかどうかを聞くことであった．

バシリ教授のグループは，NASAのSELでクリーンルーム･プロセスの実験を試みたが，失敗に終わっている．その主な原因は，「プログラマ達が，単体テスト(実はデバッグの延長)がなくなることに強く反対した」ためである．質問の途中でバシリ教授がいったように，「単体テストは，われわれに最も分かりにくい工程である」のは事実である．

2.5 オムロン

オムロンの田中氏は，同氏の所属するATMや自動改札システムなどを開発している組織で収集されているソフトウェアのエラー･データとその分析について発表した．対象となるソフトウェアは，平均で約130KLOC，その約60%は再利用されているコードである．

ソフトウェア開発のプロセスは，ウォーターフォール･モデルにもとづいたものである．概要設計から開発試験(田中氏は「デバッグ」ということばを用いて説明した)の終わりまでは，開発単位別のサブプロセスが同時平行的に流れる．開発試験が完了すると，製品の品質保証のためのテスト(一般のシステム･テストに対応)が実施される．

品質尺度として，同組織では，設計･プログラム品質，開発品質，テスト品質，および製品品質の4つの尺度を定義している．設計･プログラム品質は，設計やプログラムのレビューで発見されるエラー数で測定される．開発品質は，デバッグで発見されるエラーの信頼度成長曲線から測定される．テスト品質は，開発試験の終了後のソフトウェアに残存し，テスト中に発見されるエラー数で測定される．製品品質は，ユーザによって発見されるエラー数で測定される．

同氏は，最後にレビューと開発生産性の間に観測される『弱い正の相関』について説明した．ステングライン氏の発表でも指摘されていたように，「レビューでは時間をかければかけるほど多くのエラーを発見できる」ことが，同組織でも確認されている．また，「レビューに時間をかければかけるほど，ユーザによって発見されるエラーは減少する」ことも確認されている．

バシリ教授は，「テストに投入されている労力が，工数全体に占める割合はどの程度か」と質問した．これに対する田中氏の答えは，「通常，約40から50%の間である」とのことであった．NASAのSELでの調査では，この割合は約50%であった．バシリ教授の頭の中にはこの数字があって，この点について，すなわ

ち，日本の一流企業での組み込み型ソフトウェアのテストがどれくらい本格的なものであるかを，確認しようとしたものであろう．

2.6 日立製作所

日立製作所の野瀬氏は，同氏が所属している情報システムの開発部門で実施されているソフトウェア品質データの収集と分析について発表した．同組織におけるソフトウェア開発プロセスは，ウォーターフォール・モデルにもとづいたものである．

同氏は最初に，同組織における『品質目標管理のシステム』について説明した．品質目標管理は，次のステップを通して実施される．品質目標の設定，予想信頼度成長曲線と管理曲線 (管理限界曲線) の決定，開発チームによるテスト中のエラー・データの収集と実績信頼度成長曲線の追跡，開発チームによるテストでのエラー・データにもとづいたエラー数の推定，開発チームによるテストの結果レビューとサンプリング試験による品質レベルの確認，テスト・チームによる第三者テストでの品質確認．

同氏は，あるプロジェクトを例として，そのプロジェクトで使われた管理表を示した．その表には，テスト項目数，成功したテストケース数，テスト・カバレージなどのデータが詳細に記録されていた．また，開発チームや品質保証の作業者が実際に利用しているワークシートについても説明した．

最後に同氏は，単体テスト開始後に除去されるエラーの工程別分布について説明した．それによるとエラーの約 78% が，開発チームによる単体テストからシステム・テストまでの期間に除去されている．品質保証のチームによるサンプリング試験で，ソフトウェアに残存する約 22% のエラーのうち約 1% が発見される．その結果開発チームにシステム・テストの継続が要求され，残存エラーの約 15% がこの期間に発見される．その後品質保証によるシステム・テストが実施され，残存するエラーの約 1% が発見される．

2.7 新日鉄情報通信システム (ENICOM)

新日鉄情報通信システムの熊本氏は，新日鉄と同社による共同プロジェクトでの収集データとその利用について発表した．同プロジェクトは，約 470 人月のプロジェクトで，収集されたエラー・データはプロセス改善に用いられる．

同プロジェクトにおけるシステム・テストの品質目標は，1KLOC 当たり 0.7 個以上のエラーを取り除くことであった．テスト後 (本番稼働への移行後) の品質目標は，1KLOC 当たり 0.3 個以下とすることであった．

収集されたデータは，エラー発見日，エラーの分類 (設計エラー，コーディング・エラーなど)，対処方法，エラー修正者の氏名，エラー修正日などであった．これらのデータにもとづいて，システム・テスト中に発見されたエラーの信頼度成長曲線，試用期間の間にユーザによって発見されたエラーの信頼度成長曲線，フェーズや原因別のエラー分布などが分析される．

2.8 ダイムラー・ベンツ

ダイムラー・ベンツ社のヘーゲル女史は，同社の社内情報システム開発における品質データの収集と分析について発表した．ダイムラー・ベンツは，自動車で有名なメルセデス・ベンツ社，AEG 社，ドイツ・エアロスペース社，とダイムラー・ベンツ社内サービスから構成されている．

ヘーゲル女史は，シュトゥットガルトに近いウルムにある社内情報システム・サービスの品質保証を専門とする組織の一員である．同社のソフトウェア・システムは，大部分が開発後 20 年以上を経過した「古いシステム」である．

同女史は，「ユーザ視点からのソフトウェア品質」に関するデータの収集と分析について発表した．この調査の目的は，「顧客満足度」(カストマ・サティスファクション) とソフトウェア保守の改善である．ユーザ視点からのソフトウェア品質として同社では，機能性，使い易さ，稼働率 (広い意味での信頼性)，性能，その他を定義している．

同社における 2 つの社内用情報システムのユーザの満足度調査によれば，同社が現在抱えている最も大きな問題は，ソフトウェアに関する文書の不備と教育の不足であった．古いシステムでその機能などに関する詳細な情報が完全でなく，ユーザがそれらのソフトウェアを使いこなせないでいるのである．

この調査の結果にもとづいて現在，同社では「コードの古さ」を測定するための尺度について研究中である．たとえば，ソフトウェアのインタフェースの数，情報の流れの複雑さ (サイクロマチック数などで測定される)，ソフトウェアの規模，文書化の度合いなど

を複合的に組み合わせた尺度が検討されている．

このヘーゲル女史の発表は，「ソフトウェア品質の概念」が国や組織によっていかに違っているのかを浮き彫りにした．ISO9000 の関係で，ヨーロッパの企業では，ソフトウェアに対する品質保証態勢を整備する必要性が出て来ている．ところが「何をソフトウェアの品質と考えるべきか」が，国や企業のおかれている状況によって異なっている．特に歴史のあるダイムラー･ベンツでは，「古いソフトウェア」が問題になっているようである．

2.9 SRA

SRA の松村氏は，アンケートに対する同社の回答の背景について説明した．アンケートの対象となったソフトウェアは，1988 年に開始された比較的古いソフトウェア･ツールであった．1992 年の後半から，1993 年の前半にかけて，このツールは大幅な拡張がなされた．この機能拡張プロジェクトのシステム･テスト期間と出荷後データが収集され分析された．

このツールは，メーンフレーム上で稼働している COBOL アプリケーション･プログラムの保守作業を支援するツール群から構成されている．ツール群は C 言語で書かれており，LAN で結合されたワークステーション上で稼働し，ソースコードの分析や影響分析を行なう．

ソフトウェア開発プロセスは，ウォーターフォール･モデルにもとづいたものである．ソフトウェアの規模は，コメントを含んで約500KLOC であった．収集されたデータは，エラーのタイプ (設計エラー，コーディング･エラー，保守エラー，導入エラー，ユーザ環境エラー，ユーザ･データ･エラー)，エラーの発見日時などである．

収集されたデータは，開発完了後の作業を改善するために利用される．たとえば，バグ修正プロセスの改善，次の機能拡張のための要求の収集，ユーザ教育や支援の改善などである．同プロジェクトでは，システム･テストとフィールド･テスト (ベータ･テスト) が同時進行していた．

2.10 NEC

NEC の砂塚氏は，同社のビジネス通信システム事業部における組み込み型ソフトウェア開発プロジェクトでの品質データの収集と分析について発表した．データは，同事業部ソフトウェア開発部門の品質保証グループによって収集され分析されたものである．プロジェクトには，子会社や協力会社が参加していた．プロジェクトに参加した技術者の数は，232 名で，その約 20% が NEC の社員であった．ソフトウェアの規模は，約 1,300KLOC であり，約 1,000 システムが現在稼働している．

プロジェクトの期間中品質生産性向上運動が実施された．その運動では，プロセスを定義し，ソフトウェア品質保証システムを定義し，統計的品質管理の実践に関する定期的なチェックすることを重点とした．開発チームは，単体テストと機能テスト (「インテグレーション･テスト」) を実施する．システム･テストは，テスト･チームが実際のハードウェアを用いて実施する．機能テストの目標は機能の確認であり，システム･テストの目標はユーザの視点からの品質保証である．

品質向上のステップは，品質目標の設定，問題の分析，問題の設定，解決方法の定義，結果の評価からなる．これらのステップは，日本的品質管理のソフトウェアへの応用である．

同部門におけるソフトウェア開発プロセスは，ウォーターフォール･モデルにもとづいている．エラー･データの収集は，機能テストの開始とともに始まる．データとしては，エラーの原因や混入フェーズなども記録される．設計工程では，エラー数だけが記録されている．発見されたエラーを記録するために使われるワークシートには，エラー検出時期 (時間とフェーズ)，分析結果 (原因と混入フェーズ)，修正内容 (機能コード) が記入される．

これらのデータについての典型的な分析としては，エラー原因別の分布，混入フェーズ別の分布の統計的な解析がある．たとえば，フェーズ別で見た場合，設計エラーが最も多い．

2.11 グループの編成の説明

鳥居教授がグループ編成の試案として，２つの「組み込みソフトウェア･グループ」，「ビジネス･ソフトウェア･グループ」，そしてシステム･ソフトウェアを含む「その他のソフトウェア･グループ」の４つのグループに分けることを提案した．参加者はこの案に賛成した．参加者は，自分の参加するグループを選択して，各グループのメンバー表に登録することになった．

3. 第２日目：グループ討議

第2日目の朝，全体会議が招集され，グループ編成とグループ討論の課題が発表された．第1日目の夜に行われた各グループへの登録の結果，「組み込みソフトウェア」グループは1グループだけ編成することになった．これに対し，参加申し込みの多かった「ビジネス·ソフトウェア」グループを2グループ編成することになった．「その他のソフトウェア」グループには，システム·ソフトウェア，ツール，ソフトウェア教育などの経験者が参加することとなった．

グループ討議では，ウォーターフォール·モデルにもとづくプロセスの各工程でどのようなエラー·データを収集していて，どのように利用しているかを，各グループで議論することとなった．

以下に，各グループに配属された日本人のリーダの方々にお願いして作成していただいたグループ討議の記録を編集して掲載する．各グループの記録を作成したのは，松下通信の増子氏，東芝の山田氏，大阪大学の菊野教授，日本IBMの大場の4名である．

3.1 組み込みソフトウェア·グループ

この分科会への参加者は，電子交換機，データ端末機，POSシステムなどに組み込まれるソフトウェアの開発や品質管理を担当している技術者と管理者であった．

グループでは，「異なる組織で収集されたデータを一定の条件下で比較し，意味のある結果を導き出すことを可能にするために，どのような条件について標準化すべきかを明確化するか」を主として検討することとした．

この目的のため，最初に，各参加者が所属している組織で採用している開発プロセスの共通点と相違点を明確化する作業を実施した．プロセス·モデルについては，グループの参加者の所属している組織では，いずれもウォーターフォール型のモデルを採用していることが判明した．

次に，各組織で実施されているプロセスについて，具体的な作業の内容，適用されている手法，作業を担当している部門，成果物の内容，レビュー実施の有無，収集データの各点について，参加者が発表を行った．

討議の結果，工程の分割方法，工程の名称は，組織によってかなり違っていることか理解された．特にテストに関して，異なる組織で同一の名称をもつ作業があっても，その作業内容が違っている例があった．エラーや品質データの収集についても，その開始時期や内容が組織によってまちまちであった．

3.2 ビジネス·ソフトウェア·グループ1

この分科会への参加者は，各企業で大規模な企業情報システムの開発やその品質保証に関する技術支援を担当している技術者または管理者が主体であった．

グループ討議では，最初に各組織でデータ収集を開始する時点について議論した．多くの組織が，テスト工程からデータ収集を行っていたが，設計工程からデータ収集している組織もあった．次に，テストとしてどのようなテストが実施されているかを議論した．客先に納入してからの運用テストについても議論した．

ロンバッハ教授から，「欧米ではテストを協力会社などに発注するケースが多いが，日本ではどうか」との質問が出され議論した．ある組織では，「最近試験的にテストを協力会社に発注している」との例があった．その他の日本の組織では，テストを発注するケースはなかった．反対に，「日本では開発作業自体を発注するケースが非常に多いように聞こえるが，なぜそうなのか」との質問が出され議論した．代表的な意見は，「開発コストの大部分を占める人件費が，その部分を発注することによって低減できるから」とするものであった．

ロンバッハ教授の「いつから関連会社の技術者が参加するか」という質問については，多くの場合「仕様をまとめるかなり早いの段階から参加している」ことが判明した．「それでは，日本では独立系のソフトウェア·ハウスが不利で，育ちにくいのではないか」との質問に対して，「関連会社に発注すれば，その会社にノウハウが蓄積されやすいので，そうするほうが長い目でみて有利である」との意見が出された．

収集されているデータは，主として開発者が特定のシートに自己申告で記入したものを基礎としている．これに対してロンバッハ教授は，「開発者が自己申告したデータは，正確でないものがある」と指摘した．同教授が以前に実施した調査の結果では，「データの約半分が何らかの意味で正しくなかった」そうである．「開発者の作業記録を専門家が見て分類する方がよい」ということで意見が一致した．エラー·データの分類については，機能別のエラーの分類や，原因別のエ

ラーの分類は全ての組織で行われていた．これらのデータの収集や分析は手作業で行われている．データベースに蓄積している例は少ない．

3.3 ビジネス･ソフトウェア･グループ2

この分科会への参加者は，各企業で大規模な企業情報システムの開発やその品質保証を担当している技術者または管理者が主体であった．

グループ討議では，最初に各参加者がワークショップのために用意したアンケートへの回答を発表し，質疑応答を行った．ここでは特に，各組織で採用しているプロセスのモデル，レビューの実施時期と実施者，エラー･品質データ収集の時期と方法，データの検証方法と蓄積，独立した品質保証グループの有無などについて議論した．

この議論を通じて，同じウォーターフォール型のプロセス･モデルを採用していても，詳細については違っていること，用語の使い方や意味にかなりの違いがあることが判明した．そこで，それぞれの組織で実際に実施しているプロセスの各フェーズでの作業内容について議論し，「何が同じで，何が違うか」を確認することにした．しかし，会社やその国の文化･習慣の違いもあり，十分な理解は得られなかった．

また，本来同じプロセス･モデルにもとづいて作業を実施している同じ組織の中でも，実際のプロセスはプロジェクトごとに違っていることが確認された．これは，プロセスの各フェーズで実施すべき作業の内容についての定義が，組織内でも十分には定義されていないことが一因であると分析された．また，ユーザとのやり取り，会社の文化･風土，政府や公共機関などの慣習などによっても変わってくることが指摘された．

次に各組織で収集しているエラー･品質データの内容について議論した．ここでわかったことは，データの定義が一様でなく，それぞれの組織に合わせて用語の定義が微妙に違っていることであった．特に，「機能拡張に伴う変更」と「機能修正のための変更」をどう区別するのかが問題となった．

3.4 システム･ソフトウェアとソフトウェア教育グループ

この分科会への参加者は，システム･ソフトウェア，ソフトウェア開発ツールの開発やそのプロセスの標準化を担当している研究者や，大学や企業でソフトウェア開発の教育を担当している研究者であった．参加者のほとんどが，大学または企業の研究所に所属していたことが，このグループの特徴である．

このグループでは，各参加者の参画したプロジェクトがいろいろな意味で，まったく違った性格をもっていることが問題になった．開発されたソフトウェアの規模は，数百行の教育実習課題から百万行を超える大規模オペレーティング･システムのコンポーネントまで，さまざまであった．また，プロジェクトの規模でも，1人から数人で数週間のプロジェクトから1,000人以上のプロフェショナル技術者が1年以上参加しなければならないプロジェクトまで，さまざまなものがあった．

当然のことながら，各参加者の議論するプロセスは，かなり性格の異なるものであった．教育の実習課題では，かなり詳細な機能仕様と基本設計が作業者に与えられるので，作業は詳細設計工程からデバッグ工程までに限定される．言語処理系の開発でも，言語仕様は厳密に定義されているので，要求分析･定義，仕様定義の工程は実施されない．これに対して，ソフトウェア開発ツールなどでは，要求と仕様の定義にかなりの重点が置かれる．オペレーティング･システムでは，実現後のテストに多大な労力を必要としている．

以上のようなことから，グループではプロセスを議論することをやめ，ソフトウェア開発に必要な一連の作業を独自に定義することとした．すなわち，「要求の分析」，「要求の定義」，「要求仕様のレビュー」，「機能仕様の定義」，「機能仕様のレビュー」等などである．

このようにして定義した各作業について，各参加者は，「どの作業をプロジェクトで実施しているか」，「誰が実施しているか」，「どんなデータを収集しているか」，「収集したデータをどう活用しているか」などについて議論した．

4. 第3日目：まとめ

第3日目は朝から全体会議で，各グループによるグループ討論の後，今後の活動に関する議論が行われ，正午に解散した．

4.1 組み込みソフトウェア･グループ

モトローラ社のステングライン氏が，このグループによるグループ討論の総括を行った．

要求定義プロセスは，機能仕様，要求仕様，その他

の文書を自然言語で書くプロセスである．このプロセスの段階でエラー･データを収集している組織はただ１つだけであり，その組織においてもデータは活用されていない．

仕様化プロセスは，「どのようなソフトウェアを開発すべきか」を決定するプロセスである．この段階では，レビューがエラーや品質問題を発見する最も有効な手段である．いくつかの組織では，そのようなレビューで発見されたエラーや品質問題に関するデータを収集している．そのようなデータは，品質保証グループによって品質向上のために有効利用されている．

基本設計プロセスは，機能仕様を満足するソフトウェアの構造を設計するプロセスである．この段階の作業には，データ構造の定義なども含まれている．厳格なレビューこそが，この段階で混入するエラーを除去するための基本的な手段である．レビューによって発見されたエラーに関するデータは，品質保証グループによって収集･蓄積され，製品の品質保証や品質向上のために有効利用されている．

詳細設計プロセスは，基本設計ににている．ただし，このプロセスの出力には製品に関するより詳細な情報が含まれている．

コード化プロセスは，製品の詳細設計にもとづいてコードを書くプロセスである．この段階でも厳格なレビューは，エラーを除去するための最も基本的な手続きである．

テスト･プロセスは，単体テスト，機能テスト，統合テスト，システム･テスト，受け入れ検査，アルファ･テスト，ベータ･テスト，運用試験等の工程から構成されている．「誰が何をするか」や「どの工程で何をするのか」はそれぞれの組織で大きく違っている．これについて組織間の共通性はほとんどない．ただ，この段階になるとデータ収集は「組織的」に展開されるようになる．

保守プロセスは，ユーザの要求に合わせて開発済みのプログラムを変更するプロセスである．

4.2 ビジネス･ソフトウェア･グループ１

ドイツのカイザースラウテルン大学のロンバッハ教授が，このグループによるグループ討論の総括を行った．

討議された工程は，要求定義，仕様定義，基本設計，詳細設計，コード化，単体テスト，機能テスト，システム･テスト，フィールド･テスト，受け入れ検査，運用と保守であった．これらの工程の呼び名は，各組織によって違っていた．

一般的にいって，ソフトウェアの実現に関する作業は，日本の企業では協力会社が実施している．レビューはアメリカで実施されているものよりも「柔軟なもの」(英語の表現は「形式的でない」) である．また，品質保証を業務とする独立の組織をもっている会社は多くない．

日本の企業では，エラー･データの収集は単体テストの完了時から始まる例が多い．品質データは，一般的には日本では「エラー･データ」の意味しかもっていない．エラー･データ以外の品質データも収集している組織は，このグループには１つだけであった．その組織では，「エラー」や「欠陥」のかわりに「(品質) 問題」ということばを用いていた．その組織では，「コードまたは文書の変更に帰結するものをすべて"問題"と呼んでいる」とのことであった．

「エラー･データ」とは，日本の企業では，「エラーが発見された日時」と「エラーがユーザに与えた (または与えると予想される) 悪影響の度合い」を意味する場合が多い．数は少ないが，「機能のタイプ」，「欠陥のタイプ」，「ミスの混入理由」などについてもデータを収集している企業もあった．日本の企業のソフトウェア技術者は，「エラー (ミス)」と「欠陥 (バグ)」と「故障 (エラーの現象)」を区別していない．

エラー･データは，日本ではプロジェクトに参加している管理者や開発担当者，品質保証グループの担当者などによって広く有効利用されている．ただし，そのようなデータの利用については，各組織に決まった使い方があるのではなく，「個々人がそれぞれの工夫で」利用しているのが現実である．組織化されている訳ではない．データは，その利用に興味のある人には広く公開されているようである．

驚くべきことだが，日本ではエラーや品質問題に関する記録の収集･処理･保管は，ほとんど自動化されていない．記録は規定の帳票に記入され，別の帳票に転記され，帳票のまま机の中に保管されている．データは，その保管者に聞けば出てくるが，質問しないかぎり出て来ない仕掛けになっている．

4.3 ビジネス･ソフトウェア･グループ2

ベル研究所のボッタ博士が，このグループによるグループ討議を総括した．

抽象的なレベルでは，プロセス･モデルの組織間の違いはほとんどない．ところが，実際のプロセスはそれぞれの組織で大きく違っていた．また，参加者の間に大きな誤解があった問題の一つに「欠陥」の定義があった．

IEEE の標準的な「エラー」,「欠陥」,「故障」の定義はほとんど理解されていない．各組織がそれぞれ特有な「問題」の定義をしており，それらの定義は必ずしも同じものを意味しているものではなかった．

抽象的なプロセスは，要求，仕様，基本設計，詳細設計，コード化，単体テスト，機能テスト，システム･テスト，受け入れ検査，ベータ･テスト，運用，保守などの工程から構成される．

エラーや品質に関するデータは，プロセスのかなり早い時点から記録され収集されている．しかし，そのようなデータが集計され報告されるのは，公式なテストが開始された後のことである．データは，主として品質保証グループによって品質保証のために活用されている．

4.4 システム･ソフトウェアとソフトウェア教育グループ

オーストラリアのニュー･サウスウェールズ大学のジェフリー教授がこのグループによるグループ討論の総括を行った．

このグループでは，オペレーティング･システム，CASE ツール，言語処理システム，ソフトウェア教育などのプロジェクトにおけるソフトウェア開発を議論した．これらのプロジェクトで採用されているプロセスには，大きな差があった．そこでグループでは，一連の抽象的で一般性のあるソフトウェア開発作業を定義した．

その抽象的なプロセスは，要求の定義とレビュー，機能仕様の定義とレビュー，プログラムとデータ構造の定義とレビュー，モジュールの定義とレビュー，ソースコードの作成とレビュー，モジュール機能の確認，モジュールの統合，機能の確認，ユーザ環境でのシステムの品質評価，開発作業終了確認，プロセスの評価と検証，実環境での品質評価，システムの運用，システムの改善などの作業からなる．

エラー･データや品質データとしては，「レビュー時に発見されたエラー」,「テスト中に発見されたエラー」,「テスト中に観測された故障」,「テスト中に発見･修正された欠陥」,「ユーザによって発見された故障」などがある．

機能仕様のエラーとしては，要求仕様の誤解，矛盾した機能などがある．プログラムやデータ構造のエラーとしては，バランスの悪い構造，設計規約からの逸脱などがある．モジュールのエラーとしては，アルゴリズムの誤り，機能仕様の誤解などがある．コードのエラーとしては，不完全な実現 (初期設定の見逃し)，モジュール設計の誤解などがある．

4.5 総括とクロージング

バシリ教授は，各参加者が所属している企業や団体が，このワークショップのテーマに興味をもち，同様のワークショップに来年も参加することに積極的であるかどうかについて，参加者の意見を求めた．参加者の意見は，「来年のワークショップにも参加するであろう」とするものが多かった．

しかし，参加者の間から，「ソフトウェア･プロセスが組織によって異なっている現状では，同じようなワークショップを再度開催するまではなく，プロセスの違いを吸収できるようなフレームワークについて，まず参加者が合意したうえで，そのフレームワークにもとづいてデータの議論を展開すべである」とする複数の指摘が出された．

最後にバシリ教授は，バシリ教授と鳥居教授が今回のワークショップでの議論にもとづいて，アンケートの質問を改定し，各参加者宛に再度送付すること，そして各参加者が新しいアンケートの質問に対する回答を，再度バシリまたは鳥居教授に送付することを提案した．参加者はこの提案に賛成した．

「ソフトウェア･データとプロセスに関するワークショップ」参加者のための ソフトウェア品質データに関するアンケート

編集部

去る3月11～12の2日間，奈良先端科学技術大学院大学において「ソフトウェア･データとプロセスに関するワークショップ」が開催された．このワークショップでは，それぞれが持ち寄る定量的品質データについての議論を意味のあるものにするために，参加者全員に対して，以下のようなアンケートが行なわれた．

このアンケートは，「まえがき」に述べているように，それぞれの現場で収集されている品質データについて，その内容(意味)，背景，収集および利用方法などを理解するためのものである．いいかえれば，それは，データの背後にある開発プロセスの特性を理解する手掛かりと考えてもよいだろう．

現在，ソフトウェア･プロセスに関しては，さまざまな角度からの分析/研究が行われており，また，その品質面に関する定量的マネジメントの試みもあちこちでなされている．今回のワークショップは，そうしたさまざまな努力の成果をお互いに比較検討しうるような土台を作ることを目的とするものである．

ワークショップの成果は，また，別の機会にあらためて報告されるだろうが，とりあえず，アンケートの内容をここに公開して，ソフトウェアの品質問題に関心を持っておられる方々の参考に供したい．

アンケート

このアンケートは，あなたが所属または関係している組織においてこれまでに収集され，ワークショップで議論の対象とすることのできるソフトウェア品質データ（以下では単に「品質データ」と呼びます）について，その内容，意味，背景，収集方法，利用状況，などを理解する目的のものです．

アンケートの構成は，「品質データの利用目的」と「品質データの収集と分析の例」について質問の後，背景の理解を目的として，データを実際に収集した「組織のプロフィール」，対象となった「プロジェクトの特徴」，開発された「ソフトウェアの特性」，ソフトウェアの「開発プロセス」，「品質データの定義」，「品質データの収集」に関する質問からなっています．

背景に関する質問の中には答えにくいものがあるかもしれません．答えにくい質問に対しては，無理に回答する必要はなく，「答えられない」と回答していただいてかまいません．なお，補足説明が必要な回答については，それぞれのコメント欄に説明をお書きください．

回答内容は，本ワークショップにおける議論の場で参照されることはありますが，回答者の承諾なしに，ワークショップ以外の場において公開されることはありません．

---

1. 品質データの利用目的について

品質データの利用目的についてお答えください．

Q1-1: 収集した品質データを「ソフトウェアを受け取るべきか(検収)」または「ソフトウェアを納められるか（納品／出荷)」の判断の資料として利用していますか？

「はい」と答えた場合
- ･利用するのはどのようなデータですか？
- ･どのように利用していますか？

Q1-2: 収集した品質データをプロジェクトの現状把握や開発計画の変更といった問題解決のための資料として利用していますか？

「はい」と答えた場合
- ･利用するのはどのようなデータですか？
- ･どのように利用していますか？

Q1-3: 収集した品質データをプロジェクトの計画や開発コストの見積りなどの参考にしていますか？

「はい」と答えた場合
- ･参考にするのはどのようなデータですか？
- ･どのように利用していますか？

Q1-4: 収集した品質データを保守計画や保守コストの見積りなどの参考にしていますか？

「はい」と答えた場合
- ･参考にするのはどのようなデータですか？
- ･どのように利用していますか？

Q1-5: 収集した品質データをソフトウェア･ツール（デバッガ，テストツールなど）の研究や導入評価などのための基礎データとして利用していますか？

「はい」と答えた場合
- ･利用するのはどのようなデータですか？
- ･どのように利用していますか？

Q1-6: 収集した品質データを組織内でのソフトウェア教育･研修のための基礎資料として利用していますか？

「はい」と答えた場合
- ･利用するのはどのようなデータですか？
- ･どのように利用していますか？

Q1-7: 収集した品質データをその他の目的に利用していますか？

「はい」と答えた場合
- ･どのような目的に利用していますか？
- ･利用するのはどのようなデータですか？
- ･どのように利用していますか？

Q1-8: 品質データとして最も利用価値が高い，とあなたが判断されるものはどのようなデータですか？

·品質データ名
·利用目的
·利用方法

Q1-9: 品質データをどのように蓄積·保存していますか？次の中から該当するものを選んでください．複数ある場合には，それぞれの割合の概算値もお答えください．

[ ] 計算機データとして蓄積．(%)
[ ] 帳票など、紙で蓄積．(%)
[ ] その他の形式で蓄積 (詳細をお書きください)．(%)

Q1-10: 蓄積されている品質データ（収集されたままの生のデータ）は誰が利用しますか？（複数回答可）

[ ] 収集者本人．
[ ] プロジェクトの管理者．
[ ] プロジェクトの関係者．
[ ] 品質保証グループ．
[ ] 組織の関係者．
[ ] 会社（関係会社を含む）の社員．
[ ] 外部の人間（コンサルタント，大学等の研究者)．
[ ] その他（具体的にお書きください）

Q1-11: 品質データの分析結果は誰にフィードバックされていますか？（複数回答可）

[ ] 収集者本人．
[ ] プロジェクトの管理者．
[ ] プロジェクトの関係者．
[ ] 品質保証グループ．
[ ] 組織の関係者．
[ ] 会社（関係会社を含む）の社員．
[ ] 外部の人間（コンサルタント，大学等の研究者)．
[ ] その他（具体的にお書きください）

2. 品質データの収集と分析の例について

品質データの具体的な収集例と分析例についてお答えください．

Q2-1: 品質データを何らかの基準（エラーの混入工程，エラーの種類等）で分類し，分析していますか？

[ ] 答えられない（知らない．開示できない)．
[ ] 分類していない．
[ ] 分類しているが，どのような分類かは知らない．
[ ] 分類している．

「分類している」と答えた場合
(1) 分類方法
·分類の観点（例：エラーの混入工程）
·分類の基準（例：どの作業で混入したエラーか）
(2) 分析方法の例
(3) 分析結果の例
·分析結果
·分析結果の意味するもの
·分析結果の利用法
·データと分析結果の信頼性
·信頼性の根拠

Q2-2: 品質データに何らかの統計処理（開発工程別のエラー発見率の算出，各テスト作業での発見エラー数分布の算出等）を行い，標準的な品質管理法を実施していますか？

[ ] 答えられない（知らない．開示できない)．
[ ] 統計処理していない．
[ ] 統計処理しているが，どのような処理かは知らない．
[ ] 統計処理している．

「統計処理している」と答えた場合：
(1) 処理方法
·処理対象のデータ（例：各工程で発見されるエラー数）
·処理結果（例：単位規模当りのエラー数）
(2) 分析方法の例
(3) 分析結果の例
·分析結果
·分析結果の意味するもの
·分析結果の利用法
·データと分析結果の信頼性
·信頼性の根拠

3. 組織について

いままで議論してきた品質データ（これまでの回答のもととなった品質データ）は，あなたが所属または関係している会社のある組織（部門）で収集され分析されたものだと思います．データ収集源となった組織のプロフィールについてお答えください．

Q3-1: その組織の規模は？

[ ] 答えられない（知らない．開示できない)．
[ ] 約___名．

Q3-2: 直接ソフトウェアに関係している人の割合は？

[ ] 答えられない（知らない．開示できない)．
[ ] 約___%．

Q3-3: 地理的に分散していますか？

[ ] 答えられない（知らない．開示できない)．
[ ] _____箇所に分散．
[ ] 1ヶ所に集中．

Q3-4: 独立したテストグループを持っていますか？

[ ] 答えられない（知らない．開示できない)．
[ ] 持っている．
[ ] 持っていない．

Q3-5: 独立した品質保証グループを持っていますか？

[ ] 答えられない（知らない．開示できない)．
[ ] 持っている．
[ ] 持っていない．

4. プロジェクトについて

データ収集の対象となったプロジェクトの特徴についてお答えください．なお，複数のプロジェクトが対象となっている場合には，プロジェクトごとにお答えください．

Q4-1: 管理者と技術者の総数は？

[ ] 答えられない（知らない．開示できない)．
[ ] 約___名．

Q4-2: プロジェクトの契約形態はどのようなものでしたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 2社間契約による開発.
[ ] 複数企業による共同開発.
[ ] 社内情報システムの開発.
[ ] 市販ソフトウェアの開発.
[ ] その他（具体的に）

Q4-3: プロジェクト組織の構成･形態はどのようなものでしたか？次の中から該当するものを選んでください.

[ ] 答えられない（知らない．開示できない）.
[ ] 協力会社の管理者が参加（全体の ____%）.
[ ] 協力会社の技術者が参加（全体の ____%）.
[ ] 社内の管理者･技術者だけ.
[ ] 協力会社が開発したソフトウェアを組み込んだ
（全体の ____ %）.
[ ] その他（具体的にお書きください）

Q4-4: 開発期間はどれくらいでしたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 約____ヶ月.

Q4-5: 開発工数はどれくらいでしたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 約_____人月.

Q4-6: 再利用されたコードはありましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] いいえ.
[ ] はい．全体の約_____%.

「はい」と答えた場合
･再利用コードの約____% はコピー.
･再利用コードの約____% はツールで生成.

Q4-7: そのプロジェクトは新規プロジェクトでしたか，それとも，従来からのプロジェクトの延長でしたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 新規プロジェクト.
[ ] 従来プロジェクトの延長．約____% は既存のコード.

Q4-8: 開発チームは地理的に分散していましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] _____ つのサイトに分散していた.
[ ] 一ヶ所に集中していた.

Q4-9: 独立したテストグループがありましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] あった.
[ ] なかった.

Q4-10: 独立した品質保証グループがありましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] あった.
[ ] なかった.

5. ソフトウェアについて

データ収集の対象となったソフトウェアの特性についてお答えください．なお，複数のソフトウェアが対象となっている場合には，ソフトウェアごとにお答えください.

Q5-1: ソフトウェアの開発目的は何でしたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 自社内で業務に利用.
[ ] 自社商品に組込む.
[ ] 特定の顧客に納入.
[ ] パッケージを特定顧客用にカストマイズ.
[ ] パッケージの部品として納入.
[ ] パッケージ又は基本ソフトとして販売.
[ ] その他(詳細をお書きください)

Q5-2: そのソフトウェアはどのくらいインストールされ，使われていますか？

[ ] 答えられない（知らない．開示できない）.
[ ] インストール数は _____ である.

Q5-3: そのソフトウェアの種類は？

[ ] 答えられない（知らない．開示できない）.

＜処理形態＞
[ ] バッチ処理.
[ ] 対話型処理（オンライン処理／リアルタイム処理）.
[ ] 組込型のリアルタイム処理.
[ ] その他(詳細をお書きください)

＜処理内容＞
[ ] 事務系トランザクション処理用ソフトウェア.
[ ] 意思決定支援用ソフトウェア.
[ ] 生産管理･在庫管理システムなどのソフトウェア.
[ ] システム制御／機器制御用ソフトウェア.
[ ] その他(詳細をお書きください)

Q5-4: そのソフトウェアの革新性について，次の中から該当するものを選んでください.

[ ] 答えられない（知らない．開示できない）.
[ ] 研究用のプロトタイプ.
[ ] 新しいシステムのプロトタイプ.
[ ] 新しいシステムの最初のバージョン.
[ ] 既存のシステムの大幅な機能拡張.
[ ] 既存のシステムの改良バージョン.

Q5-5: 外的要因（ユーザの意向等）による開発中の仕様変更はどのくらいの頻度で発生しましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 仕様確定後の変更はなかった.
[ ] 仕様自体がなかなか確定しなかった.
[ ] 仕様は早くから確定していたが，変更は多かった.
[ ] その他(詳細をお書きください)

Q5-6: そのソフトウェアに対するメモリ，性能，品質面での制約はどのくらいの強さでしたか？

[ ] 答えられない（知らない．開示できない）.

メモリの制約
[ ] 厳しい.
[ ] 普通.
[ ] 緩い.

[ ] なし.

性能面の制約
[ ] 厳しい.
[ ] 普通.
[ ] 緩い.
[ ] なし.

品質面の制約
[ ] 厳しい.
[ ] 普通.
[ ] 緩い.
[ ] なし.

Q5-7: そのソフトウェアの品質特性として最も重視されたものは何ですか？

[ ] 答えられない（知らない．開示できない）.
[ ] 機能性 (Functionality) であった.
[ ] 信頼性 (Reliability) であった.
[ ] 使用性 (Usability) であった.
[ ] 効率性 (Efficiency) であった.
[ ] 保守性 (Maintainability) であった.
[ ] 移植性 (Portability) であった.
[ ] その他（名称とその定義をお書きください）

Q5-8: 最終製品として出荷または納入されたものは何ですか？(複数回答可)

[ ] 答えられない（知らない．開示できない）.
[ ] 機能仕様書.
[ ] 設計文書.
[ ] プログラム（ソースコード，オブジェクトコード）.
[ ] テストデータ，テスト結果，テスト記録.
[ ] レビュー報告書（レビュー記録）.
[ ] その他（具体的にお書きください）

6. プロセスについて

データ収集の対象となったプロジェクトで実施されたソフトウェア･プロセスについてお答えください.

Q6-1: プロジェクト計画の作成はあなたの所属する組織が参加して行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
･誰が作成しましたか？
･誰がレビューしましたか？
･どんな文書に記述されましたか？
･あなたはその文書を読みましたか？

Q6-2: 契約書の作成は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
･誰が作成しましたか？
･誰がレビューしましたか？
･どの程度の詳細さで記述されましたか？

Q6-3: 要求の定義は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
･誰が定義しましたか？
･誰がレビューしましたか？
･どんな文書に記述されましたか？

Q6-4: 機能の定義は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
･誰が定義しましたか？
･誰がレビューしましたか？
･どんな文書に記述されましたか？

Q6-5: ソフトウェア構造（モジュールとデータの構造）の設計は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
･誰が設計しましたか？
･誰がレビューしましたか？
･どんな文書に記述されましたか？

Q6-6: モジュール（処理の手順）の設計は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
･誰が設計しましたか？
･誰がレビューしましたか？
･どんな文書に記述されましたか？

Q6-7: コードの作成は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
･誰が作成しましたか？
･誰がレビューしましたか？

Q6-8: モジュール単位での実現の確認テストは行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
･誰がテストを設計しましたか？
･誰がテストを実施しましたか？
･誰が結果をレビューしましたか？

Q6-9: モジュールを統合し，ソフトウェアを構成した後の機能の確認テストは行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
·誰がテストを設計しましたか？
·誰がテストを実施しましたか？
·誰が結果をレビューしましたか？

Q6-10: 開発されたソフトウェアの品質をユーザの視点から評価するためのテストは行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
·誰がテストを設計しましたか？
·誰がテストを実施しましたか？
·誰が品質を評価しましたか？

Q6-11: 開発作業の完了を確認するための検査は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
·誰が検査を計画しましたか？
·誰が検査を実施しましたか？
·誰が開発完了を承認しましたか？

Q6-12: ソフトウェアの完成度を確認するためのユーザによるテストは行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
·誰がテストを計画しましたか？
·誰がテストを実施しましたか？
·誰が結果をレビューしましたか？

Q6-13: ソフトウェアの運用に先立って，導入後一定期間，実際の稼働環境での試験（試運転）をする場合があります．そのような試験は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
·誰が試験を計画しましたか？
·誰が試験を実施しましたか？
·どの程度の期間の試験でしたか？
·誰が結果をレビューしましたか？

Q6-14: ソフトウェアの検収（受け入れ）は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
·誰が検収項目を設定しましたか？
·誰が検収を実施しましたか？
·誰が結果をレビューしましたか？

Q6-15: ソフトウェアの運用は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
·誰が運用していますか？
·誰が障害の分析をしていますか？

Q6-16: ソフトウェアの保守は行われましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
·誰が修正をしますか？
·誰が修正の確認をしますか？
·誰が追加機能を定義しますか？
·誰が追加機能を設計しますか？
·誰が追加機能を実現しますか？
·誰が追加機能をテストしますか？
·誰が品質保証をしますか？

Q6-17: 他にも作業（組み込みソフトウェアの品質管理等）は行れましたか？

[ ] 答えられない（知らない．開示できない）.
[ ] 行われなかった.
[ ] 行われた.

「行われた」と答えた場合
·作業名
·作業内容
·作業実施者
·作業結果のレビュー方法

7. 品質データの定義について

品質データの定義（内容）についてお答えください．

Q7-1: 品質データとして，開発工程で発生した「人為的なミスや誤解（一般的には「エラー」と言います）」に関するデータを収集していますか？（産業界でよく使われている「バグ」は，「エラー」を意味すると解釈される場合が多いと言われています.）

[ ] 答えられない（知らない．開示できない）.
[ ] 収集していない.
[ ] 収集している.

「収集している」と答えた場合
·データ収集を行う工程名
·データの名称と定義（内容）
·計測の単位

Q7-2: 品質データとして，開発工程で発見され修正された「記述の誤り（一般的には「欠陥」と言い，プログラムの誤りを含みます）」に関するデータを収集していますか？（産業界でよ

く使われている「修正箇所」は,「欠陥」を意味すると解釈される場合が多いと言われています.)

[ ] 答えられない（知らない．開示できない).
[ ] 収集していない.
[ ] 収集している.

「収集している」と答えた場合
·データ収集を行う工程名
·データの名称と定義（内容）
·計測の単位

Q7-3: 品質データとして，テストや運用段階で観測され報告された「不具合（一般的には「故障（現象)」と言います)」に関するデータを収集していますか？（産業界でよく使われている「クレーム」は,「故障」を意味すると解釈される場合が多いと言われています.)

[ ] 答えられない（知らない．開示できない).
[ ] 収集していない.
[ ] 収集している.

「収集している」と答えた場合
·データ収集を行う工程名
·データの名称と定義（内容）
·計測の単位

Q7-4: ソフトウェア品質データとして,「エラー」,「欠陥」,「故障」以外の範疇のデータを収集していますか？（例えば，ソフトウェアの利用状況などは，間接的に「品質の良さ」を示すデータになります.)

[ ] 答えられない（知らない．開示できない).
[ ] 収集していない.
[ ] 収集している.

「収集している」と答えた場合
·データ収集を行う工程名
·データの名称と定義（内容）
·計測の単位

Q7-5:「エラー」,「欠陥」,「故障」などのデータを２つ以上の工程で収集し，集計（総数の算出）を行っていますか？

[ ] 答えられない（知らない．開示できない).
[ ] 工程別に集計し，工程にまたがっての集計はしない.
[ ] 集計している.

「集計している」と答えた場合
·集計を行う工程名
·集計データ（総数）の名称と定義（内容）
·計算方法（総数の算出方法）
·集計データ（総数）の単位

8. 品質データの収集について

品質データの具体的な収集方法についてお答えください.

Q8-1:「エラー」に関するデータの収集方法等をデータ収集を行う工程ごとにお答えください.

[ ] 答えられない（知らない．開示できない).
[ ]「エラー」に関するデータは収集していない.

·データ収集を行う工程名
·収集データ名
·収集方法
·収集者

Q8-2:「欠陥」に関するデータの収集方法等をデータ収集を行う工程ごとにお答えください.

[ ] 答えられない（知らない．開示できない).
[ ]「欠陥」に関するデータは収集していない.

·データ収集を行う工程名
·収集データ名
·収集方法
·収集者

Q8-3:「故障」に関するデータの収集方法等をデータ収集を行う工程ごとにお答えください.

[ ] 答えられない（知らない．開示できない).
[ ]「故障」に関するデータは収集していない.

·データ収集を行う工程名
·収集データ名
·収集方法
·収集者

Q8-4:「エラー」,「欠陥」,「故障」以外の範疇のデータの収集方法等をデータ収集を行う工程ごとにお答えください.

[ ] 答えられない（知らない．開示できない).
[ ] そのようなデータは収集していない.

·データ収集を行う工程名
·収集データ名
·収集方法
·収集者

Q8-5: ソフトウェア品質データの収集は，ソフトウェアの出荷または納入後，どのくらいの期間にわたって行われますか？

[ ] 答えられない（知らない．開示できない).
[ ] 出荷または納入後は収集していない.
[ ] 出荷または納入後_____ヶ月間は収集している.
[ ] ソフトウェアが使用されている期間中は収集している.

これで終りです．御協力ありがとうございました.

今回のワークショップは，大場さんがレポートされた昨年秋の IWSED の経験にもとづいて，ソフトウェア·マネジメント (特に品質管理) のための各種データとそれらのデータが収集されたソフトウェア·プロセスとの意味論的なつながりを探ることを目的としたものであった．この秋 (11 月) には，今回のワークショップでの討論を踏まえて，第２回目の IWSED 開催が予定されている.

SEA Seminar & Forum (March 1994)

ソフトウェア・プロセス・アセスメントの国際規格
SPICE の動向

去る 3 月 24 日 (月) の午後に，青年会議所会館 (東京・平河町) で行われた表記のセミナー &Forum のレポートです．

最近，ISO 9000 をめぐる話題がジャーナリズムを賑わしています．SEA Forum でも，すでに昨年 10 月に，その具体的内容や認証手続きに関するチュトリアルを実施しました．また，よりソフトウェア工学分野に近いトピックとしては，CMU/SEI から提案されたソフトウェア・プロセス成熟度モデル (または CMM) があります．SEA では，数年前，モデルの提案者であるワッツ.ハンフリー氏が来日された際，プロセス成熟度の実態調査に協力し，その結果は氏の著書 (邦訳あり) にも紹介されています．

ところで，今回のフォーラムで取り上げられた SPICE は，Software Process Improvement and Capability dEtermination というそのフルネームが示す通り，ソフトウェア・プロセス・マネジメントに関する新しい国際標準規格案であり，近い将来，われわれの仕事に多くの影響を与えるものと予想されるのですが，まだ，その内容を御存じない方が多いでしょう．

ISO 9000 は，国際規格とはいいながらも，欧州 (英国) の主導で，しかも当初ハードウェア製造業での品質管理を念頭に置いて作られたために，そのソフトウェアへの適用については，いまだに多くの国際的な反発があります．また SEI の成熟度モデルは，本来米国の DoD 関連の業者査定が主目的であったため，米国以外での関心はあまり高いとはいえません．

それに対して，SPICE は，現在，世界中のソフトウェア・エキスパートたちが協力して作っている国際規格です．スパイスをふんだんに使った料理は，ピリッと辛いが，それだけ深い味わいを含んでいます．SPICE の目的は，それと同じように，ソフトウェア開発組織のプロセス成熟度の改善を通して，より高い品質を追及することであり，近い将来，SPICE にもとづくソフトウェア組織の認定・登録が行われるようになるものと，ほぼ確実に予想されます．それは，国際的にも，また国内的にも，ソフトウェア・ビジネスに大きな規制力を持つようになることでしょう．

今回の Forum では，ISO/IEC や JTC1/SC7/WG10 で実際に SPICE 規格の制定に努力されている方々を講師に迎えて，解説セミナーおよびパネル討論という形で行われました．

プログラムは以下の通り：

(1) セミナー (13:30 ～ 15:00)
- SPICE の背景と概要
  — 松原友夫 (Office Peopleware)
- Baseline Practice Guide (BPG)
  — 堀田勝美 (NTT)
- Process Assessment Guide (PAG)
  — 青山和之 (日立製作所)

(2) パネル討論 (15:30 ～ 17:00)
  司会： 熊谷章 (PFU)
  パネリスト： 上記のスピーカ各氏
  コメンテータ： 岸田孝一 (SRA)

参加者は約 40 人でした．以下に，当日スピーカの方々が使われた OHP の内容を要約します (文責：編集部).

1. 背景

社会的背景
- システムの巨大化・広域化
- クリティカル・システムの増加
- 開発組織の技術格差の拡大
- ソフトウェア産業の相互依存性
- 各産業のソフトウェア依存度の増加

先行的な組織アセスメント
- SEI のプロセス成熟度モデル
- ISO-9000 による認定

技術的背景
- プロダクト改良からプロセス改善へ

－さらに組織としてのプロセス成熟へ
－メトリクスを用いた評価手法の発展
－開発方法論の発展

管理的背景
－各種管理手法の普及
－欧米は構成管理重視
－日本は品質管理重視

2. これまでの経緯

国際標準化機構 (ISO) における SPICE 開発の経緯は，およそ次の通りである．

1991 年夏にイギリスから，ソフトウェア･プロセス･アセスメント標準化の必要性についての調査が提案され，翌年に調査終了．その結果，新しい作業アイテムとしての提案･投票が行われ，1993 年初めに承認．作業グループ WG10 がスタートした．

WG10 には，米･英･日･豪･伊をはじめとして，主だった国はほとんど参加している．国際ミーティングは，1993/1(Dublin)，1993/9(Rome)，1993/11(Brisbane)，1994/2(Helsinki) とほぼ四半期に一度ずつのペースで開かれ，審議が進んでいる．今後の予定は，1994/6(Ottawa)，1994/9(Pittsburgh)，1994/12(Hong-Kong)，1995/3(場所未定)．

また，構成管理下に置かれたマスタ･ファイル (@ カナダ) とバックアップ･ファイル (@ オーストラリア) を介し，国際インターネットを頻繁に利用して，規格作成のための具体的な作業が継続的に行われている．

3. SPICE の概要

[狙い]: SPICE プロジェクトの狙いは，ソフトウェア･プロセス･アセスメントの手法と応用をカバーする国際規格を作成することである．規格案は，当面,"タイプ 2 の技術報告書"として発行される予定になっている．"タイプ 2"とは，取り上げられたテーマがまだ技術的に発展途上にあるなどの理由で，いますぐに国際規格にはできないが，将来そうした合意が得られる可能性がある場合に出されるものである．

[目的]: SPICE 規格の目的は，プロセスの改善を通じてソフトウェアの品質を改善するために各組織 (ソフトウェア開発者/購入者/利用者) が用いるツールを用意することである．

[適用範囲]: この規格はソフトウェア･プロセスのための規格であって，プロダクトの規格ではない．ソフトウェア支援およびソフトウェア依存システムの購入･開発･納入･運用･保守･関連サービスのすべてに適用される．この規格は，ソフトウェア開発組織におけるプロセスの改善に利用できなければならないし，また，２者間契約によってソフトウェア (および関連サービス) を購入する組織が供給者の実施能力レベルを評価するさいにも利用できなければならない．

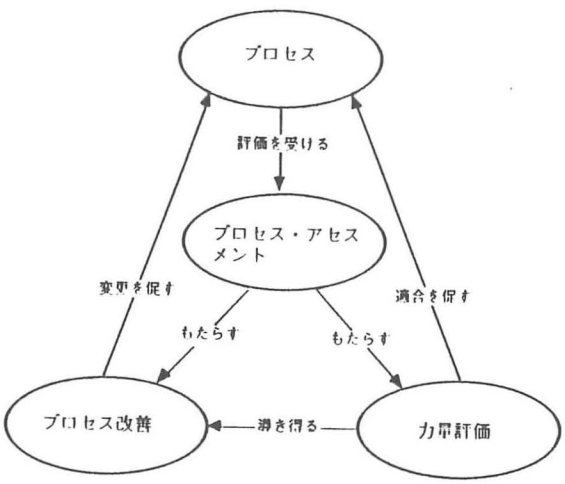


図１　プロセス･アセスメントの位置づけ

[規格への要求]: 機能的には，どんなプロジェクト/作業分野でも正しく評価できること，さまざまな応用ドメイン/ビジネスニーズ/組織規模に対して適用できること，各プロセスに対して適用可能なベースライン･プラクティスを規定すること，評価者が備えているべき資格を規定すること，などである．また，非機能的な要求としては，単純でわかりやすいこと，文化とは無関係であること，特定の組織構造/経営理念/ライフサイクルモデル/ソフトウェア技術や開発手法を前提とはしないこと，などが求められている．

4. 効果

SPICE は，(1) ソフトウェア･プロセスを規定し，計測し，管理し，継続的に改善することが可能であり，また，(2) プロセスの改善が作り出される製品の改善に決定的な役割を果たすものである，という２つの前提に立って，プロセス･アセスメントの確立を目指している．それによってもたらされる期待効果として

は，次のようなものがある：

－ それぞれの組織に対して，プロセスの全体的な状況を把握するための手法を提供する．
－ プロセス･アセスメントを実施することにより，継続的な改善の文化を育てる．
－ 開発組織は，そうしたプロセスの改善によって，コストの削減や品質の向上，顧客ニーズへの素早い対応が可能になる．
－ ソフトウェア購入者にとっては，プロセス･アセスメントによって，製品またはサービス提供者の能力をより正確に把握することができ，契約にさいしての不確定性が減少する．
－ 結果として，世界全体におけるソフトウェア(製品およびサービス)の品質向上がもたらされる．

5. 規格の体系

SPICE 規格の体系は次の通りである：

[IG] Introductory Guide: SPICE 導入ガイド．
[BPG] Baseline Practice Guide: アセスメントの基準として，望まれるソフトウェア活動を規定する．
[PAG] Process Assessment Guide: アセスメントの実施，結果のまとめ方などのガイド．
[AI] Assessment Instrument: アセスメントを行い，データを抽出するために必要なツールの要件．
[PIG] Process Improvement Guide: アセスメントの結果を利用して，プロセスを改善する方法のガイド．
[PCDG] Process Capability Determination Guide: アセスメント結果を利用して，開発組織の能力評価を行なう方法のガイド．
[ATQG] Assessor Training & Qualification Guide: アセッサ育成用の訓練プログラムの開発，第三者アセッサ資格のためのガイド．

6. BPG について

SPICE では，ソフトウェアの開発，運用，保守，管理，サービス，etc に関わる活動を，図 2 に示すように，２つの方向(次元)から捉えている．

１つは,"プロセス･エリアープロセスーベース･プラクティス" という階層的フレームワークである．

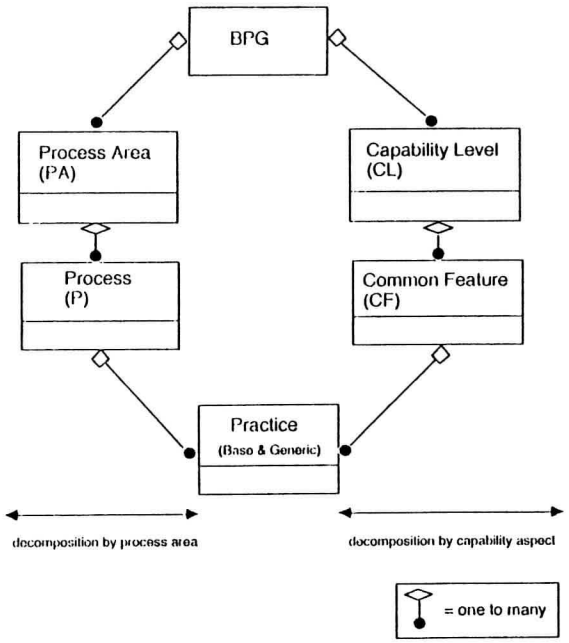


図 2 SPICE のフレームワーク

ここで,"プロセス･エリア (Process Area)" とは，同じカテゴリに属するプロセスの集合を指す．次の 5 種類のプロセス･エリアが考えられている：

顧客-供給者 (Customer-Supplier)
エンジニアリング (Engineering)
プロジェクト管理 (Projest)
サポート (Support)
組織基盤構築 (Organization)

"プロセス (Process)" とは，同じ目的を持つ活動 (Practice) の集合を指す．各エリアごとに，全体で 35 種類のプロセスが定義されている：

1.0 Customer-Supplier Process Area
  1.1 Perform Audits and Reviews
  1.2 Package, deliver, and Istall Software
  1.3 Support Operation of Software
  1.4 Provide Customer Support
  1.5 Assess Customer Satisfaction
  1.6 Manage Customer Requirements and Requests
2.0 Engineering Process Area
  2.1 Develop Systems Requirements and Design
  2.2 Develop Software Requirements
  2.3 Develop Software Design

2.4 Implement Software Design
2.5 Integrate ans Test Software
2.6 Integrate ans Test System
2.7 Maintain System and Software
2.8 Develop User Documentation
3.0 Projest Process Area
3.1 Establish Contact
3.2 Plan Project Life Cycle
3.3 Establish Project Plan
3.4 Build Teams
3.5 Coordinate Teams
3.6 Manage Requirements
3.7 Manage Quality
3.8 Manage Project resources
3.9 Manage Subcontractors
4.0 Support Process Area
4.1 Develop Documentation
4.2 Configuration Management
4.3 Quality Assurance
4.4 Peer Reviews
4.5 Problem Resolution
5.0 Organization Process Area
5.1 Engineer the Business
5.2 Define the Process
5.3 Training
5.4 Improve the Process
5.5 Enable Reuse
5.6 Provide Development Environment
5.7 Provide Work Facilities

"ベース・プラクティス (Base Practice)"とは，特定のプロセスの目的に対応したエンジニアリングまたは管理の活動を指す．全体で 178 種類のベース・プラクティスが定義されている．例をあげれば,"2.3 ソフトウェア設計" のプロセスにおけるベース・プラクティスは：

B2.3.1 再利用戦略の決定
B2.3.2 トップレベルの設計
B2.3.3 トップレベルのインタフェイス設計
B2.3.4 詳細設計

といった具合いである．

もう 1 つの次元は,"能力レベルーコモン・フィーチャージェネリック・プラクティス" という階層的なフレームワークになっている．

"能力レベル (Capability Level)" とは，プロセスを形成する能力を段階的に高めるコモン・フィーチャの集合を意味する．次の 6 段階のレベルが定義されている (CMM では 5 段階であった)：

0. Initial
1. Performed
2. Managed
3. Defined
4. Measured
5. Optimizing

"コモン・フィーチャ (Common Feature)"とは，プロセスの実行上，同じ様相を示す活動 (Practice) の集合を指す．それぞれの能力レベルごとに，全体で 13 種類のコモン・フィーチャが定義されている：

0. Initial
(なし)
1.0 Performed
1.1 Base Practices Are Performed
2.0 Managed
2.1 Committing to Perform
2.2 Planning Performance
2.3 Disciplined Performance
2.4 Tracking and Verifying Performance
3.0 Defined
3.1 Defining a Standard Process
3.2 Tailoring the Standard Process
3.3 Using Data
4.0 Measured
4.1 Establishing Measurable Quality Goals
4.2 Determining Process capability to Acieve Goals
4.3 Objectively Managing Performance
5.0 Optimizing
5.1 Establishing Quantitative Process Effectiveness
5.2 Improving Process Effectiveness

"ジェネリック・プラクティス (Generic Practice)" とは，プロセスを形成する能力を高めるような改善を行なうための活動である．全体で 47 種類のジェネリック・プラクティスが定義されている．たとえば，"2.3 Disciplined Performance" におけるジェネリック・プラクティスは：

G3.3.1 文書化された標準/要領の利用
G3.3.2 構成管理または版管理による生産物管理
G3.3.3 生産物に対するレビューの実施と是正措置
G3.3.4 プロセスの実施に関するデータの記録
G3.3.5 プロセス実施記録の活用

といった具合いである.

実際のアセスメントにさいしては, この2次元のフレームワークをまとめた図3のようなマトリクスが用いられる.

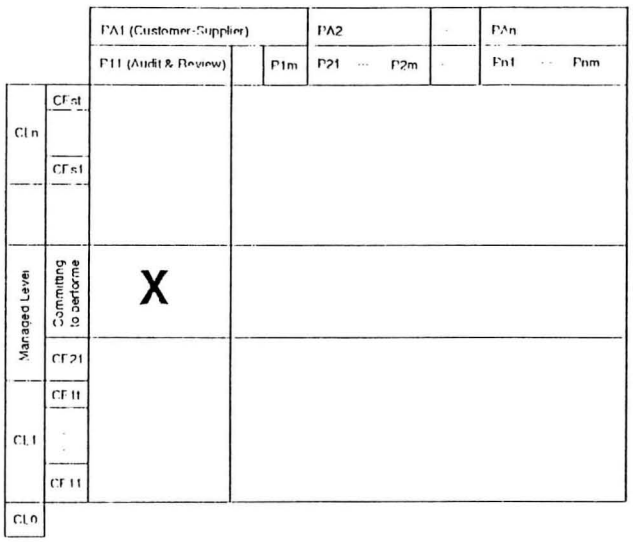


図3 評価のマトリクス

この例では,"P1.1 Audit and Review" という特定の活動のアセスメントにおいて, P1.1 に属するベーシック・プラクティスが実施されているかどうかを評価し, "CF2.1 Committed to Perform" に属するジェネリック・プラクティスで, 個々のプロセスを評価する.

6. PAG について

SPICE/PAG の目的は, どのような環境にも適合するプロセス評価の指針を提供することである. プロセスを改善したい, あるいは,「特定の」要求または契約に対してプロセスが適切かどうか評価したい, と考えている組織が, ガイダンスの対象として考えられている. すなわち, PAG は ISO-9000 などと違って, 一般的な評価基準を与えるものではない.

PAG の特徴は, 自己評価を推奨していること, ビジネス・ゴールというものを考慮していること, BPG への一致度 (いわゆる Conformance) と, プロセスの有効性 (Effectiveness) を分離して考えていること, また, 環境に依存しない評価ガイドを狙っていること, であろう.

PAG の構成は次のようになっている:

1. Introduction
2. Overview of Process Assessment
   評価のフェイズ, 方法, 成功因子
3. Guidance of Assessment
   評価の流れ
4. Rules for Assessment
   評価の入出力と評点基準

プロセスを評価する目的は, 現行のプロセスを2つの視点から理解することであると考えられる. 第1は, それがどの程度 BPG と一致しているか (Conformance の視点), そして第2は, 与えられた特定のビジネス・ゴールを達成する上でそれがどの程度有効であるか (Effectiveness の視点) である.

プロセス評価には, いくつかのアプローチがある:

自己評価:
   評価チーム自身が評価対象に含まれる.
内部評価:
   評価チームが評価対象と同じ組織に属する.
外部評価:
   評価チームは評価対象と異なる組織に属する.

プロセス評価の成功因子は次のようなものである:
Cimmittment:
   評価に必要な資源の確保, etc.
Motivation:
   建設的に評価を進める, etc.
Confidentiality:
   入力情報が信用できること, etc.
Relevance:
   組織にとって有意義な評価を目指す, etc.
Credibility:
   評価チームが信用されること, etc.

プロセス評価のフェイズ, および作業の流れは次のようになる:

プロセス評価の準備
   ー アセッサの選出

- 目的，範囲，制約の審査
- アプローチの選択
- チームの選出と準備
- 評価計画の立案
- 評価対象組織ユニットの準備
- 情報提供者との打ち合わせ
- ドキュメントや記録の収集

情報の収集･分析

- 情報の収集
- 情報の分類
- 情報の検証
- スケールの選定
- 評点の導出
- 評点の検証

評価結果の提示

8. 今後のスケジュール

規格文書の作成

- PAG，BPG は 2 月でほぼ完了.
- PIG，PCDG，ATQG は 6 月で完了予定.
- IG，AI は 9 月で完了予定.

アセスメント実験

- すでに実施中.

試行

- 今年後半から開始予定 (日本も参加).

アセッサーのトレーニング

- 今年 8 月の第 1 回を開催予定.

広報活動

- 国際会議，ニュースレターなど計画中.

9. まとめ

以上の説明から分かるように，SPICE は，これまでソフトウェア･プロセスに関して提案されてきたさまざまなモデルや基準 (CMM, ISO 9000, Trillium, Bootstrap, SLCP, STD ImproveIT, etc.) を参照しながら，それらのほとんどを包含する形で作られつつある国際規格である．これまでのものと違って，その策定にあたっては，世界中のソフトウェア工学のエキスパートたちが熱心に協力して，作業が (急ピッチで) 進められつつある．日本のソフトウェア･コミュニティも，INSTAC のプロセス･アセスメント委員会を中心に参画してきており，今後の試行その他のフェイズで，より積極的な貢献が世界から期待されている．

リポジトリ標準 PCTE とそのまわりの最新状況

塩谷 和範
(SRA)

知っている人しか知らない PCTE とその回りの最新状況を，簡単にまとめてみました．どうやら，PCTE が，今年中にも新たなソフトウェアリポジトリの国際標準となることは確実のようです．近々，JIS 化の作業も始まることでしょう．

1. 歴史

PCTE (Portable Common Tool Environment) は，EU(欧州連合) の ESPRIT 計画の一環として，ESPRIT で開発された各種の CASE ツールおよび支援環境の相互利用のための共通プラットホームを実現することを目指して，1983 年に仕様化の研究が開始されました．このアイデアのもとになったのは，Ada 環境のアーキテクチャ APSE (Ada Programming Support Environment) で，APSE は Ada という特定言語環境の支援を目的としているのに対して，ESPRIT 計画では，プログラミング言語を特定しない，ソフトウェア開発プロジェクトのライフサイクル全域に渡る総合的な支援を目的とする統合環境 "IPSE" (Integrated Project Support Environment) の構築を目指して，仕様の開発とプロトタイピング実験が行なわれました．

このプロトタイピングと実用化検証を通じて，PCTE 仕様が実用に耐えることが確認されたことを受けて，PCTE 仕様を管理する組織 PIMB (PCTE Interface Management Board) が ECMA (欧州電子計算機工業会) の傘下に組織され，PCTE 1.5 を ECMA 標準として決定しました．これをもとにした製品 Emeraude PCTE V12 が，1994 年 3 月現在では唯一の市販製品となっています．その後，NATO 向けに強化された PCTE+ の仕様をもとに，ECMA PCTE の仕様が作成されました．

1994 年 3 月現在 ISO での ECMA PCTE 仕様に対する Fast Track DIS 郵便投票は終わり，投票結果確認がおこなわれています．投票状況は欧州·米国とも肯定投票であると思われますので，本年中には国際標準なることは確実のようです．ECMA PCTE 対応製品は，IBM カナダや Emeraude を含む 4 社からこの春にも発売を開始するというアナウンスがなされています．

2. PCTE とは?

上で説明したように，PCTE は，ソフトウェア開発プロジェクトのための統合環境 (IPSE) の構築を目指すものです．このために，固有環境である "マシン +OS" を，PCTE インタフェースという移植可能な共通プラットホームで覆ってしまい，CASE 環境はこのプラットホーム上に構築されます．いわば，仮想 PCTE マシンを定義するのです．この仮想 PCTE マシンは，異なる機種上に

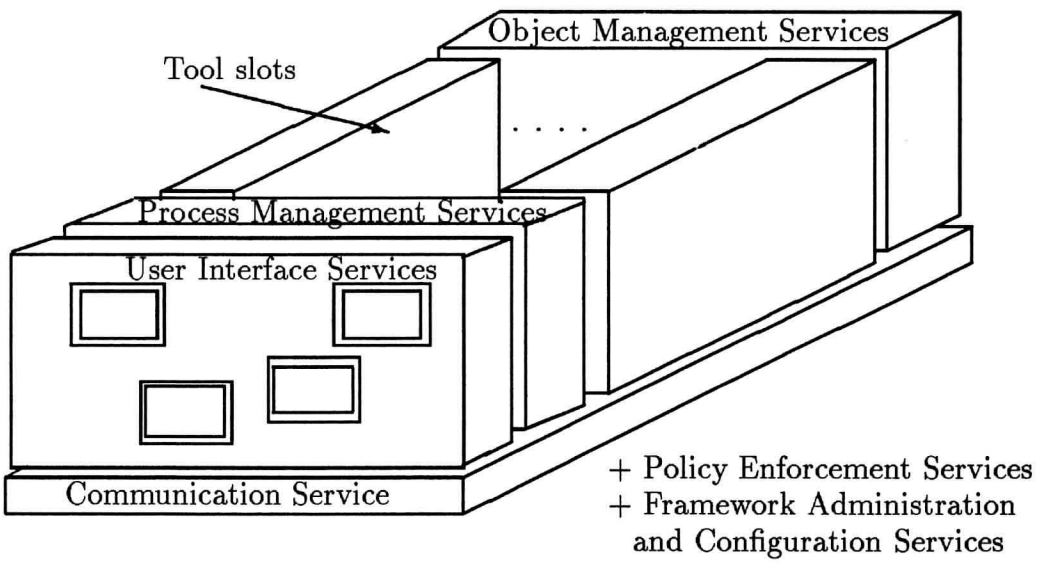


Figure 1: Reference model organization.

移植可能ですから，この上で動くアプリケーションも基本的にはコンパイルのみで移植できることになります．

PCTE の概念モデルとしては，中心に "マシン +OS" を置いた IPSE モデルと，HP 社の Softbench で有名になったトースタ･モデル (正式には「参照モデル (ECMA-NIST Reference Model)」) の 2 つがあり，参照モデルは，現在ではソフトウェア統合環境モデルとして定着しています．

参照モデルは，ユーザ･インタフェース･サービス，プロセス管理サービス，オブジェクト管理サービス，コミュニケーション･サービスの 4 つの共通サービスと，それらを利用するプログラムのためのツール･スロットの 5 つからなります．この内，ユーザ･インタフェース･サービスは，大勢を支配している X-Windows/Motif の使用を推奨しています (もともと計画されていた PCTE 独自仕様は破棄されました)．

コミュニケーション･サービスは，PCTE 上の各サービスおよびアプリケーション間通信のためのサービスで，現在の仕様では Posix 類似の基本的な機能のみを定めています．実際の各社の PCTE の実装上では，PCTE 基本機能に加えて BMS / ToolTalk / CORBA の 3 種類の実装がなされ，特に非 PCTE 環境アプリケーションとの統合･連係のために利用されます．将来的には，CORBA に対応しつつ CMT (COSE Messaging Technology: ToolTalk がベース) を利用する方向になって行くのではないかと思われます．

プロセス管理サービスについては，いまだに有力な技術が出てきていません．現在の PCTE 仕様の上では，やはり Posix 類似の基本的な機能のみを定めています．PCTE 自体が分散環境を想定して仕様化されていますので，分散環境上のプロセス実行が可能です．

ツール･スロットとは，PCTE の各共通サービスが提供する API のツール側の切り口をイメージしたものです．したがって，このための具体的な機能があるわけではなく，PCTE インタフェースに従うツール (プログラム) であれば，"ツールスロット" に収まり機能することができます．しかし，まだ PCTE 上で動く商用プログラムが少ないため，あるいは他環境上のアプリケーションを PCTE 上で利用するために，非 PCTE プログラムを PCTE 上で動かすための支援機能が各社の PCTE 上に用意されています．その代表が HP 社の Encapsulator (IBM/Bull PCTE) です．

2.1. リポジトリ

オブジェクト管理サービスあるいは「リポジトリ」サービスが，PCTE 仕様の真髄であるといえます．オブジェクト管理サービス (OMS: Object Management Service) は，ソフトウェア開発に関連するすべての物を対象とする「保管庫」に対するアクセスサービスを提供します．PCTE でのオブジェクトとは，オブジェクト指向のオブジェクトと違い，method 抜きの型定義されたデータおよび属性からなるオブジェクトです．これにオブジェクト間の関係を含めたオブジェクトネットワークがリポジトリの内容です．今後の説明中の"オブジェクト" は，PCTE オブジェクトを意味します．

このように，リポジトリ内には各種オブジェクトと，それらの関係を指定するリンクあるいはリレーション (双方向リンクの対を，PCTE ではリレーションと呼ぶ) と，オブジェクト，リンク/リレーションが持つ属性が記録･格納されます．このような情報のネットワーク構造を規定したものをスキーマと呼び，拡張 ERA (Entity Relationship Attribute) モデルで記述します．

2.2. PCTE スキーマの利点

(1) 作業スキーマ

PCTE におけるスキーマ (リポジトリ)･システムの優れた点は，第 1 に，スキーマを複数指定することによって，視点 (単一スキーマの世界) を複合化できることです．これを作業スキーマと呼びます．1 つのスキーマは，あるデータ･モデルを規定します．PCTE プログラムは，このデータ･モデルにより，リポジトリ上のオブジェクト世界にアクセスするわけです．当然，このスキーマ中に定義されたオブジェクト間の関係しか見えません．これは，リポジトリにアクセスする利用者に対しても同じことがいえます．

もし，あるスキーマ中にないオブジェクトが必要な時には，そのオブジェクトが定義されているスキーマを追加指定して，複数のスキーマ定義の和を作ることができます．このスキーマの和集合が作業スキーマです．また，スキーマの追加削除は動的に行なえます．したがって，必要が生じる都度，必要なスキーマを指定し，視点を変更するといったことが動的に行なえます．

(2) 動的スキーマ変更

第 2 に，スキーマの定義は，DDL (Data Definition Language) によって行ないます．各 PCTE ベンダからは，グラフィカルなスキーマ･エディタが用意されています．先に説明したように，スキーマの重ねあわせが可能ですので，スキーマ定義時には，あるデータ･モデル単位，ある機能グループ単位など，モジュール化してスキーマ定義を行ない，完成するつど公開して行くことができます．利用者はこれらを順次必要なつど重ねあわせて利用す

る/プログラミングすることができます．スキーマの公開 (リポジトリ上へのスキーマ･オブジェクトの登録) は，動的に行なわれます．

公開時にすでにリポジトリ上に存在するオブジェクト，リンク/リレーションあるいは属性は，この影響を即座に受け新たな機能を享受することができます．たと利用者オブジェクトからメール箱オブジェクトへの参照リンクの定義を新たに追加すると，この時点からすべての利用者がメール箱を利用することができるようになります．

(3) スキーマベース･プログラミング

第 3 に，PCTE プログラムは，すべて対応するスキーマを持っています．従来はプログラム中に埋め込まれていて，外部からは分からなかったデータ･モデルが，外部にスキーマとして形式化されていることになります．スキーマは，基本的に公開されていて，だれでもが利用可能です．したがって，この公開スキーマを共用することによって，従来プログラムに閉じていたデータの共用が可能となります．つまり，PCTE では，積極的にデータ･モデルを公開して行くことにより，データ中心のツール統合とデータ共用を促進しようとしています．したがって，PCTE 上ではスキーマを先に決定してから，コーディングを開始するスキーマベース･プログラミングを行なうことになります．

ここでの課題は，再利用し易いデータ･モデルを規定したスキーマを定義することと，仕様を固定する部分の公開データ･モデルと，将来変更する可能性のある専用データ･モデルとにスキーマを分けて作成し，その旨表示しておくことです．公開スキーマだけを利用すれば，将来の変更に影響されることはありません．

もう一つの課題は，オブジェクトの内部の形式に対する規定がないことです．PCTE では，オブジェクトとその関係あるいは意味づけについては，スキーマで規定しますが，オブジェクト内部の表現形式は規定していません．この点は，現在
CDIF の拡張あるいは，ISO のソフトウェア工学データ定義交換形式 (SEDDI) の拡張の両面での検討が進んでいます．

(4) プロジェクト管理 (多角的リポジトリ利用)

第 4 に，PCTE リポジトリでは，従来プログラムが扱っていたふつうの"データ"以外に，さまざまなオブジェクトを定義することができます．たとえば，ユーザ，グループ，プロジェクト，課･部，会社などなど，抽象的なオブジェクトも定義できます．これにより，プロジェクトのモデル化などができ，リポジトリ内にソフトウェア開発に関わるあらゆることがらを格納し，意味づけし，利用することができます．プロジェクト管理･バージョン管理･構成管理･利用者管理･製品管理･バグ管理などなど，リポジトリにすべての必要な情報を格納することによって，ソフトウェア開発を総合的に進め，各種データを多角的に関連づけて，総合的に利用することが可能となります．

(5) 分散リポジトリ機能

第 5 に，PCTE は，当初から分散環境を想定して仕様化されています．つまり，利用者あるいはプログラムは，必要とするデータ (オブジェクト) あるいはプログラム (オブジェクト) の物理的な位置に関わらず，スキーマを通して透過的にアクセスできます．また，分散環境においてのホスト停止，ネットワーク停止に対しては，システム資源の複製の保持と正常化時の自動復旧などを行ないます．ロングトランザクション処理時の異常終了などに対処するための，コミット機能などのトランザクション処理機能も備えています．

(6) オブジェクト･アクセスの通知機能

第 6 に，特定のオブジェクトに着目し，これを監視するプログラムに対して，そのオブジェクトの参照･更新･移動･削除などの事象発生時に,"通知先" として関連づけられたプログラムに，監視事象の発生を通知することができます．

(7) セキュリティー機能･会計機能

第 7 に，オブジェクトグループ，プログラムグループ，ユーザグループなどを定義し，グループ単位でのアクセス制限･セキュリティー制限･会計情報取得を行なうことができます．これによって，役割･資格･グループなどに応じて，視点の設定とリポジトリアクセスの制限が行なえます．

3. PCTE とその回りの最新状況

最後に，PCTE とその回りの最新状況について簡単に説明します．

3.1. PCTE 上のツールおよび環境製品

当然というべきか，残念ながらというべきか，現状では PCTE 上のツールはそれほど数多くはありません．これは，これまで PCTE が，欧州を中心に発展してきたことが大きく影響していると思います．多くは，米国のソフトウェア･ベンダが市販しているツールの一部が，カプセル化機構を使って PCTE 上に移植されているというのが現状です．その中には，Teamwork, StP, Framemaker など著名なツールもありますが，いずれもカプセル化さ

れたバージョンです．したがって，先に述べたような公開スキーマを通してのデータ共有の可能性を十分活かしているとはいえません．しかし，将来的には，これら製品も，リポジトリ利用のメリットを活かすような形で移植されてくると思われます．その理由は，欧州連合も米国も政府あるいは軍が積極的にPCTEの利用を推進しているので，ソフトウェア･ベンダもハードウェア･ベンダもしたがわざるをえないからです．

3.2. PCTE Workbench

カプセル化したツールではなく，最初からPCTE上に作られた環境の１つが，米国Vista社のPCTE Workbenchです．この製品は，リポジトリ上に格納されたさまざまなソ

さまざまウェア開発に関わるオブジェクトを，HyperLinkという概念で互いに関連づけ管理する機能を持っています．また，HyperLinkで関連づけられたオブジェクトを，そのオブジェクトを読むためのプログラムと対応づけ，あるオブジェクトの読み出しが指定された時に，自動的に対応するプログラムを立ち上げ，そのオブジェクト内容を画面上に表示することができます．そして，その表示中に他のオブジェクトへの参照があれば，その部分をクリックすることで，そのオブジェクトを呼びだすことができます．つまり，簡単にHyperText環境を分散環境上で，マルチ･ウィンドウで実現できるのです．

このようにして，オブジェクト間にまたがるHyperLinkをたどり，必要な情報を次々に得ることができます．この機能によって，PCTE Workbench環境では，複雑かつ多様なソフトウェア関連情報の網の目(Software Web)を管理し，必要な時に必要な情報を，読みだしツールを意識せずに参照することができます．

また，この環境の中に既存のツールを組込み，HyperLink機能によりそのツールが扱うオブジェクトと，他のツールのオブジェクトを関連づけ統合化することができます．このツール統合を支援する機能がHyperWebサーバーです．

3.3. HP-SoftBench

ツール統合のための環境として有名なのは，HP社のSoftBenchです．HP社はすでに，SoftbenchのEmraude PCTE環境への移植実験を行なっていました．しかし，HP社自身がPCTEを提供していなかった関係上，世の中に出てはいませんでした．しかし，近年のIBM社の機構改革とOpen System志向政策の結果，IBM社のワークステーションを担当するIBMカナダ社は，HP社からSoftBenchとBMS (Broadcast Message Server)技術をライセンスし，RS/6000 WS上にSDE Workbench 6000として製品化しています．このWorkbenchがIBMカナダ社のECMA PCTEとフランスBULL社のISD (Integrated System Development)環境上で提供されています．また，IBMカナダ社は将来のToolTalkベースのメッセージサービスにも対応すると約束しています．

3.4. ISD (Integrated System Development)

ISDは，フランスBULL社のPCTE製品です．この製品は，Emeraude PCTE上に，IBM社からライセンスしたSDE Workbench (つまりHP-SoftBench)を載せることによってデータ統合とツール統合を実現しています．さらに，NIM (Neutral Information Model)というデータの中間表現形式を定め，これを中心に各種のRDBや，メインフレームDB，ツール固有DBとのデータ交換を行なう機能を提供しています．

3.5. SDLC (Software Development Life Cycle)

ソフトウェア開発のライフサイクル全般を支援する環境として，フランスSFGL社のEAST環境とフランスSyseca社のEnterprise-2 (米国および日本ではAlsys社を通じてFreedomWorksの名前で販売)があります．この2つのシステムは，ともに，プロジェクト管理･プログラム開発･ドキュメンテーション･バージョン管理など，ライフサイクル全般を支援するツール群をそなえた総合環境です．

3.6. その他のツール

PCTE環境上にも少数のPDSが公開されています．現在のところ2つのリポジトリ･ブラウザ(リポジトリ内のオブジェクトの検索用)と，音声つき自動スライド表示ツールがftpサーバ(srawgw.sra.co.jp: pub/doc/PCTE/Distribution)上で公開されています．

その他にも，C++, Eiffel, Modula-2に対する言語インタフェースが開発中/提供中です．また，PCTE分散環境に接続したWindows-PC上にWS上のリポジトリを利用するPCTEクライアントプログラムを作成するための開発キットも提供されています．

4. PCTEの移植状況

昨年11月にパリにて開催された第2回PCTE'93国際会議に併設されたツール展示会場では，現時点で購入可能な唯一のPCTEを出荷しているフランスのEmeraude社，昨年末に米国国防省のI-CASEプロジェクトに採用されたIBM Canada社，英国のEDS/Scicon社が，それぞれToolTalk,BMS, CORBA & ToolTalkメッセージング

機能を取り込んだECMA仕様PCTEのアナウンス(1994春発売)とデモを行なっていました．いずれもリポジトリ･エンジンとしてOODBを使っているか，OODBの使用を考慮していました．

また，インドのHuristix社もIBM-PC/AT SCO-OpenDeskTop環境用に，ECMA PCTEのサブセット版の出荷(時期不明)をアナウンスしていました．

5. 国際標準化と関連標準

ECMA PCTEは，CASE環境統合化のためのオープン･リポジトリとして，EUのESPRIT計画の中で開発され，現在ではISOにおいて国際標準化の手続きが進められています(DIS 13719: ISO-IECで1994/3/9投票終了，賛成22票，反対2票，保留2票，したがって早ければ1994年中にも国際標準IS-13719として発行する見込み)．

ISO/SC22日本委員会では，国際語化を仕様に盛り込むことができれば肯定するという旨の投票を行なったようです．その際，日本側で国際語化仕様を提出することを約束したようです．JIS化への準備も始まるようです．

PCTEは，開発主体であるヨーロッパはもとより，アメリカにおいても，NISTその他で高く評価され，最近ではDoD(国防総省)のI-CASEプロジェクトで採用されたりしています．

最新の情報では，ANSI X3H4委員会がリポジトリの国際標準IRDSを置き換える規格として，PCTE規格を将来のリポジトリの基礎となる規格として推進することを決めたようです．

また，PIMBは，OMGと共同してPCTE環境とCORBA環境を統合する技術について検討を開始しています．北米PCTEユーザ･グループ(NAPUG)は，OMGの中の作業グループとして活動することになったようです．米国でのPCTE情報ftpサーバは，これまでSDA社が提供していましたが，この動きに伴い，OMGのftpサーバの一つとしてサービスを継続することになりました．

非PCTE環境とのデータ交換のためには，EIA-CDIFグループと共同で，ソフトウェア･エンジニアリング･データ定義交換規格(SEDDI)として，JTC1/SC7/WG11での検討が始まっています．

現在ECMA PCTEは，第3版としてオブジェクト指向を中心とした拡張について検討を開始しています．

6. 日本PCTEユーザ･グループ(JPUG)について

日本の数少ないPCTEコミュニティにおいても，いいかげんに国際貢献を考えなければいけないという危機感を抱きつつ，今年の1月に，日本PCTEユーザ･グループが発足しました．当日は，急なアナウンスにも関わらず，定員60人の会場をほぼ満席にする57人もの参加を得て(一般参加は41人)，たいへんな盛況でした．

ミーティングは，午前中が，京都大学の松本先生と沢田助手を講師として，PCTEのチュートリアルを行ない．午後は，松本先生による「日本PCTEユーザグループ(JPUG)設立報告」と質疑応答．JPUG会員による「PCTE'93視察報告」と続き，最後に一般参加された方の中から，SEA元代表幹事PFU(株)の熊谷さんに即席のチェアマンをお願いし，PCTE'93に参加した松本先生を始めとする上記会員諸氏をパネラとして，会場と一体となった討論が行なわれました．次回は，春に関西地区でのミーティング開催が予定されています．

JPUG会員には，ISOの日本委員会およびその下の作業部会に参加している方々が多数含まれています．まだ発足したばかりで実質的な活動は行っていませんが，会長の京都大学の松本先生を初めメンバの多くが，この活動を通じてPCTE技術への国際貢献と，PCTE技術の国内での普及に微力ながらも力を注ぎたいと考えています．

7. まとめ

以上，PCTEの歴史，PCTEとは何か，PCTE上のツールおよび環境，国際標準化の動向と，PCTE回りの技術動向について簡単に説明しました．すでに，PCTEは，将来の技術ではなく，現実の技術となっていることがお分かりいただけたかと思います．

しかし，PCTEは，それをを使えば何でも解決する「ドラえもんの何でもポケット」かといえば，そうではありません．お気づきのように，PCTE仕様の中で充実しているのは，リポジトリあるいはオブジェクト管理システム(OMS)部分だけで，ほかの部分では，基本機能だけしか規定されていません．その理由は，PCTE設計者にとって一番大事だったのが，リポジトリを中心とするデータ統合と共有であったこと，また，他の部分はまだ技術的に未成熟で，有力な技術あるいは事実上の標準が存在していなかったことが，大きな理由だと思われます．これは，逆に考えれば充実したリポジトリ仕様を中心として，他の分野の技術，たとえばToolTalk / CORBAのような，別に開発されて事実上の標準となった技術とを組み合わせ，互いに保管させて行くだけの仕様上の余裕がPCTEにはあると考えることができます．

PCTEから派生した，ソフトウェア開発環境の参照モデルという大枠の中で，PCTEリポジトリを中心とし，各

種技術を組み合わせて，よりよい環境を構築して行くことになると思います．

また，現在実用段階にまで達しつつあるオブジェクト指向の技術，特にOODBとCORBAへの対応も，各社のPCTEの実装にあらわれはじめています．これを受けて，オブジェクト指向技術を取り込みを目指すECMA PCTE第3版の検討が始まっています．国際語化が行なわれるのも間もなくでしょう．

しかし，残念ながら，現在PCTE上で稼働するツールは少なく，ツール･ベンダのPCTEへの移植の約束も今のところ履行されてはいません．この点はおそらく将来に渡っても改善される見込みは少ないでしょう．なぜなら，ツール･ベンダは，数の論理つまり利益が多いところに集中的に投資を行うからです．したがって，現実的には，拡大PCTE環境として，従来の環境つまりUnix環境およびPC環境，メインフレーム環境を接続し，相互にデータとツールを共用しあいながら，ソフトウェア開発を行なって行くことになると思われます．この線に沿った仕様上の検討が次のECMA PCTE第3版の中心だと，私は解釈しています．同様に，各社が提供するPCTE実装におけるOODB利用，メッセージング技術の利用，外部データベースデータの交換機能などが，この仕様を先取りする形で提供されて行くことになります．

もう一点，米国ANSI IRDS (X3H4)検討委員会のIRDSを止めて，PCTEに乗ることを決定したという事実に見られるように，より大きな枠組みのPCTE仕様は，IRDSが目指していた事務処理関係リポジトリも吸収し，PCTE上での実現を目指すことになるようです．しかしながら，現在メインフレームDB上で実現されている機能およびRDB上で実現されている機能までを取り込んでPCTE上に実現するというのは現実的ではありません．これら既存のDBとPCTEリポジトリ環境は，一方を他方が置き換えるのではなく，データ交換機能を中心に共存して行くことになります．(BULL社のISDなど)

結論としては，ソフトウェア開発者は，その開発効率の向上と開発プロセスの改善と管理，ソフトウェア資源の有効利用のために，いかにリポジトリを活用して行くか? いかにリポジトリを中心とした開かれたソフトウェアを作成し，自社の開発に役立てるか? 更に進んでソフトウェア･コミュニティにこのようなツールを提供し，全体としてのソフトウェア品質の向上，データモデル共用と利用の促進，データ共用の促進に役立てるかを真剣に考えてみる時期に来ているのではないかと思います．これが結局は，国際的なソフトウェア･コミュニティへの貢献につながるのではないかと思います．

付　録

●関連略号一覧:

PCTE	Protable Common Tool Environment
IPSE	Integrated Project Support Environment
ECMA	European Computer Manufacturers Asscociation
PIMB	PCTE Interface Management Board
NIST	National Institute of Standards and Technology
OMG	Object Management Group
CORBA	Common Object Request Broker Architecture
IRDS	Information Resource Dictionary System
EIA	Ectronic Industries Association
CDIF	CASE Data Interchange Format
SEDDIS	Software Engineering Data Definition & Interchange

●参考文献 (書籍):

書名: PCTE - The Standard for Open Repositories
著者: Lois Wakeman & Jonathan Jowett
ISBN: 0-13-065566-X
出版社: Prentice Hall
価格: $48.00
[*] 最近東京日本橋丸善で数冊が平積みされていたそうです．

書名: PCTE'93 Proceedings
入手先: Michele CHARLES, EC2
269-287, Rue de la Garenne
92024 Nanterre Cedex, France
TEL: (+33.1)47-80-70-00
FAX: (+33.1)47-80-66-29
価格: 600.00 FF (送料込み)

●参考文献 (雑誌):

(1) 鯵坂、沢田、満田: "ソフトウェア評論 Emeraude PCTE", コンピュータ･ソフトウェア Vol.10 No.2, Mar. 1993.
(2) Ian Thomas: "PCTE Interfaces: Supporting Tools in Software Engineering Environments", Nov.1989

IEEE Software.
(3) 塩谷: "ツール間連係を実現する CASE フレームワーク PCTE", 日経エレクトロニクス 1992 年 12 月 14 日号外.
(4) J.C.Ferrans, D.Hurst et al: "Hyperweb: A Framework for Hypermedia-Based Environment Dec.1992, ACM-SDE.
(5) Ken Thompso: "CASE Tool Data Integration: The European Standards Management, Summary & Conclusions (First Draft).

● anonymous ftp archives:

京大： ftp.kuis.kyoto-u.ac.j
(cd doc/PCTE)
SRA： srawgw.sra.co.jp
(cd pub/doc/PCTE/Distribution)
OMG： amethyst.omg.org
(cd /pub/PCTE/ECMA_PCTE)

● アーカイブ上の PCTE 紹介文献
(PS ファイル gzip 圧縮済)

69379 Sep 6 1993 PCTE_Overview.ps.gz
18190 Sep 13 1993 PCTE_summary.ps.gz

最後に，日本 PCTE ユーザグループの案内をつけておきます．

日本 PCTE ユーザ グループ
Japanese PCTE User Group(JPUG)

◇事務局: 準備中.

◇暫定連絡先:
京都大学 工学部 情報工学教室
松本研究室 (担当: 朝井)
〒606-01 京都市左京区吉田本町
TEL: 075 - 753 - 5375
FAX: 075 - 753 - 4970
E-MAIL: jpug-request@kuis.kyoto-u.ac.jp

◇会の趣旨: ソフトウェア･リポジトリとして国際標準になりつつある PCTE (Portable Common Tool Environment) に関して，国際的な動きを紹介し，かつ技術の普及，発展，および国際協力の推進を目的とする，非営利団体である.

◇会員資格: 個人 (制限なし)

◇組織:
会長： 松本 吉弘 (京都大学)
幹事： 鯵坂 恒夫 (京都大学)
伊藤 昌夫 (MASC)
大城 卓 (新日本製鉄)
小野 茂 (シグマ システム)
落水 浩一郎 (北陸先端大学院)
松尾 正敏 (SRA)
向山 博 (JIPDEC)
米山 隆 (PSI)

◇活動: e-mail および ftp サービスによる情報の交換を行うほか，年 4 回程度のミーティングを実施する.

◇ 会費: 当分のあいだ無料.

● 日本 PCTE ユーザグループ (JPUG) 入会申込書

暫定申込先:
京都大学 工学部 情報工学教室
松本研究室 (担当: 朝井)
〒606-01 京都市左京区吉田本町
TEL: 075 - 753 - 5375
FAX: 075 - 753 - 4970
E-Mail: jpug-request@kuis.kyoto-u.ac.jp

氏名: (ふりがな)
会社名/学校名:
部門:
役職:

住所:(〒 -)
TEL:
Fax:
E-mail:
PCTE の使用経験: □ あり □ なし

塩谷 和範
e-mail: shioya@sra.co.jp
Fax: 03-3351-0880

SEA関西システム技術分科会
「賢い情報端末vs賢いＳＥ」

パネラー(順不同、敬称略)：
阪井@奈良先端大(SRA)　　麓(ふもと)@応用技術
牧野@オムロン ソフトウェア　　臼井@jip
柳原@クボタ　　中野@通信.阪大

司会/記録： 横山@松下電器

1994.1.20 18:30-20:30 日本電子計算(株)会議室にて　参加者２３人

過日、情報端末ってどんなもんか？
どんな端末なら使えるか？使いたいか？
利用者視点から情報端末を作りあげてみようということで、SEA関西では、パネルディスカッション形式の分科会を持ちました。

メーカーはだらしない!
ユーザの欲しいものを提供していない!
等々辛口の言葉も聞かれましたが、終始穏やかな雰囲気の中、和やかに討論は続くのでした。その場でアンケートもしました。
持参した情報端末の話は尽きず２次会まで、もつれ込んだのでした。
　とてもここに一部始終は載せられませんが、少しでも何かのご参考になればと考え抜粋ですが掲載します。

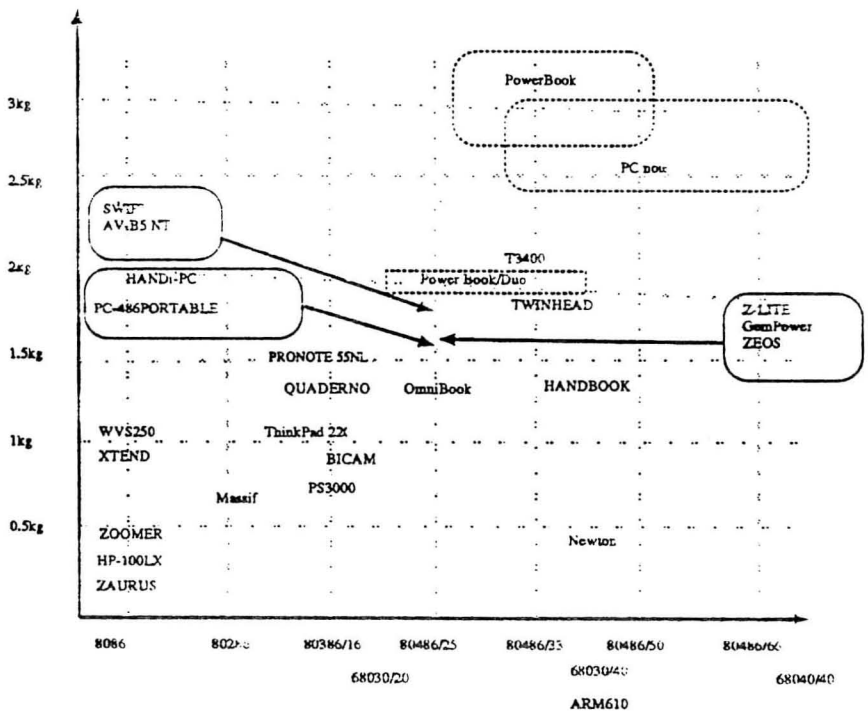
　さて、ここに登場する「うすいpad」を最初に提供するのはどこかな？
今後が楽しみですね。

掲載している項目は、以下の通りです。

1. 分科会での発言の記録
2. seakansai メーリングリストでの
   事前アンケート
3. 分科会での挙手によるその場アンケート

最後に、事前アンケートにお答えくださった方々、分科会で発言してくださった方々、その場アンケートにお答えくださった方々、議事進行にご協力くださった方々、紙面をお借りして厚くお礼申し上げます。

阪井@奈良先端大(SRA)さんがまとめた 携帯端末の分類



<パネル討論>

阪井@奈良先端大(SRA)

用途：一番重要な問題を解決する

→ どこからでもインターネットにアクセスできる.

麗(ふもと)@応用技術

用途:X端末の補助
パソコン通信
画像
プレゼンテーション

牧野@オムロン ソフトウェア

用途: 個人ユースとして
秘書
TEL
DM
FAX
メモ
スケジュール
Just in time(会社、自宅、出先)
光通信
単3乾電池で東京<->大阪往復使用可能
重量:720g(愛妻弁当相当)
価格:10万円以内

柳原@クボタ

用途:スケジュール,メモ,ワープロ,通信(キーボードは不要)、長距離(1-2時間)移動中に使用
重さ:500gまで

以前ダイナブックで腰に障害を起こした事があり軽さは絶対条件

中野@通信.阪大

用途:出張、通信、講演、VGA
電子手帳(カシオ8100)を使用
ICカード辞書(国語、漢和、英和、和英)
メモ、電話帳、電卓、スケジュール
仕様:漢文(持ち運びするため)、音声

今後は、2極化していく
(軽い:massil,普通:gateway2000-1.3kg)
システム手帳は、バックアップが取れないと便利でない。

臼井@jip

情報端末：
いつでもどこでも、行動・知識・知恵の諸活動を支援するために、情報の収集・加工・発信ができる端末

将来の姿：
3つに集約される。情報端末、情報机、情報家屋。情報机は机上が全面液晶の机。
情報家屋は全ての壁が全面液晶で表示したいところにウィンドウが表示できる。

<うすいpad仕様(臼井さん提唱) 概略図添付>

表示:液晶マルチウィンドウ
無線:TV,ラジオ,トランシーバー,TEL,FAX
通信:光通信
グループウェア:ビデオカメラ、マイク、スピーカー
接続端子:プレゼンテーション、キーボード
情報ステーション、ヘッドフォン、
B-ISDN
性能:メモリー10GB,1000mips
電源:太陽電池
重量:180g
価格:10万円
発売時期:3年後

竹中@プロップステーション

21台のMacintoshを使っているがすぐ壊れる。

(特に、トラックボール、ボタン)

->修理費がかさむ。

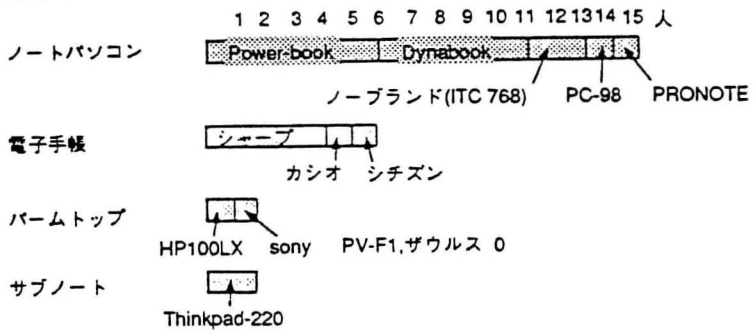
<seakansai メーリングリスト 事前アンケート結果>

私の欲しい情報端末

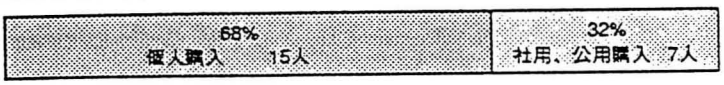
氏名	阪井@奈良先端大	ふもと@応用技術	まきの@ オムロンソフトウェア	中野@通信.阪大
目的	ドキュメントの作成(WYSIWYG) とブラウズ システム手帳のかわり メインパソコンの非常時用 こたつトップパソコン(^_^;)	何時でも何処でも、ニュースソース へのアクセスをしたい。	Personal Communicator	普段は研究室にあって X端末の補助 出張ではログイン端末 ときどき授業や講演での プレゼンテーション装置 それでXなぞがうまく動く と家からX環境でログイン
重さ	1.5(1年後は1)Kg 以下	1Kg 以下	880g 単3×4込み	
大きさ	B5(1年後はA5)ファイルサイズ	流行りのポーチに入る大きさ、 もしくは背広のポケット。 大きくてもＶＨＳテープサイズ。	B5＋α	
I/F	マウスがわりになるデバイス フルキーボード (サイズは小さくていい) ICMC(JEIDA4.1 論理／物理 フォーマット) フロッピー (外づけのこと) (3mode:1.44/1.2/720) シリアル、パラレル、 PCM音源 ビデオ(1000x768)出力 拡張バスユニット 外部CD-ROMユニット (Photo-CD,2～4倍速)	ペン入力 キーボード（オアシスのポケット２ くらいがいいな）	フルサイズキーボード FD接続可、 ICカード2枚挿入可能	
性能	i80486SX 25-50 (1年後はDX2 66MHz) または 68030(25MHz) (1年後は68040 33MHz)以上 メモリー8(1年後は16)M (Macなら16M(1年後32M))以上 HDD 120M (圧縮できるなら60Mでも可) (1年後は330M)以上 バッテリー駆動2.5時間以上 640x480画素階調表示 (1年後は800x600画素256カラー)	ストレス無く利用できること。# 電池での駆動時間は、通常利用で、 東京、大阪間を新幹線で往復できる 時間であること。 （今なら、６時間位!!）	F8680/8Mz DR-DOS Ver5.0 メモリ:ユーザー1M システム1M	
機能	スクリプト実行機能、GUI	電子メール、ニュース等が読み書き できること。 電子ブックが読めること。 ＣＤもしくはＭＤが聞けること。 ラジオやＴＶが聞けること。 ☆ニュースソースは、有線、無線、 パッケージの三種類が、扱えないと ヤダ	パソコン通信FAXモデム	
アプリケーション	スケジュール管理、住所録、 スプレッドシート、 vi,Emacs,TeX 実行機能		スケジュール、 ワープロ(Vje-Pen)、 備忘録、世界時計、 名刺カレンダー、 メモ、電卓 以上標準装備	
価格	実売価格20万円以下 （買えませんけど(^_^;) (1年後は¥168,000)	最初は10万円へ、 普及すれば2～3万円	¥99,800	

<参加者アンケート結果>

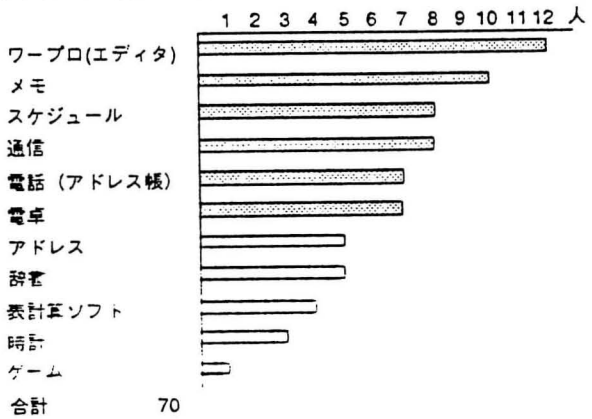
1.何をもっていますか？



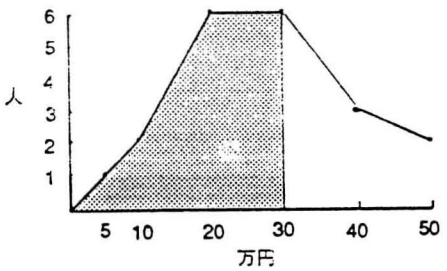
2.費用の出所はどこですか？



3.使っているソフトウェアは何ですか？



4.うすいpad いくらだったら購入しますか？



プロセス成熟度モデルに関する議論
― オブザーバの視点からのサマリ (連載予告) ―

岸田 孝一
(SRA)

インタネット上でソフトウェア·エンジニアリングに関する話題を議論するニュースグループ comp.software-eng で，この2月初めごろから (だと思う．私がウォッチしだしたのは2月の下旬から)，ソフトウェア·プロセスの成熟度モデル (主として CMU/SEI が提唱している CMM) をめぐる賛否両論が，延々と続けられている．いま，この原稿を書いている4月7日現在でも，激しい (?) やりとりが続いていて，しばらくは収束しそうにない．

興味深いのは，CMM が日本的品質管理 (TQM) の延長線上で論じられていること，それからまた，もともとプロセス成熟度の概念が出てきた DoD まわりのソフトウェア·エンジニアリング·コミュニティとはあまり関わりのない民間市場相手のソフトウェア·ビジネスの関係者 (たとえば，PC 向けの各種ツールや環境プロダクトを開発している人たち) が，「現実はもっとキビシイんだぜ!」といった立場での批判を展開していることである．

過去7週間ためこんだアーカイブがそろそろ 1M バイト近くになるので，1人のオブザーバの立場から，独断的かつ偏見に満ちたサマリを Seamail にと考えているのだが，何せ時間がない．次号から，何回かに分けて連載させていただくことにして，今回は，去る3月 Forum の参考資料として用意した「いくつかの代表的意見のサンプル集」を，予告篇のつもりで載せることにする．

---

1. 賛成論

Opinion-1:

最近，私の会社で開発プロセスのアセスメントが行われた．そこで得られた教訓．

(1) SEI の成熟度モデルは，プロジェクトが時間の経過とともにどんな風に成熟して行くべきかをきわめてよくあらわしており，その意味で，1つのガイドラインとして有用である．ただし，それは決して「銀の弾丸」ではない．すなわち，ただ何も考えずにそれにしたがって行きさえすれば結果が得られるということはない．われわれは，モデルの基礎となっている原理を正しく認識し，それを守って行く必要がある．多くの人々は，このモデルを，モデルではなく，遵守すべき命令の一覧表であるかのように誤って解釈しているきらいがある．

(2) プロジェクトのすべての部分は相互に関連を持っており，したがって，一緒に成熟して（させて）行かなければならない．SEI のモデルは，そのことを教えている．われわれのプロジェクトの場合，SCM(構成管理) の部分だけが進みすぎていて，いくつかの問題が発生した．モデルを持つことの利点は，そうしたプロジェクトの歪みをだれもが認識し，管理者に対して適切な報告を行えるということである．

Opinion-2:

新しいプロセス指向の考え方および CMM(プロセス·マネジメント) は，従来のアセンブリ·ラインとは違って，幅広い訓練を受けた熟練技術者が，グループあるいは組織間の連携を重視しながら，高品質のシステム開発という目標を目指して努力するというものである．アセンブリ·ラインは製造のための仕掛けであり，ソフトウェアはエンジニアリングなのだ．

もし，品質上の基準が達成できなければ，全員が迷惑をこうむる．CMM が述べているように，プロジェクトの見積りは妥当なものでなければならず，方法論はチーム·メンバによって十分マスターされていなくてはならず，各種の定量的指標が入手可能であり，また，最適化が，プロダクトのみならず，プロセスそれ自身についても達成されなければならない．こうした目標は，万人にとって利益をもたらす．より高品質のシステムをより低コストで開発できるような最適化は，長期的な展望にもとづくたえざる改善を通じて初めて可能になるのである．

CMM は，品質およびソフトウェア·エンジニアリングに関する基本的な標準，いいかえれば，ソフトウェア開発のための高品質フレームワークとそれを達成するための手段をプロモートするものだと考えられる．それは，ソフトウェア開発に関するすべての問題点を解決する銀の弾丸ではない．特に，そのエンジニアリング·プロセス的な側面をターゲットにしたものである．もちろん，それ以外に，ビジネスおよびマネジメントの問題がきわめて重要である．したがって，CMM は，それ単独で，単なるアセスメントの道具として用いられるべきではない．それは，あくまでも，エンジニアリング·プロセスを評価するためのツールなのである．

Opinion-3:

W. ハンフリーは次のように述べている：「ソフトウェア·プロセスにおけるもっとも重要な構成要素は人間であ

る．よりよい仕事をしたいという人びとの欲求を認識することが，まず大切である．したがって，人間を矯正するのではなく，プロセスを矯正することに焦点をあてなければならない．もし，人間に問題があると管理者が考えたとすれば，プロセス改善の動きは，恐怖感をもって迎えられ，人びとは自分の仕事がどうなるかを心配し，結局は改善に反対するようになるだろう．」

SEI のアプローチは，まず現状のアセスメントに始まり，ついで連続的な改善へとつながって行く．アセスメントは，数多くの人びとへのインタビューを含んでおり，そこで発見されたことがらは，上級管理者に報告される．このアプローチは，デミングが日本で行ったのと同じものである．デミングによれば：

- 人びとの恐怖感を取り除き
- リーダシップを導入し
- 全員を改善活動に巻き込み
- 訓練を繰り返す

ことが大切である．つまり，基本的な考え方は，マネジメントが従業員の声によく耳を傾け，すべての人間が相互に学びあうということである．現在，錆びついた製造ライン上で時間を浪費しているプログラマたちにとって，そうした改善は歓迎すべきものであろう．

Opinion-4:

CMM や TQM は，人間の教育や能力開発をもっとも重要な要素として扱っている．しかし，同時に，「必要な」な構造や指針を提供するものでもある．人びとは，「つねに」会社のビジョンやゴールとの調和を考えながら行動するように要請される．そうしたビジョンやゴールを自分自身の行動目標のレベルにブレークダウンして考えなければならない．そして，そのさい，既存の標準に盲従することは固くいましめられる．もし，現在の標準が創造性を阻害し，顧客の要求に矛盾するのであれば，その場で変更し行動すればよい．これは，すでに日本でふつうに行われていることである．これが TQM であり，それはまた，CMM の基本的な精神でもある．

Opinion-5:

プロジェクトが巨大化するのは，決して管理者たちの帝国主義的欲望のせいではなく，必然的にそうならざるをえないからである．たとえば，ある種の通信制御ソフトウェアは，もし少人数で時間をかけて開発していたのでは，市場ニーズに追いつかず，競争に破れてしまう．そうした大型のプロダクトを少数精鋭のチームで開発するやり方など，まだ，だれにもわかっていない．

したがって，われわれは，やむなく大規模なプロジェクトの運営を強いられることになる．そして，そのさい，CMM が助けになるのである．

また，少規模なプロジェクトを複数抱えている場合にも，CMM は，それらのプロジェクトが相互に経験を学びあい，同じあやまちを繰り返さないようにするのに，有用であろう．

ハンフリーやデミングが強調しているのは，管理者はさまざまな形で部下のスタッフの声に耳を傾けるべきだということである．「いつでも交代が可能な要員」などというコンセプトは，かれらの頭には初めから存在していない．むしろ，その逆である．もしある人間の仕事がうまくない場合には，きちんとカウンセリングして，かれに向いた仕事を探すべく会社が努力しなければならない，とデミングは述べている．もちろん，こうしたコンセプトは日本で生まれたものである．人びとが一生同じ会社で働くので，会社側としても，それぞれの社員の素質に適した仕事を探しだし，人材を有効活用しなければならないわけである．

Opinion 6:

能力開発の試みは，必ずしも無秩序なプロセスと結びつくものではない．たとえば，総合的な品質管理体系の中での草の根型 QC 活動や，新技術実験プロジェクトにおける PDCA サイクルの実践といった例を考えてみればよい．

標準とは何らかのマニュアルの形に文書化されるものだというのは，誤った固定観念である．プロセスや方法論の定義を含む標準は，まず最初は柔軟なものでなければならない．それは何を第一義に考えて行動すべきかをはっきりと記述したものでなければならず，すくなくとも 6 ヶ月ごとに，できればそれを利用している人びとの手で改訂されるべきである．

2. 反対論

Opinion-7:

アセンブリ·ラインの概念が発明される前，自動車は一度に 1 台ずつ，職人 (craftman) の手で作られていた．やがて，ヘンリー·フォードが登場し，もっと効率的な仕組みを導入した．フォードがやったのは，自動車作りの仕事のプロセス記述を，「さまざまな機械技術に精通し，複雑な活動に取り組んでそれを完成へと導けるような，経験を積んだ職人」のためのものから，「ベルト·コンベアによって定められたペースにしたがって，反復的な仕事を一定時間行えるような，半熟練労働者」向けに変換したことである．

われわれは，いま，ソフトウェアの組立工場化の道を歩みつつあるのか？

われわれの会社のプロセス成熟度レベルが 1 から 5 に上がったとき，ソフトウェア技術者の質はもはやあまり問題ではなくなり，「プログラミングの楽しさ」は失われてしまうのではないか？

Opinion-8:

現実世界の仕事の上では，プロセスよりも重要な要素がいくつも存在する．もし，それらの要素を考慮に入れないならば，プロセスに関するほとんどの努力（もしかしたらすべての努力）は，無に帰してしまうであろう．このことは，特に，会社内におけるオフィシャルなプロセスの場合に真実であるが，プロジェクト･レベルで定義されるプロセスについても，あてはまる．

反復可能なソフトウェア･プロセスという考え方は，もしかしたら，有用かも知れない．しかし，それよりもむしろ，正しい訓練を受け，経験を積んで，十分な指導力を持つ技術者をそろえて，プロセスを実行させたほうがよい．

CMM の全体的な構成の中では，わたしが大切だと考えているスキルや経験は，どちらかといえば，軽視されているように見える．私には，だから，CMM が目指しているのは，中身のない貝殻のように感じられる．

もちろん，わたしたちは何らかのプロセスを実行している．しかし，別に，あらかじめ定められた特定のプロセスに従う必要はない．ときには，仕事の途中で新しいプロセスを発明したりする．与えられたプロセスを手直しすることもある．

わたしたちは，ある種の水晶球によって前途を占ったりもする．それは，人間の "頭脳" という水晶球である．これは，科学の力ではモデル化できないすばらしい装置である．頭脳の数がふえればふえるほど，仕事の筋道はきちんとしてくる．そして，既存の手続き上のルールから離れて，自由な意志決定を下すことができるようになる．

そうした意志決定は，できあいのルールよりもはるかに柔軟なので，わざわざプロセスを研究してたくさんのルールを書きためるよりは，すぐれた頭脳を集めること，または，現有の中級の頭脳を改良することのほうが，ずっと有益であろう．少なくとも，柔軟性の持つ利点に気づくことができるのだから....

Opinion-9:

たしかにソフトウェア開発は製造型のプロセスではなく，エンジニアリング活動であるが，しかし，これまでに私が経験した大型のプロジェクトは，いずれも製造工場的な様相を呈していた．

たとえば，何十かの類似の問題のインスタンスを含むようなソフトウェア開発作業があったとしよう．私なら，いくつかの解の共通点を見つけだして，テーブル化し，すべての解を生成することを考えるだろうが，ふつうの管理者が思いつくのは，全体をいくつかに分けて各グループに割りふることだけであろう．そうしたやり方を続けて行けば，いずれは，何らかの組織的解決方法が必要になってくる．そこでの選択肢は，(1) 少数の有能な技術者を必要とするような技術的解法か，それとも，(2) 平均的能力のプログラマ･グループを集めればすむような管理指向型の解法のいずれかである．後者の場合，ベルト･コンベアによって，類似の問題群が次々に供給され，全体の問題に対するエレガントな解法のことは完全に忘れ去られてしまう．

CMM のアプローチは，権力の帝国を作り上げたいと心の中で望んでいる管理者たちのメンタリティにとって，絶好の餌であるかのように思われる．大規模組織に関する問題を解決するこのメカニズムの発明は，まさしく自己実現的な予言としての効果をもたらす．われわれはいまや大規模なスタッフを管理するすべを手に入れたのだから，そうしようではないか．そして，状況は以前より改善される．ソフトウェア･クライシスもこれで何とか乗り切れそうだ．

こうして，ソフトウェアの持つ本質的な特性や，コンピュータ･サイエンスの進歩がもたらした新しい開発技術，あるいは平均的な人間と比較すれば数十倍の仕事をこなせる優秀なプログラマの能力などはまったく無視され，次々に大型のプロジェクトが立ち上げられる．その問題がほんとうにそうした大規模なスタッフを必要とするのかどうかが検討されることはほとんどない．なぜなら，大規模プロジェクトを運営するのに必要な組織上のマシンが発明されたからである．

Opinion-10:

「組織」とは，単にわれわれの頭の中にひそんでいる概念にすぎない．すべての重要な問題は，一定のスキルをそなえた人間の手で解決される．われわれの社内で，いくつもの重要な問題を解決しつつある人びとがおこなっているほんとうのプロセスやかれらの能力を，「凍結乾燥」してどこかに保存しておくことなどできはしない．

どの会社でも，熟練スタッフの何人かが退社したとすれば，その損失はすぐには回復できないであろう．ソフトウェアの設計が，ハンバーガーショップでポテトフライを揚げるのと同じようなものであるならば，そのプロセスをきちんとドキュメント化しておきさえすれば，たとえ技術者が退職したとしても，すぐに代わりの人間をみつけることができるだろう．しかし，われわれのビジネスでは，そうは行かないのである．組織を横断する (または越えるような) コミュニケーションのスキルは，さらに複雑な要因である．ほんとうのプロセスは，だれにもモデル化できないし，単純化することもできない．

私が，スキル開発のほうがプロセスより重要だという意見に固執する理由は，そのあたりにある．いまわれわれにとって必要なのは，十分な人数の熟練技術者およびリーダを揃えてチームをきちんと構成し，使命を遂行するのに十分な時間と開発環境を提供することである．だれかが記述したプロセス･ドキュメントがいかにすぐれたものであろ

うと，同じ人間が直接仕事に携わったほうが，プロジェクトに対して与える影響は大きいだろう．ときには，ドキュメントもたしかに有用である．しかし，通常，それを作るためのコストが高すぎるように思われる．

Opinion-11:

成功への鍵は次の３つである：(1) 人びとのスキルを開発すること，(2) リスクを認識しそれをうまく回避すること，そして (3) ビジネスに精通すること．このうち，第３の要因は，部分的にではあるが，CMM においても扱われている．Capability とは，明らかにビジネス上の概念である．

私は決してプロセスに反対しているわけではない．私がいいたいのは，プロセスは問題の焦点ではないということである．プロセスまわりの議論は，たしかにカッコイイし，興味深い．しかし，その背後にもっと重要で優先的な要因が隠れているのである．

どんなにすぐれたプロセスを導入しようと，それを遂行する人びとが十分なスキルを持っていなければ，ほんとうの目的を達成することはできない．なぜなら，ソフトウェア開発にさいして人間が行なう問題解決のプロセスは，絶対に単純化しえないからである．そのプロセスは，モデル化することさえできない．したがって，もし，あなたの会社の管理者や技術者が適切な訓練を受け，十分な経験と能力を持っていなければ，プロセスの確立にさいしてあなたが費やしたすべての努力は「水泡」に帰してしまうのである．

また，リスクに関して十分な検討と見積りがなされていなければ，どんなプロセスも効果がないし，結果として，会社に大きな損害を与えることになるだろう．そして，もちろん，プロセスは，対象とするビジネス分野に関する十分な知識にもとづくものでなければならない．

もし，上にあげた３つの要因をきちんと考慮に入れて行動すれば，たとえ反復可能なプロセスを持っていなくとも，うまく行くだろう．もし，これらの３つの次元を考慮した反復的プロセスが用意できているというなら，それに越したことはないが....

私のいうスキルとは，ソフトウェア開発という仕事の背後にあるダイナミクスに関する訓練や経験を十分に持ち，必要に応じてプロセスを定義しなおすことができる能力のことである．

Opinion 12:

一般に組織というものは，創造性や技術の進歩を嫌う．少数の熟練者によるエレガントな仕事よりは，むしろ，平均的な人間の手による月並みな仕事の進め方を好む傾向がある．そのことが別に悪いとはいわない．しかし，そうした大組織は，決して万人のためのものではない．

ある種のソフトウェア開発の仕事は，そうした伝統的な大組織には適していない．管理者たるものは，この事実をはっきりと認識しておくべきである．単細胞の管理者たちは，よく，「もしこのプロセスに従えないのなら，君はだらしないハッカーだ！」いったいい方をするが，それはまちがっている．どんなプロジェクトでも唯一の正しいやり方で運営できると考えている管理者諸君は，顔を洗って出直すべきである．

ハッカーに対して否定的な態度をとる管理者が多いが，これもおかしい．もっとも優秀なハッカーたちは，ふつう，大学や研究所で働いている．かれらは，また，すぐれた文章家でもある．なぜなら，研究の成果を世の中に発表する義務を負っているからである．人並以上のコミュニケーション能力なしには，その世界では生きて行けないのである．

3. 傍観者の眼

Opinion-13:

ここでの興味深い議論は，ソフトウェア開発上の問題点に対する自然な反応によって，人々をはっきりと２つのグループに分割しているようである．すなわち，一方のグループは，開発技術者個人に対して何らかの制御を加えるようなシステムを構築するような反応を示し，他方は，システムからの制御を最低限に押さえて個人の能力の発展を意図しようとする．

ここでの議論からすると，ソフトウェア･ハウスが成功を収めるには，そうした対極的な２つの考え方のいずれか一方を採用し，それを最適化するしかないように感じられる．たとえば，システム重視型の会社は，要求仕様の明確化につながるようなエンジニアリングを考えるべきであり，人間重視型の会社は，顧客側の要求仕様が決して定まらないような問題の解決を目指すようなエンジニアリングを考えるべきであろう．

私個人としては，そうした２つの考え方の効果的な組み合わせに関心がある．すなわち，「組織の一部としてのよりよい個人のあり方とは？」とか，「個人の影響によって組織が強くなるには？」とかいったアプローチである．しかし，これまでの議論を見るかぎり，それぞれの陣営とも，相手方の価値観をまったく認めていないように思われる．その点が，私にとっては不満である．

Opinion-14:

議論のポイントをわかりやすくするために，２つの陣営の諸氏に，特定の事例を挙げて具体的に説明してほしいと思う．たとえば：

1.「私は明確に定義され制御されたプロセスの有効性を信

じているが，しかし，次のケースでは，どうにもうまく行かなかった....」

2.「私は制御されたプロセスを重視しすぎるのはよくないと信じているが，しかし，次のケースでは，そうしたプロセスなしには....」

Opinion-15:

私の考えでは，2つの陣営は次のように分かれている：

(1) 開発プロセスのモニタリングや改善という考え方に価値を認めているグループ．ここで，モニターの対象はプロセスであって，人間ではない．もし何か問題が発生したとすれば，それはきわめて容易に発見でき，解決できる．その解決作業自体もプロセスの一部分である．

(2) どんなプロセスやシステムにもあまり価値を認めていないグループ．すべての問題は新しくユニークなものであって，その解決には一定の知性と才能が必要だと信じている．プロセスの概念が存在しないので，以前に類似の問題が解かれたかどうかの知識もありえない．したがって，同一の (または類似の) 問題を，別々の人間の手で何度も解決しなおすことで，多くの創造的エネルギーが浪費される．

いずれの場合にも，人間がきわめて重要な因子である．

Opinion-17:

プロセスのモニタリングや品質の定量化に関連して，「あるアプローチについて計測を行なうことは，そのアプローチを理解することだ」というケルビン卿のことばを，だれかが引用している．そして，それを拡張して「いいかえれば，もし計測を行わなければ，あなたはそれを理解しないことになる」と述べているが，これはおかしい．「p ならば q である」という命題が成り立つからといって，「not p ならば not q である」とはいえないのである．

私は，知性·経験·想像力·コミュニケイションなどのプロセスは，少なくとも，ソフトウェア開発プロジェクトの実践的な側面に関していえば，だれにも理解できないようなものであると思う．これらの要因をどう扱うかであるが，たとえば，定期的なプロジェクト (チーム) のミーティングにおいて，「われわれのプロジェクトにそうした要因 (たとえば知性) がどんな形で働いており，どんな結果をもたらしているか？」を全員で議論してみるのもおもしろいであろう．それを何ヶ月か続ければ，必ず，そのチームあるいは会社全体のプロセスの成熟に何らかの影響があらわれるものと予想される．

一方また，プロジェクトにおいて，心理学的または社会学的なプロセスがどんな風に働いているかを理解すべく努力することも重要である．私自身，あるプロジェクトでテスト·カバレジの導入を試みたが，みごとに失敗した．原因は技術的なものではなく，心理的なものであった．つまり，チーム·メンバが私の意図を理解してくれず，単に「仕事がきつくなる」という解釈のもとに反発したからである．結局は個々のメンバとの徹底インタビューを通じて，ひとりひとりが何を感じ何を考えているかを把握しなければならなかった．

4. カウボーイの TQM

Opinion-18:

多くの人びとが TQM や CMM を抑圧のためのツールだとして恐れているのは，悲しいことだ．私の考えでは，TQM は，もしそれが完全に具体化されるならば，人間の創造性を支援するための最終的なツールなのである．

(1) TQM は西欧的思考に向いていないということはない： TQM は開発プロセスではなく，任意のプロセスを改善するための方法論である．日本人がそれを，コンセンサス指向型の設計プロセスに適用したのは，かれらの国民性がそうさせたのであって，アメリカがそれを見習う必要はない．

(2) TQM は階層的管理を意味しない： TQM が完全に具体化されれば，すべての作業グループが自分自身のプロセスをコントロールできる．マネジメントは仕事のやり方まで指示することは許されない．仕事をするのはかれらではないからだ．TQM は個人の味方であり，その意味では，個人主義的な傾向の強い西欧社会に向いているといえよう．

(3) TQM は決して創造的な人びとへの呪いではない： プロセスのドキュメンテーションや追跡は，適当なツールを用いさえすれば，そんなにコストがかかるものではない．もしそうでないなら，そのプロセスをまず改善しなければならない．

プロセスの計画をあらかじめ書き下しそれにしたがって仕事をするのは時間のムダだといっている人びとは，効率的な計画の立て方を知らないのである．書きあらわせないような計画，またはしたがえないような計画にはどこか悪いところがあるのだ．計画プロセスそれ自体もまた改善の対象である．

世の中には，自分の持っている能力は外からは見えないと主張する人びともいる．かれらは，そうした能力を検証できるような組織には住みにくいであろう．私が知っている多くの有能な人々は，TQM を愛している．なぜなら，TQM のおかげで，みんながこれまでのようなつまらない過ちを犯さなくなったので，その尻拭いの仕事が減り，より創造的な仕事に没頭できるからである．

体験レポート「揺れる雇用」

大場 充
(ソフトウェア工学研究所)

1. ごあいさつ

「リストラ」の大場です．去る1月31日で19年と10ヶ月間勤務した「ビッグ･ブルー」をやめました．2月1日からフリーの身となり，先輩の松原さんの前例にならって，当分のあいだ，世間的には「ソフトウェア工学研究所」を名乗ることにしています．

残念ながら「定年退職」ではなく，「自己都合」による円満退職です．

これにはもっと「深いワケ」があります．最近では，新聞紙上でも時々報道されていますように，米国IBMだけでなく，「遅まきながら」日本IBMでも，リストラのための「人員削減」を実施しています．せっかくの機会なので，私もそのプログラムを「体験」したいと思い，応募した結果，幸運にも「早期退職」が認められました．

アメリカと違って，日本では，従業員の「年齢枠」を指定した人員削減プログラムが違法ではありません．そのため，日本IBMが昨年導入した最初の「退職奨励プログラム」は，50歳以上の社員だけを対象とした「高齢者向けプログラム」でした．幸か不幸か，世間でも社内でもそのプログラムは評判が悪く，社員に対する心理的な効果は別として，予想したほどの効果はありませんでした．

私の勝手な分析をいえば，「日本IBMでは，7,000～10,000人程度の過剰人員を抱えており，早急に，少なくとも5,000人は削減しなければならない」というのが，常識的なベースラインです．去年の「高齢者向け退職奨励プログラム」では，1,000人を超える人が退職したと聞いていますが，それは，問題を解決するレベルから考えると「ほど遠い」ものです．そこで，特に退職者の年齢を規定しないプログラムが，昨年暮れを前にして発表されました．

個人的な感想をいえば，最初のプログラムよりも「公正」でよいプログラムだったと思います．今回の条件は，「会社が命じた転勤に，個人的な理由で応じられない社員」というもので，組織間や事業所間の移動に伴うものでした．典型的な例は，「今後の発展が見込まれない工場や開発の技術者を，営業へ移動させ，その過程で問題のあるものは切り捨てる」ものです．

しかしこの超長期不況の中では，「寄らば大樹の蔭」ということを皆が実感しています．多少の無理難題であれば，社員は「会社の意」にしたがいます．つまり，多くはやめません．そこで，プログラムの運用に柔軟性をもたせ，「会社が認めた者であれば，だれでも有利な条件で退職できる」ようにしたというのが実態です．それだけでは十分な効果は期待できないので，「いますぐ退職すれば，あなたはこれだけの退職奨励金がもらえます」という文書を従業員宛に作成しました．その文書の配布の責任は，現場の管理者に委譲されたので，全員に配布した管理者と，そうでなかった管理者がいたそうです．

実は昔，入社後5年目ぐらいのとき，日本IBMの人口構成を知って驚きました．私より1歳年長の世代をピークに，きれいな「山」を形成していました．それを見て最初に考えたことは，「これでわれわれの退職金は出るのだろうか?」という疑問でした．次に考えたのは，「これだけの人たちが数年間毎年何千人というペース退職して行くとき，会社はビジネスを継続してゆけるのだろうか?」でした．そのときから，私は40代での転職を考え始めました．

よく考えて見ると，同じことが国のレベルでも起こります．私たちよりも1歳年長の世代をピークに，人口構成の山が形成されています．このことから，2つのことが数学的に推論されます．第1は，「われわれの世代で国の年金システムは崩壊する」ことです．第2は，「定年退職後の再就職は，競争激化のためますます困難になる」ことです．そのときの私の解決策は，個人年金への加入と，再就職のリスクを分散するための「45歳選択定年制」の会社への提案でした．

個人年金への加盟は，個人的なことなのですぐに実施しました．当時としては法外な金額を毎月給与から差し引かれていましたが，給与の上昇率も高かったので数年後には負担にならなくなりました．と同時に，年金絶対額に変化はなかったものの，インフレによる受給額の目減りは大きいものでした．10年後に日本IBMが，主としてわれわれの世代を対象に，遅まきながら団体として個人年金システムを導入したとき，結局その年金にも加入しなければならなくなっていました．「世の中そんなに甘くない」ことを思い知らされました．

「45歳選択定年制」の方は，そうは行きませんでした．部門の長などに機会があるたびに必要性を説明しましたが，労働省を中心として社会全体が「定年延長」を目指していたこともあり，聞いてもらえませんでした．この私の持論を聞いてくれたのは，以前の上司のL.A.ベラディさんと，そしてほとんど同じころ日本に滞在していた米国IBMのGというマーケティング･アナリストだけでした．私の話を聞いたベラディさんは，「ドラッカーの本を読んでみなさい，君がいまいったのと同じような分析が書かれている」と，1冊の古い「IBM本社図書館」のスタンプの押されたの本を貸してくれました．

ともかく，今回の早期退職プログラムの「お金に目がくらんだ」私は，所属していた研究所の所長さんの積極的な勧誘に，「前向きな検討」を約束し，すぐに「応募の意志」を表明しました．所長さんも「ノルマ」を課せられて大変でしょうし，「遅

かれ早かれ退職するのであれば，条件のよいときに退職するのが一番」だと考えたのです．前回の早期退職奨励プログラムでは，「有資格者」(50歳以上のベテラン) 6名のうち，実際に退職したのは2名でした．今回のプログラムでは，私以外にも1人ベテランの退職者がいたので，部門としては「ノルマ」を達成できたと思います．

アメリカ滞在中，同僚と雑談をしているときなど，時々「早期退職」の話が出ました．当時，日本IBMは景気がよく，そんなプログラムはありませんでした．ある1人の同僚が，職場のミーティングの最中，「お前は辞めないのか?」と訊いたので，「日本にはそんなプログラムはないよ，残念だけれど....」というと，若いマネージャが「おいおい，もっと楽しい話をしようよ」と割って入ったりしました．そのころから，自分では「チャンスがあれば辞めよう」と決めていました．

こんな経緯で，24ヶ月分の給料を前払いしてもらい，「会社を辞める」ことになりました．もちろん，成算なしに辞めるわけではありません．ちなみに，新聞報道によれば，今回の特別早期退職プログラムに応募して退職した日本IBMの社員は，約1,500人だったそうです．そのうち，50歳以上の応募は約300人．つまり，1,000人以上が，私と同様に50歳未満だったことになります．前回のプログラムもあわせて，退職者の総数は3,000人を少し下回りました．

2. 私の「IBMよさようなら」

アメリカ滞在中に，10年来の友人が，35年間のIBMでの生活にピリオドを打ち，退職しました．実は，その友人が退職を決意した日に，私はかれの大きなオフィスを訪ね，ある国際会議でのパネル討論に参加してくれるように頼んだのです．かれは一瞬躊躇しましたが，「OK，ミツ」といって快諾してくれました．「ミツ」は，私のアメリカでの呼び名です．ちなみに，免許証の名前もそれで通しました．

次の日，かれの秘書から「エドが辞めるからパーティーをしよう」といわれて，かれのオフィスへ飛び込みました．

「エド，辞めるんだってね? コリーンから聞いたよ」
「そうなんだ．昨日はいえなくて悪かった」
「それでなんだけど，辞めた後でもラーレーの会議に来られるの?」
「ミツ，お前も俺が一瞬口ごもったのに気づいたろう」
「ああ，何かあるとは思ったよ」
「でも大丈夫，金は何とか工面するさ．おれもコンサルタントとして生きなきゃいかんし，投資だよ」
「ありがとう，エド．ボクが助けられることがあったらいってくれ」
「ありがとう，ミツ」

その会話からほぼ一カ月が経ったころ，私とその友人は，かれの新車「サターン」に乗っていました．

「いい車だね，エド」
「俺もそう思うよ，ミツ．前のシェビー・カバリエはずいぶん乗ったからなあ」
「でも，高そうな車だなあ」
「安くはないよ．でも，最近のアメリカ車，特にこのサターンは，品質もよくなっているし，値段も日本車よりも2割は安いよ」
「そうだね．ボクにも日本車は買えなくなってる」
「これが俺の人生最後の贅沢さ」
「そんなことあるわけないよ，エド」
「ミツ，離婚っていうのも，こんなものなのだろうなあ．長い間連れ添って，でも，今日で全部終わりだ」
「そうかも知れないね」
「今日はどうもありがとう，俺のIBM最後の日に昼飯をおごってくれて．死ぬまで忘れないよ」
「元気でね，エド」
「ありがとう」

そのとき，私は，自分の1年半後の「IBM最後の日」がどんな日になるか想像もしませんでした．

1月30日，日曜日，晴れ，昨日の雪がまだあちらこちらに残っています．以前の私の上司であり，いまは三菱電機ボストン研究所の所長をしているベラディさんと，銀座で食事をしました．当時，同じグループのメンバーだったAさんも一緒でした．12時の待ち合わせでしたが，レストランを調べるために，30分ほど早く，ベラディさんが泊まっている八重洲富士屋ホテルに到着．近くの日本料理の店を何軒か，インフォメーションで訊いてから，Aさんと一緒にロビーで待ちました．

ベラディさんは私と違って，あまりカジュアルな服装をしません．ドレスコードについて相談しておかなかったことを悔やみながら，「でも，ベラディさんはスーツとネクタイに違いない」と予想し，私はスーツとネクタイで着てきました．ところが，Aさんはセーターにボサボサ髪です．もっと驚いたのは，ベラディさんがジーンズで来たことでした．私が「あなたもジーンズを着るんですね!」というと，「着替えた方がいいかい?」と訊いてました．私が気分を害したとでも思ったのでしょう．「そんなバカな!」と答えました．本心をいえば，「ジーパンとブーツで来ればよかった」と後悔していたのです．

ベラディさんは歩きながら，「昨夜，SEAの岸田さん，熊谷さん)，山崎さんたちとこの近くで食事をしたとき，明日が君のIBM最後の日だと教えておいたよ」といいました．

「結局，何年勤めたのかね?」
「19年，もうすぐ20年です」
「あんまり長くはないなあ」
「あなたと比べればの話です」
「まあ，潮時ということだね」
「そうです」
「ところで，昨夜は"うどんすき"をごちそうになったのだけど，あれは外来語かい?」
「え??」

「ウドンスキーはロシア人の名前に似ているから」
何というダジャレ!

3人の天麩羅屋での会話です．私の将来に関する話が一段落したとき．

「Aさん，これからどうするの?」
「当分は，いままでのプロジェクトを継続して，技術移転に全力を注ぎます」
「その先は?」
「大学へ移りたいと思っています」
「どこかアテでもあるの?」
「ハッキリとは決まっていませんが....」
「大事なことは，この手の問題は，シーケンシャルには処理できないってことだよ．何かをしながら，将来どうするかを決めて行かなければならない」

まさにその通り．「亀の甲より年の功」とはよくいったものです．日本人にはこれができない．というより，慣れていないので分かっていないのでしょう．

「ところで，私の友だちは皆引退してしまったよ．知っているだろう?」
「そうですね．ワッツ(ハンフリー)も事実上はCMU/SEIを引退したようですし，あなたがポケプシーで会った人たちの多くが辞めて行きました」
「そうなんだ．ワッツの堅物も辞めたからなあ．でもあいつは70歳を過ぎたんじゃないか?」
「年までは知りません．でも,"柔軟でない"ことはたしかですね」
「私もそろそろ引退するべきかね? どう思う?」
一瞬，「ベラディさんのところもこの不況で苦しくなているのかなあ?」という考えが脳裏を過りました．
「引退してもいいと思います」
「ところで，電子英和辞典はどうかな? すでに商品化されているものがあるかね?」
この人，ほんとうに引退するつもりなんてあるのだろうか?

1月31日，月曜日，快晴．私の「IBM生活最後の日」です．昼食を，妻の智津子と一緒に近くの中華料理屋で取ってから，研究所の同僚たちに配るマロングラッセを買い，会社へ向かう．片道2時間15分の通勤も，今日が最後．それにしても，何という時間のムダなんだろう．新聞や本を読んだり，携帯ワープロで原稿を書けたとても，何てムダだったんだろうと思う．それも，今日が最後．電子手帳で，今日会社でやるべきことのチェックリストを作っているうちに，幕張に到着してしまいました．

最初に，アメリカ滞在中からいろいろお世話になった健康管理室へ，チョコレート持参であいさつに．

「今日で会社を辞めます．いろいろお世話になりました」

2人の顔見知りのアルバイトの女性たちが出てきました．声をそろえて，

「ええー，本当ですか?」
「まあ，こういうことになりました．ありがとうございました．これ，つまらないものですけれど皆さんでどうぞ」
「そんな気を使っていただかなくても.... でも，これからどうされるんですか?」
「当分は，家でブラブラします．では，これで....」
「それじゃ，お元気でどうぞ.」

エレベータで最上階のオフィスへ．研究所の人数も減った関係で，部門のスペースが半分になり，新しい部門が残りの半分のスペースへ引っ越して来ていました．荷物の開梱で辺りは騒然としています．オフィスの入り口で，一緒に退職となった先輩に出会い，挨拶をすると，

「今日は来ないのかと思っていました」
「来ますよ．この前電話でいったじゃないですか．帰りに，一緒に乾杯しようって」
「僕は，もうすぐ帰るよ」
「ちょっと待っててください．4時半には終わります」
「それじゃ，下の喫茶室へ行ってますから.」

自分の席で机の中身を梱包していると，所長さんが来て，「郵便物が来てます」というので，「後で取っておきます」と答える．

「返さなければならないものがありますから後でうかがいます」
「待ってます」

きわめて事務的な会話である．隣りの席で働いていたAさんが，「まさか今日が最後で，もう来ないってことはないでしょう?」という．

「少なくともう1回は来なければだめですね」
「そういわずに，10回くらい来たら?」

ありがたいけど，そういうわけには行かないようです．

自分の机を片付けた後，借りていたノートパソコンと書類の束，鍵類，身分証明書を持って，所長の部屋へ行きました．いつも忙しそうにしている人です．事務的に，「ノート(パソコン)，鍵，身分証明書，健康保健証をお返しします．これらは退職に関する書類ですが，私が関係部門に送りましょうか?」と訊くと，「お願いします」という．いくつかの書類に判を押し，サインをして，「ありがとうございました．お世話になりました」と最後の挨拶．「こちらこそ」という言葉の響きが妙にそらぞらしい．

席に戻って，もって来たマロングラッセの箱を開けました．Aさんの席へ行き，「お世話になりました．お一つどうぞ」と勧める．

「俺には，これでは足りないよ」
「だから昨日，お昼をおごったでしょう」.

同僚の席を順番に廻って，同じ挨拶を繰り返す．こちらは，同じことばかりいっているので，だんだん簡単になって行くのが自分でも分かる．「ちょっと手抜きかな?」と心の中で反省．

1階下のフロアーには，子会社の人たちがいます．1月1日でその子会社に移った，去年までの仕事仲間を訪ねました．運がよいことに，去年の7月にその子会社へ移った先代の所長さんもかれの席に来ていました．まずは型どうりのご挨拶．「大場さんは，運がよかったね．選択は正しかったと思うよ．これからは厳しくなるだろうね」なんとなく本心で話しかけてもらっているようで安心しました．「まだ，こんな人が残っているんだ」と心の中で感心してしまいました．「明日から人事制度が変わるそうだよ．成績評価や退職金の計算方法も変わるみたいだ」「年俸制になると聞きましたけど....」．これも時代の流れかと感じました．

先代の所長さんは立派な人です．現在の日本IBMの社長が就任したとき，「コストの削減と顧客満足度の向上が急務です」と挨拶しました．これに対して「それが社長になって最初にいうべきことか?」といったそうです．まさに同感です．その人はシステム·エンジニアとしては，社長さんの先輩です．よいSEであることと，よい管理者であることとは違うようです．「部下の首を切って，自分が安泰というのではスジが通らない」といって，日本IBMを退職され，昔の同僚が社長をしている子会社へ転籍した人です．私とは，何人かの共通のアメリカ人の友人があります．かれらからも尊敬されています．

そんな共通の友人の1人に，ハーバードで日本文化を専攻した開発部門のBという管理者がいます．かれの部門は，IBMが独自に開発したシステム·ソフトウェア開発用のオブジェクト指向言語で，オペレーティング·システムの一部分を開発しました．私は，オブジェクト指向言語がOS の開発にどこまで有効かについて興味を持っていたので，かれの話を何回か聞きに行ったことがありました．

アメリカから帰ってくる直前に，「ぜひ一度，昔のボスのベラディさんにOBとして講演してもらおう」と思い，IBMの社内セミナー招待しました．ベラディさんは「昼飯をごちそうする」というだけの，破格の条件で来て下さいました．そのセミナーの案内状を電子メールで流しとき，最初に返事をくれ，わざわざ私のオフィスを訪ねてくれたのがBでした．日本文化について意見を交換しているうちに，日本IBMの話に発展し，「お前の日本のボスは，私の友だちだ」ということが明らかになりました．「あの人は，立派な人だ」というのがかれの結論でした．

自分の席へ戻って，いよいよ例の先輩と乾杯に出ようと帰り支度をしていると，かれが私の席に来て，「今日は家族とお祝いの夕食を約束してますので，もう帰ります」という．「そういうことであれば，また機会にしましょう．では，お元気で」といって別れました．あとで，かれの席の前を通ると，机の上に愛用のブリーフケースが置かれたままになっていました．それを見て，複雑な気持ちになりました．早期定年退職する人と，私のように勝手に退職する者では全然違うのかもしれないと感じました．

その先輩には，まだ小学生の低学年の長男がいます．「長男が大学を卒業するまでは働かなくちゃ」とよくいっていました．そういったまじめな人の退職の決断と，もうすぐ独立心の強い長女が高校を卒業し，「みんな自分の人生なんだから，最悪の場合は自分で稼いで学校へゆくべきだ」と考える私とでは，将来に対する危機感が違うようです．そんな状況の私にとっては，退職は「自分の(人生)プロジェクト」の通過点の一つにすぎないのですが，ある人びとにとって，退職は「最初の(自分の)プロジェクト」の終わりを意味するものかも知れません．新しいプロジェクトを始めるのは容易ではなく，それなりのリスクを伴います．そのような複雑な状況があるのではないかと考えてしまいました．考えすぎだったかも知れませんが....

そんなわけで，5時前にすべてが終わってしまい，することもないので帰ることにしました．ロビーを出て外を見ると，夕焼けに幕張のビル群の黒い陰がくっきりと写っていました．とてもきれいではありましたが，人工的で無味乾燥な景色でした．「今日でこの景色ともお別れだな」と思いながらビルを後にしました．「片道2時間15分の通勤からもこれで解放されるのか」と思うと，何となくうれしく感じると同時に，「何と感動のない退職なのだろう．他の人もこうなのだろうか?」と考えてしまいました．

こんなわけで，私のIBM最後の日は，予期していたものとまったく違って，「門出を祝ってくれた」のは，何と以前の上司だけだったという寂しい結果になってしまいました．

3. 失業して何を学んだか?

日本IBMをやめておよそ2週間経った2月16日に，会社から「離職票」というのか，とにかく失業を証明する1枚の紙が自宅に届きました．それを待っていた私は，さっそく職業安定所(最近は「ハローワーク」と呼ばれています)へ電話をして，場所を訊き，車で駆けつけました．その「職安」は，どういうわけか日立·戸塚工場の近くにありました．最初からそう説明してくれれば簡単だったものの，戸塚駅からの道順を説明してくれたので，ただただ，いわれた通りに走っていきました．しかし，曲がり方を間違えたり，交差点を通り越したりで，到着したのは4時半ぐらいになってしまいました．受付はまだ開いていましたが，「明日また来るように」といわれてしまいました．

次の日，今度は朝早く(とはいっても9時半ごろ)行きました．受付に書類を出すと，簡単にチェックをして，「3番の窓口に書類を提出して待ちなさい」とのこと．何事も命令口調の職員の応対ぶりに，少々カチンときました．言葉の端はしに，「お前らは半人前だ」といっているようなニュアンスが感じられます．さすがに，定年退職者に対してはもう少していねいでした．窓口に呼ばれて行くと，今度は若い女性の職員がていねいに，「月に数回，大学に行くと書いてありますが....?」と尋ねます．「ええ，前の会社でやっていた共同研究の続きがあるもので」と説明すると，「それでは，新しい就職口を紹介できないので，そのような活動は土日を使うようにできますか?」と訊きます．「いえ，実験をやっているので，土日というのは少し難しいのです

が....」と答えると,「それでは,働く意志がないとみなします」と断言されたのには驚きました.

「働く意志がなければ失業保険は出ませんよ」と脅かされ,「それではしかたがありません」とシブシブ条件を受け入れることにしました.回りを見渡すと,独身らしい若い男女でにぎわっています.「これでは多少高飛車に出てくるものしようがないか」と納得してしまいました.事務はかなり機械化されていて,OCRに書き込むと,データベースに登録されるようになっています.OCRカードの作成を終わった女性の職員が席を立って,後ろの読取り装置の所へ行き,カードを入れるとディスプレに入力されたデータが表示されました.ディスプレのデータを見て,かの女がキーボードをたたきます.OCRの読取りエラーを修正しているのです.「最初からキーボードをたたいた方が速いのでは?」と思いながらも,うかつなことはいえないので,黙って見ていました.席に戻ってきた女性は,「これを持って2階の雇用保険の窓口に行ってください」と今度はていねいにいいました.

雇用保険の窓口に行って書類を提出し,名前を呼ばれるのを待ちます.暇を持て余して見回すと,奥のほうで偉そうに座っているオジサンがヒマそうにしています.そのオジサンの右,少し離れた所に案内係のオバサンが座って居眠りをしています.保険窓口では,2人の職員が若い人の応対をしています.その左の窓口では,若い男性の職員が,書類を片付けていました.私があまりにジロジロ見ているのに気がついたオジサン職員は,すうっと立ち上がり,書類を片付けていた職員に向かって,「待っている人がいるから,終わったら応対するように」と小声で指示して,部屋を出て行きました.その職員はすぐに,「大場さん」と私を呼びました.

私が席に着くと,その職員は私の書類を見渡してから,新しい書類を出して私の名前やら勤めていた会社の名前などを転記します.「こんな処理こそコンピュータでやればいいのに」と思いながら見ていました.転記が終わるとその書類を私の方へ差し出し,「一番下の欄に退職の理由を書いて下さい」とぶっきらぼうな口調でいいます.「はい」とはいったもののそんなに直ぐには書けません.少し考えて,何を書くべきかを決めました.職員はその間,私の提出した書類に目を通していました.3分間ほど考えて,3行書きました.

「11月に上司から早期退職プログラムに応募するよう勧誘を受けました.アメリカ滞在中に,早期退職プログラムは早いほど条件がよいことを知ったので,応募するなら第1回目に応募しようと決めていました.それで,今回の勧誘に同意しました」

書かれた私の文章を読んで,その若い職員は少し驚いた様子でした.少し間を置いて,「退職奨励があったのですね」と訊くので,「ありました」と答えると,「離職票の理由の項には,"自己都合"とありますが...」と尋ねます.文脈をまったく理解できない私は,何と答えてよいか分かりませんでした.すると,「辞表を書いたのですね」と尋ねます.ここまでいわれて,私は「ハメられた!」と心の中で叫んでしまいました.退職する2週間前の上司とのやり取りが,3倍速のビデオのように頭の中に映し出されていました.

「大場さんちょっと」と所長が私を呼ぶ.「はい」といって席を立ち,所長室へ行く.部屋へ入るや否や,「辞表を書いて下さい」という.「どんな書式でも結構です」とケゲンな顔の私に取ってつけたようにいう.「ハア,そうですか?」といって部屋を出る.よく考えると,2人の部下の退職を世話したが,「辞表を出してくれ」と自分が頼んだ記憶はない.不審に思ったが,忙しく考える余裕もなかったので,「とにかく何か書いて出そう」と決心する.家へ帰ってワープロで,辞表のようなものを書いた.

「SE研究所長殿.過日,口頭にてお応えしましたように,今回の早期退職プログラムに応募し,1月31日付で退職したいと考えます.以上,よろしくお願いします」

数日して,会社に行ったとき,会社で使っている印を押して所長に提出する.10分ぐらいして所長が来て,「三文判でもいいから,自分の印鑑を押して出して下さい」という.「それでは,今度来るときに認印を持って来ます」と約束する.また数日して,認印を押したものを持って行くと,「人事と相談したんですけど,これは少しマズイ.宛先を社長にして,この例の通りの文面にして下さい」という.その文面には,「一身上の都合によりx月x日付けで退職したく」とある.「やはり」とは思ったものの,「いまさら騒ぎを大きくすることもないか」と考え,例文をコピーして名前を書き,判を押して提出.後知恵だが,「書く必要はなかった」と反省する.

若い職員は,「退職奨励は会社都合の退職です」と説明します.「やはりそうか」と後悔しても,もうどうにもなりません.「だめでもともと」と思いながらも,一応いきさつを説明しましたが,結果は「辞表を書いたのでしたら,ここに"自己都合了承済み"と書いて下さい」とのことでした.さらに,「自己都合退職の場合,3ヶ月間は失業保険は支給されません」との宣告をされる.そして,「2月28日の説明会に来て下さい」といわれ,説明資料を渡されました.「最近は,1週間で出ると聞いたのに」と思ってもどうにもなりません.しかし,これはまだ「序の口」でした.

家へ帰って妻に結果を説明すると,「就職支度金を貰えるから大丈夫」と慰めてくれました.そこでパンフレットをよく読むと,「失業保険の支払い期間の半分以上を残して一定の条件を満足する形式(たとえば"退職した会社へ再就職する"ことなど)で再就職した場合には,残日数に応じた就職支度金が支払われる」とあります.計算してみると,うまく行けば90万円ぐらいもらえそうです.そんなわけで「就職支度金」が最後の希望になって来ました.

ところで,パンフレットをよく読んでみると,再就職には「自己就職」と「安定所紹介の就職」の区別があることが分かりました.「自己就職」とは,「自分で就職先を探して就職すること」

をいいます．職業安定所では，「安定所の紹介での再就職を奨励している」ようです．もうひとつ，「職業安定所が認めた場合，引っ越し手当を支給する」とありました．いろいろ期待が持てそうでした．

2月28日，期待を胸に「説明会」へ向かいました．少し早めに到着しましたが，駐車場は車で一杯でした．会場へ行くと，100席分ぐらいの椅子が並べられており，すでに何人かの人たちが受付手続きをしていました．訳も分からず，とりあえず行列の最後尾に並び順番を待ちます．銀行への払い込み依頼書，預金通帳をチェックして，何かカードのようなものを配っていたのでした．カードには，姓名，生年月日，離職した日，登録日，離職理由，雇用保険の支払い期限，支払い日数，加入年月と加入期間，支払額などが印刷されていました．説明会が始まる時間までには，会場は一杯になりました．ほとんどが若い女性たちでした．中にぽつぽつと，私のようなオジサンが「場違い」というような感じで座っていました．私の左隣に座った人は同年代の紳士，右となりは定年退職したと思われる老紳士でした．

説明は，まず「離職理由」から始まりました．「会社都合」と「定年退職」(選択定年を含む)は，「会社も認めた退職として扱われる」という意味の説明でした．「いろいろな事情で"自己都合退職"と認定された方々は，必要があれば再度調査しますが，いまのところ形式上は"本人の問題による"退職として取り扱われます」とのことでした．「したがって，3ヶ月間の期間をおいても失業状態が続いていると認められるときに限り，失業保険が支給されます」と，説明は続きます．ところで，「この3ヶ月の期間中は，一時的に職についてもかまいません」という注意がありました．つまり，この3ヶ月間は，職業安定所の感知しない期間のようです．皆さん，「辞表を書いたら，半年ぐらいは遊んで暮らしましょう!」

もらったカードの上に何か赤インクでスタンプが押されていました．よく読むと，「再就職の場合の支度金(一時金)は，自己都合退職の場合でかつ自己就職の場合，待機期間が2ヶ月以上過ぎなければ支払われません」とある．つまり「自己都合退職し」かつ「自己就職する」場合には，離職から少なくとも3ヶ月以上の期間を置かなければ，何の保証もないのです．たとえ，それが「会社のリストラに協力した早期退職でも」です．私の場合，4月に再就職が予定されています．つまり，「20年間で約600万円を投資して，その見返りはゼロ」でした．ある程度の「見返り」が期待できるとすれば，それは私が「悪事を働いて，次の職場を数年でクビになった」場合ぐらいでしょう．確率はゼロではないにしても，きわめて小さいと思われます．

後知恵ですが，「昨年の12月31日で退職しておけばよかった」と後悔しています．1月分の給与などと差引すると，20万円以上の機会損失が出ました．アーア，残念．

4. 労働行政と定年退職制度

自分自身を「リストラの波」の中に投じて学んだことを，少しまじめに考察してみます．多分「このようなこと」は，これからわが業界ではかなり普通に起こることだろうと考えます．そんなわけで，ここで問題を整理しておくことも重要だと考えます．みなさんは，「これはきわめて特殊な例で，私自身の参考にはならない」と思って読まれているかも知れません．ほんとうにそうでしょうか．ポスト資本主義の「知識主義社会」では，「雇用は流動化するであろう」というのが大方の予想です．多分，「失業率10%はふつうのこと」になるでしょう．重要なのは，「失業の形態が大きく変わる」ことです．つまり，現在の日本でよく見られるような「パーマネントな」失業ではなく，「一時的な」失業が増えるのです．

極端ないい方をすれば，「プロジェクトが終わったら，また失業する」ようになることが予想されています．プロジェクトごとに必要な専門知識が変わるからです．専門知識を1から学び直すほどの時間的･経済的な余裕は，企業にはなくなります．つまり，「将来役に立つか立たないか予測できない専門知識を持ったエキスパートを，高い給料を支払ってその時が来るまで雇っておく必要はない」のです．別のいい方をすれば，「A社には必要のなくなった専門知識も，B社の新しいプロジェクトの成否を左右する」こともありえます．貴重な知識資源を社会全体として効率よく配分して行くためには，専門家雇用の流動化が必要なのです．専門家雇用が硬直化した国の経済は競争力を失い，衰退するでしょう．

そのような観点で現状を見ると，わが国の労働行政は転機にあるといえるでしょう．これまでの「高度な専門知識を必要としない労働力」を念頭においた古いスタイルの労働行政から，「高度な専門知識を必要とする労働力」も視野に入れた新しいスタイルの「複眼的な」労働行政にシフトしなければならなくなっているのです．特に，このことは，わが国ソフトウェア産業の国際競争力を維持し，高めるために重要です．はっきりいえば，現在のわが国の労働行政は，「工業化時代」の枠組みにもとづいており，これからの「サービス化時代」の枠組みには合致しません．工業化時代(昭和30年代)の「雇用安定化政策」と，いまわれわれがまさに移行しようとしているサービス化時代の「ダイナミックな雇用」は，相反するものだといえるでしょう．

そのような問題の具体例の一つが，「定年退職制度」です．私の理解では，その主旨は「安定した雇用関係を国家が推奨する」ことです．「教育を終えて25年以上，50歳以上)になるまで同じ会社で働く」ことを理想とした「定年退職制度」は，価値生産の主体が資本にあり，労働者の知識になかった時代に，日本に伝統的な「家」や「村」の概念を「資本」に適用したモデルとして，それなりの意味を持っていました．しかし，いまや価値生産の主体は，資本から労働者の知識へと急速にシフトしつつあります．この急激な環境の変化で，従来はうまく機能していたモデルが，構造疲労を起こし，機能しなくなりつつあります．極論すれば，「企業に何年所属しているか」と「その人が生産した価値の総和」は，まったく線形な(比例)関係ではなく

なって来ているのです.

「ある企業に50歳以上になるまで止まる」ことは，その人の知識を「その企業向けに特化させる」結果になります．また，「50歳以上で，特定の企業の仕事しか知らない人(特に管理者)の持つ知識の価値(効用)は，他の企業にとっては意味のないもの」になり，その人の知識を労働市場で売ることが困難になります．いまの状態では，「知識労働者は特定の企業のために全力で働き，定年後の退職金と厚生年金で余生を送るしかない」働きバチのようなものです．機械との違いはあるのでしょうか．経営者から見れば，多分「何も違いはない」のではないでしょうか．

定年退職制度は，「税金」とも関係しています．定年退職か自己都合退職かでは，税率が全く違います．厳密には，控除額が大きく違っているからです．私は，所得税と地方税を合わせて，約800万円を退職金から税金として源泉徴収されました．減税の財源に困っている大蔵省や細川総理に多大な献金をしてしまいました．所得税は，500万円を超えていました．ふだん，数十万円の給料で生活しているわれわれが，ほぼ1年分弱の生活費に相当する税金を取られたのでは，冷静でいられる訳はありません．いくら多額の退職金をもらえるとはいえ，税金でその約20%が持って行かれるのでは，余程の理由がなければ定年前に「退職しよう」とは考えないでしょう．50歳まであと5年弱，少しぐらい邪魔モノ扱いされても，しがみついていたくなります．たとえ，自分の市場価値が下がったとしてもです．

定年退職制度は，「失業保険」とも強い関係をもつ問題です．現在の雇用保険システムでは，「被雇用者を失業による経済問題から守る」という目的と，「雇用者に対して安定した労働力を供給する」という目的があるようです．この2つの目的を同時に達成するために，現在の雇用保険システムでは，「定年退職」と「会社都合の退職」を優先し，「自己都合退職」に対してはペナルティを課す方法を採用しています．

定年退職以外の退職については，実際には「論理的な基準」がないため，その判断は実際には「運用に任せられている」状態です．つまり，雇用者と被雇用者間の「力関係」で決まります．「会社都合の退職」に対してもペナルティがあるらしく，企業は「会社都合」を極力避けているのが現状です．この矛盾が明らかになるのが「リストラ」による「人員整理」です．雇用者にいわせれば，「社員個人の自由意志で退職したのだから自己退職である」とのいい分になります．被雇用者にいわせれば，「会社が退職を勧告したのだから会社都合である」とのいい分になります．

定年退職制度は，「両刃の剣」です．退職者の年齢によって，定年かそうでないかを判断することは，極端にいえば，10年でも定年でありえますし，25年でも達しないこともあります．その意味では，必ずしも「安定した労働力の供給」を達成するとは限りませんし，50歳以上の人にとっては，「安定した職業を保証するシステム」であるともいい切れません．また，定年制度があることによって，ある意味では，雇用者にとっては「50歳以上の従業員を円満に解雇する手段」が用意されていることになります．そのようにして解雇された人々の生活費は，「雇用保険」で最低限保証されているからです．

私は，このような定年制は「年齢による差別」を助長する結果となりやすく，変えなければならないと考えます．むしろアメリカのように，「年齢による差別」を禁止した方がよいと思っています．考える力は，年齢に比例するわけではなく，個人差も大きいからです．専門家としては，自分の能力で勝負すべきであり，企業としては，専門家の能力を買うべきでしょう．「終身雇用制」が崩れつつある今日の状況で，「定年制」はその功よりも罪の方が目立つようになって来ると予想します．

現在の日本の労働行政の次の問題は，「職業安定所」だと思います．職業安定に特別なサービスを期待しているソフトウェア技術者は少ないでしょう．他の分野の専門家たちも同じだと思います．失業したら，みなさんも私と同様に，「自分自身の力」で新しい職場を探すのではないでしょうか．専門家としての仕事の紹介を職業安定所に期待することは非現実的です．むしろ，ヘッドハンティング会社のような人材サービスの専門会社に相談する方が現実的です．

専門家に職を紹介するためには，専門家の持っている知識や経験を的確に分析･評価する能力が必要になります．また，企業に専門家を紹介するためには，企業がどのような知識や経験を必要としているのかを分析し，そのニーズに合った人材を紹介しなければなりません．専門家にたいしても，その会社にとって，専門家の持つどんな知識や経験が必要なのかを，説明できなければなりません．これは，高度な専門サービスであり，一種のコンサルタント業務です．

現在の職業安定所には，そのようなノウハウもシステムありません．資本が価値生産の主体である業種に対して，経験や知識を必要としない職種を紹介するには，紹介する人の詳細な能力や経験を評価する必要はありません．賃金と業務内容がその人の意向と合えば，それで終わりです．人材を要求している企業にしても，「いま，何が問題か?」を説明する必要はなく，必要な人の数と採用の条件さえ明確にできればよいのです．つまり，現在の職業安定所業務は特別なノウハウを必要としておらず，単なるデータベース検索のような事務処理が主体となっています．

20年前ならいざ知らず，コンピュータが普及したいま，ほんとうにこのような大きな役所が必要なのでしょうか．私は，コンピュータを導入すれば，現在の主要業務については，かなり小さな規模の役所(むしろ私企業に任せればよい)で十分になると思います．むしろ，「新しいサービス」に移行して行くべきだと思います．その「新しいサービス」が何かは，別の機会に述べさせていただきます．

「いまの仕事と再就職活動は同時進行しなければならない」．これが知識主義社会で専門家として働くものの宿命です．現在

従事している仕事をして行くと同時に，次の仕事の準備をしなければなりません．「この仕事が終わったら」というのでは遅すぎます．前の仕事と次の仕事の間にできる失業状態が長くなるからです．むしろ，そのときどきの仕事に関係なく，つね日頃から「就職活動」を心がけておくことが必要でしょう．「就職活動」とは，「学会に参加したり，SEAのような同業の技術者の集まり(つまり「ギルド」)に参加したり，いろいろな企業の人たちが集まる「勉強会」に参加したりして，自分を磨くとともに自分を宣伝する」ことをいいます．もちろん，人材サービス会社のデータベースに自分の履歴書を登録することも含みますが，ここではもっと積極的な活動を意味しています．

いままでは，長期間にわたり同じ会社に勤めることが一般的でした．また，資本が価値生産の源泉であったため，あまり個々の労働者の質が問題になることはありませんでした．簡単に人を入れ替えることができたのです．そのため，失業しても仕事さえあれば，すぐに新しい仕事につくことが可能でした．このようなことから，在職中に次の仕事を探すなどの再就職活動をする習慣が，私たちにはありません．ですから，そのようなことが特別なことのように感じられますが，アメリカの，特に専門的な職業に従事している人々は，以前からそのようにしています．個人の生活を第一に考えれば，当然の結果でしょう．その意味でも，現在の職業安定所は，専門家にとって意味のないシステムになっています．

最後に，「雇用保険」の問題について議論します．現在の雇用保険は，専門的な職業に従事しているソフトウェア技術者にとって，どんなメリットがあるのでしょう．私自身の例でいえば，「約20年間加入して，失業しても1円ももらえない」状況です．つまり，「掛け捨て保険」です．もちろんいまの失業状態が4ヶ月以上続けば，失業給付金を受け取れるはずですが，そんなに長期間にわたって浪人生活を送るわけにはいきません．幸いなことに，私はかなり前から準備をしていたので，2ヶ月の失業で済ませられる予定です．また，新しい職場で雇用保険を継続でき，定年退職後にはふつうの人と同じ程度の失業給付を戴けるようです．

もし，一般の保険会社で似たような保険をかければ，毎月の保険料は高くなるでしょうが，必要な時に自分の意思で「給付金の受給を請求する」ことや，「積み立ての継続を選択する」ことができるものになります．専門的な仕事に従事している人に対しては，もっと自由度を持たせるよう，民間の保険会社に任せた方がよいのではないかと思います．つまり，「専門技術者向け雇用保険」を別建てにして，個人の選択の余地を与える方がよいと思います．また，現在のシステムでは会社側も毎月，各従業員の月収に見合った分の保険料を負担しています．これは，むしろ各個人への給与として支払い，各個人がどれだけを失業保険として支払うべきかを決めればよいことだと思います．「保険料が低額である」という理由で，「いまのままでよい」とはいえません．システムをもっと簡素化して人件費を低減させるべきです．

5. お知らせ

個人的な事情で，4月から，居を広島市内に移すこととなりました．SEAの東京での会議への参加の機会は減ると予想していますが，その分「地方」で頑張りたいと考えております．これといった具体的なアイデアはまだありませんが，SEAのセミナー・フォーラム，シンポジウムなどのイベントも機会があれば，積極的に地元で開催したいと考えています．

もうひとつ，地方に来たからにはぜひやってみたいことに，「広島支部」の設立があります．残念ながら現在，広島市近辺のSEA会員数はまだ1桁のようです．この会員数を少なくとも，地元企業のソフトウェア技術者を中心に，1994年末までに20名以上にしたいと思っています．できれば，3年で50人前後の会員数にまで伸ばしたいと考えています．とはいえ，この不景気ではどうなることやら分かりません．

さらにもうひとつ，退職を機会にやってみたいことがあります．それは，ＳＥＡを母体とした「ギルド」の創設です．名前は仮に，「ソフトウェア・コンサルタント・ギルド」とします．オフィス・ピープルウェアの松原さん，エス・イー・ティー顧問(?)の天池さん，そして私と，最近「大樹の蔭」を離れる人が少しずつ増えています．このような独立コンサルタントの活動を支援する「非営利」の機関が必要です．東京から離れた広島でそのような活動をすることには，経済的にも不利ですが，トライしてみようと思っています．

具体的には，クライアントへのコンサルタントの紹介です．「人材銀行」のようなもので，クライアントの要求を聞いて，コンサルタントと共同でプロジェクト企画を作成し，クライアントに提案してゆくことが主たる活動になります．松原さんはもちろんのこと，天池さんも「有名人」なので，いま差し当たって必要というわけではありませんが，将来必要になると思っています．当分は，SEAを通じて知り合ったソフトウェア技術者諸兄との個人的なネットワークを活用して，活動してみるつもりです．お金の取れるプロジェクトだけでなく，人々のためになるボランティア・ベースでのプロジェクトも，考えております．

というわけで，みなさん，これからもよろしくお願いします．

(1994年3月8日)

書 評: "Things that Make Us Smart"

Donald A. Norman 著

本書は，University of California, San Diego 校の認知科学の教授から，最近，Apple 社の顧問に転職された Donald Norman 博士の最新の著作です．Norman 先生は，通産省が推進しているヒューマン･ファクタをテーマとした Friend21 プロジェクトの顧問としても招待されており，Human-Computer Interaction の分野におけるアメリカの第一人者です．

まえがきにも著者自身が記しているように，本書は，かれのシステム･デザインとヒューマン･ファクター関するシリーズの第 4 冊目にあたります．

シリーズ第1 冊目の "User-Centered System Design" では，なぜコンピュータシステムがこんなに使いにくいのかを解析しようとしてかれの研究室で行われた一連の実験研究結果が報告されていました．そこで得られた本質的な原因が，実は，コンピュータ･システムに限らず，ドアや蛇口など，わたしたちが日常何気なく使っているものにもあてはまることを述べたのが，それに続く "The Psychology of Everyday Things" です．

3 冊目の "Turn Signals Are the Facial Expressions of Automobiles" は，科学技術がもたらす社会的なインパクトが軽い調子で述べられていました．4 冊目の本書は，同じ話題を，よりシリアスな側面から深く切り込んでいますが，専門外の人にもわかりやすいスタイルで書かれています．

著者は，読み書きや言語，論理など，知的作業のための抽象的な仕組から，s 紙やペン，計算機などの物理的仕組までを含んで，人間の知性とか思考を支援するツールを，認知器械/人工物 (Cognitive Artifacts) と呼んでいます．本書の対象は，コンピュータ･システムをはじめとするそうした Cognitive Artifact のデザインと使われ方，社会的インパクトに関して論じたものです．

システム事故が発生する度に，しばしばユーザの人的ミスが原因とされ，人間側にその責任が押し付けられます．しかし，はたして一体そうなのでしょうか? 人間は，本来まちがえたりミスを犯したりするものです．そういう人間の過ちを許容できないようにデザインされたシステムの側にこそ，問題があるのではないか? というのが著者の問いかけです．すなわち著者は，科学技術そのものに対して懐疑的なのではなくて，科学技術の使われ方，または技術に対する歩み寄りの姿勢が，人間中心であるべきだという，人間賛歌の立場をとっています．これは，長年にわたって人間を対象とする研究を行ってきた認知科学者が出した結論として，とても興味深いものです．

本書では，人間中心の科学技術の用い方の原理として，精密で詳細な測量を要求するハード･サイエンスに加えて，観察や分類，主体的測量に頼るソフト･サイエンスの重要性，そして人間の思考として，経験的認知 (Experiential Cognition：無意識に状況に対して応対する能力) と，思慮的認知 (Reflective Cognition：理論づけで状況に対して応対する能力) の両方を支援する必要性などを，例を用いて述べています．

たとえば，データや情報などの「表現の仕方」一つをとっても，人間の認知能力よりも「技術」が優先されてしまっていることが，よくわかります．数の表現がそのよい例で，アラビア数字はたしかに大きな桁数を処理するためには便利ですが，直観的な足し算をするには，従来のローマ数字の方がずっと便利なのです (たとえば，11 + 12 = 23 と，XI + XII = XXIII を比べて見ると，ローマ数字では，単に記号を合わせるだけでよいことがわかります)．

また，人間が本来何気なく社会的に用いている習慣や規則なども，知らず知らずのうちに，大切な役割を果たしていることがよくあります．このように，人間の直感や常識，習慣，学習性などは，それが論理的にハード･サイエンスとして表わせないばかりに，科学技術の側面からは軽んじられてきました．その結果が，先に述べたような人間の避けられない過ちをを，人間の責任として責めるような土壌としてでき上がってきてしまったのです．

しかし，21 世紀に向けて，マシンには持ち得ないそうした人間固有の特性を活かすように，発想の転換を喚起することが，本書の目的です．このような発想を元にしてシステムを設計するには，たとえば，マシンにできること (affordance) を自明にしたような設計であるとか，一度思い込んだら，どのような現象もそのように見えてしまうという人間のトンネル･ビジョンという現象に対応する必要性などが，掲げられています．

本書全体を通して流れているのは，"People Propose, Science Studies, Technology Conforms" という標語に乗っ取ったかれの信念，すなわち科学技術の使い方も，システムの設計の仕方も (これはコンピュータに限らないのですが)，すべて人間が中心であるべきだというものです．それは，裏を返せば，現在幅を聞かせている科学技術優先主義に対する 21 世紀に向けての反省，非難，そして警鐘にほかなりません．

1933 年のシカゴの万博で用いられた標語 "Science Finds, Industry Applies, Man Conforms" が表わすように，産業革命以降のわたしたちはいつも，科学技術に「合わせる (conform)」ことを強いられてきました．しかし，本来，技術とは，人間の能力を高めるためのもの，すなわち本書のタイトルにいう "Things That Makes Us Smart" であるべきだというのがかれの主張です．この主張が，現在使われているいろいろなシステムの例などを用いて，とても読みやすく記されています．

"Psychology of Everyday Things" と同様に，なるほどと思わされる逸話が多く，一気に読めてしまいます．時に，独断的な解釈に頭をかしげる部分がなきにしもあらずですが，そこは著名人，うまく頑固者を通しているなという，暖かみのある本です．ハードなサイエンスを用いて，ハードなプログラミング言語を使うことを強いられ，ハードウェアの気ままに振り回されながら，ハードワークに励んでいる方々を勇気づけるお薦めの一冊です．

"Things That Make Us Smart: Defending Human Attributes in the Age of the Machine", Donald A. Norman, Addison-Wesley Publishing Company, USA, 1993. ISBN 0-201-58129-9.

中小路久美代 (University of Colorado & SRA Boulder)

ソフトウェア・シンポジウム '94

1994 年 6 月 15 日 (水) ～ 17 日 (金)

於： 金森ホール (北海道 函館市)

開催案内および参加者募集

主 催

ソフトウェア技術者協会

協 賛

日本ソフトウェア科学会 (JSSST)　情報処理学会 (IPSJ)
情報サービス産業協会 (JISA)　北海道ソフトウェア協会 (HSA)

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，さまざまな場で活躍している技術者，管理者および研究者が一堂に会し，ソフトウェア技術に関する多面的な経験や知識を交流するユニークで貴重な場として開催されてきたこのシンポジウムも，今回で 14 回目を迎え，初めて本州を離れ，津軽海峡を越えて，函館で開催します．

ここ数回は，ダブルトラックでの会議がつづきましたが，今回は，会場の関係もあって，単一トラックで，招待講演 x 1，論文発表 x 5 (17 篇)，パネル討論 x 2 というセッション構成で，プログラムを編成しました．ツール展示はなくし，その代わりに夜のパネル (IEEE Software 誌で話題を呼んだエッセイ "Software Lemmingineering" に関するディベイト) を入れて，より充実した意見の交流が図れるように，と意図しました．

招待講演は，情報処理振興事業協会 (IPA) の棟上昭男さんに，「日本のソフトウェア産業の将来を考える」というタイトルで，鋭い問題提起をお願いしてあります．また，クロージングパネルでは，「ソフトウェアプロセスの国際標準化動向」と題して，最近とみに注目を集めつつある ISO-9000 や SPICE といった話題を中心に，最新状況の報告も含めホットな討論をたたかわせたいと考えています．

多数の方々の積極的な参加をお待ちします．

シンポジウム・スタッフ

実行委員長 : 杉田義明 (日本 NCD)
プログラム委員長 : 玉井哲雄 (東京大学)　渡邉雄一 (アスキー)

プログラム委員 (50 音順) :

青山幹雄 (富士通)　荒木啓二郎 (NAIST)　伊藤昌夫 (MASC)　大塚理恵 (RSK)　大場充 (広島市大)
兼子毅 (東京大)　久野靖 (筑波大)　岸田孝一 (SRA)　熊谷章 (PFU)　元治景朝 (さくら KCS)　佐伯元司 (東京工大)
坂本啓司 (オムロン)　佐藤千明 (長野県協同電算)　塩谷和範 (SRA)　武田淳男 (安川電機)　高橋光裕 (電力中研)
中野秀男 (大阪大)　布川博士 (宮城教育大)　野口正浩 (新日本製鐵)　野呂昌満 (南山大学)　藤野晃延 (FXIS)
二木厚吉 (JAIST)　細谷僚一 (NTT)　堀江進 (NES)　松本健一 (NAIST)　山崎利治 (フリー)

ソフトウェア・シンポジウム'94 プログラム (予定)

***** 6月15日(水) *****

13:00 − 14:00 **受付**

14:00 − 14:30 **オープニング** 司会:杉田 義明 (日本 NCD)

14:30 − 16:00 **招待講演** 司会:玉井 哲雄 (東京大学)

日本のソフトウェア産業の将来を考える
棟上 昭男 (情報処理振興事業協会)

16:00 − 16:30 ＜休憩＞

16:30 − 18:30 **Session 1: CASE** 司会:野口 正浩 (新日本製鐵)

クラスライブラリにおける設計ノウハウの共有と再利用
佐藤 啓太, 藤野 晃延 (富士ゼロックス情報システム)

Tcl/Tk を用いた PCTE アプリケーションの構築
星 孝哲 (SRA)

スキーマ定義を利用した PCTE 上のオブジェクト指向環境
小室 睦 (日立ソフトウェアエンジニアリング)

統合環境に於けるツールの在り方と統合のためのツールの分析及び設計法に関する考察
伊藤 昌夫 (MHI エアロスペースシステムズ)

18:30 - 21:00 ＜情報交換パーティ＞

***** 6月16日(木) *****

09:00 − 10:30 **Session 2: プロセス** 司会:大塚 理恵 (RSK)

People Meet Process: 開発プロセスとの出会いと対峙
青山 幹雄 (富士通)

現状プロセスの徹底分析によるソフトウェア開発のプロセス改善
○田中 敏文, 千頭 良造, 加地 覚 (オムロン), 北田 泰久 (オムロンソフトウェア)

ソフトウェア開発プロセス計画策定時の会話モデルとその実装方式
石若 通利, 元治 景朝 (さくらケーシーエス), 荻原 剛志 (奈良先端科学技術大学院大学), 井上 克郎 (大阪大学)

10:30 − 11:00 ＜休憩＞

11:00 − 12:30 **Session 3 : 品質・見積り** 司会:武田 淳男 (安川電機)

ソフトウェア品質の内部特性の階層化モデルの提案
○三宅 武司, 西山 茂, 古山 恒実 (日本電信電話)

機能量にもとづく情報システムの見積りと評価
高橋 光裕, 菱谷 淳 (電力中央研究所)

T 型マトリックスを活用したソフトウェア障害の分析
岡田 俊明 (富士ゼロックス)

12:30 − 14:00 ＜昼食休憩＞

14:00 − 16:00 **Session 4 : 仕様** 司会:佐伯 元司 (東京工業大学)

仕様の分類によるマイコン組込型システムの要求仕様の構造化について
○小松 幸広, 佐藤 正道, 丹羽 徹, 片山 泰司, 林 誠 (オムロン), 阿部 恵三, 大家 雅宏 (オムロンソフトウェア)

Human-Oriented な形式的仕様記述法
古木 良子, 荒木 啓二郎 (奈良先端科学技術大学院大学)

OBJ による Z 仕様の検証の試み
○谷津 弘一, 本間 毅寛, 中川 中 (情報処理振興事業協会)

マイコンソフト開発における設計手法の提案
○小幡 健司, 相阪 泰之, 飯田 薫, 山田 豊, 尾本 林貞 (ダイキン工業)

ソフトウェア・シンポジウム'94　プログラム (予定)

16:00 － 16:30　＜休憩＞

16:30 － 18:00　**Session 5 : 人間的要因**　司会 : 布川 博士 (宮城教育大学)

ACM カリキュラムに従ったプログラミング入門講座の事例
君島 浩 (富士通)

要求変更および保守要求要因の分析による利用者の開発参画に関する考察
中谷 多哉子 (富士ゼロックス情報システム)

CACE : A New Paradigm for CASE that Supports Collaborative Evolution of Software Artifact
Brent Reeves(SRA), ○ Kumiyo Nakakoji(Univ. Colorado at Boulder)

18:00 － 18:30　＜休憩＞

18:30 － 20:30　イブニング・パネル　司会 : 塩谷 和範 (SRA)

Software Lemmingineering !?
パネリスト: 伊藤 昌夫 (MASC), 熊谷 章 (PFU) ほか (交渉中)

*******　6 月 17 日 (金)　*******

09:30 － 12:00　クロージング・パネル　司会 : 大場 充 (広島市立大学)

これからのソフトウェア・プロセス・マネジメント
－ ISO-9000, CMM, SPICE 等の標準化動向とその影響　－
パネリスト: 坂本 啓司 (オムロン), 久保 宏志 (富士通),
兼子 毅 (東京大学), 桑名 栄二 (NTT)[交渉中]

12:00 － 12:30　クロージング　司会 : 渡邉 雄一 (アスキー)

最優秀論文賞発表/表彰
次回開催予告

参加費

会員: 38,000 円 (SEA および協賛団体会員)　一般: 48,000 円　学校関係者: 25,000 円　学生: 5,000 円

参加申込み要領

以下の申込書に必要事項をご記入の上，郵便または Fax または E-Mail で

〒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 事務局
TEL: 03-3356-1077,　FAX: 03-3356-1072,　E-Mail: sea@sea.or.jp

までお送り下さい．折り返し受付票および請求書をお送りいたします．

****

ソフトウェア・シンポジウム'94　参加申込み書　(申込み日：＿＿＿月＿＿＿日)

氏名：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿(ふりがな：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿)
種別：□SEA　□JSSST　□IPSJ　□JISA　□HSA　□一般　□学校関係者　□学生
参加費：＿＿＿＿＿＿＿＿円
会社 (学校) 名：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿
部門：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿役職：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿
住所：(〒　　)＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿
Tel：(　　)－(　　)－(　　)　内線 (　　)
Fax：(　　)－(　　)－(　　)
E-mail：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

****

航空券およびホテルの案内

ソフトウェア･シンポジウム'94 参加者の便宜のために，近畿日本ツーリスト(東京･錦糸町支店) の御協力により，往復の航空と宿泊をセットした格安出張プランを用意しました (通常料金と比較して，5 ～ 9,000 円ほど安くなります).

ホテルのご案内 (Check In： 6/15，Check Out: 6/17 または 6/18)

☆ 函館国際ホテル (港のそば，シンポジウム会場に最も近い)
☆ フィットネスホテル 330 函館 (JR 函館駅前，プールやジムの設備あり)
☆ ホテル法華クラブ函館 (史跡五稜郭の近く，会場までは市電利用)

フライトのご案内 (予定時刻)

往路： 6 月 15 日 ANA851(羽田 7:30 → 函館 8:45) または ANA853(羽田 9:00 → 函館 10:15)
復路： 6 月 17 日 ANA862(函館 16:00 → 羽田 17:15) または ANA864(函館 18:00 → 羽田 19:15)
復路： 6 月 18 日 ANA858(函館 13:30 → 羽田 14:45)

料金表	6/15 発 Air & Hotel 2 泊 3 日コース		6/15 発 Air & Hotel 3 泊 4 日コース		6/15 発 Hotel のみ 2 泊 3 日コース	
部屋の種別	シングル	ツイン	シングル	ツイン	シングル	ツイン
函館国際ホテル	63,000	60,000	77,000	72,500	28,000	25,000
フィットネスホテル 330 函館	56,000	53,000	66,000	62,500	20,500	19,000
法華クラブ函館	54,000	53,000	64,000	62,000	20,600	18,000

上記費用は往復航空運賃 (Air & Hotel コースの場合)，朝食付宿泊費 (税･サービス) を含みます．ただし，航空運賃は，申込人員が 20 名様に満たない場合は，片道あたり 2,000 円のアップになりますのでご了承下さい．

ご希望の方は下の申込所の該当欄に○印を入れ，必要事項を記入して，**FAX にて下記までお送りください**．

申込み締切は 4 月 28 日 (木) です．なお，これらのコース以外のご相談にものりますのでお問い合わせください．

近畿日本ツーリスト 錦糸町支店 (担当：清水，斉藤，松浦)
TEL: 03-3632-5321 FAX: 03-3632-4364

取消料: 5/24 まで無料，6/7 まで旅行代金の 10%，6/13 まで 20%，6/14 まで 30%，6/15 まで 50%.

****

ソフトウェア･シンポシウム'94 旅行手配申込書

申込日： 月 日

申込者氏名 (ふりがな)：_____( )

性別： □男 □女 年齢：_____才

会社名：_____

所 属：_____

住 所：_____

Tel:_____ Fax:_____

ツインの場合同室希望者名：_____

申込みコース:

□ Air & Hotel 2 泊 3 日 □ Air & Hotel 3 泊 4 日 □ Hotel のみ 2 泊 3 日

希望のホテルとタイプ

□ 函館国際ホテル □ フィットネスホテル 330 函館 □ ホテル法華クラブ函館
□ シングル □ ツイン (同室者指定) □ ツイン (同室者はだれでもよい)

希望フライト

往路： 6 月 15 日 □ ANA851(羽田発 7:30) □ ANA853(羽田発 9:00)
復路： 6 月 17 日 □ ANA862(函館発 16:00) □ ANA864(函館発 18:00)
復路： 6 月 18 日 □ ANA858(函館発 13:30)

****

SEA 1993 年～ 1994 年のイベント（実績と予定）

1993	5/28	Seminar (技術の国際化·標準化)	東京
	5/28	Forum (業界リストラクチャリング)	東京
	6/9 ～ 11	Software Symposium '93	仙台
	7/16	Seminar & Forum (ダウンサイジング時代の技術者教育)	東京
	9/1 ～ 4	第 11 回 夏のプログラミング·ワークショップ	金沢
	9/14	Seminar & Forum (インターネット)	東京
	10/13 ～ 16	泰安国際 CASE シンポジウム (TICS'93)	泰安
	10/26	Seminar & Forum (ISO 9000)	東京
	10/28 ～ 30	第 7 回 教育ワークショップ	福山
	11/12	Seminar & Forum (CASE, OO, Formal Approach)	東京
	11/25 ～ 26	第 14 回ソフトウェア信頼性シンポジウム	奈良
1994	1/27	Seminar & Forum (技術者教育)	東京
	1/27 ～ 29	SIGENV ワークショップ	長野
	3/24	Seminar & Forum (SPICE)	東京
	4/21	Seminar & Forum(最新インターネット事情)	東京
		**********	
	5/26	Seminar & Forum(ビジュアル開発環境)	東京
	6/2 ～ 4	第 6 回 テクニカル·マネジメント·ワークショップ	秋田
	6/15-17	Software Symposium'94	函館
	9/21-24	第 12 回 夏のプログラミング·ワークショップ	盛岡
	10/27-29	第 8 回 教育ワークショップ	四国
	11/16-19	Kunming International CASE Symposium	中国·昆明

入会申し込み先
〒 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）

SEA 入会申込書（正会員：入会金 3,000 円，年会費 7,000 円）　94-03

氏　名：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿　(ふりがな：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿)

生年月日：19＿＿＿年＿＿＿月＿＿＿日　　性別（男　女）　血液型（A　O　B　AB）

勤務先名：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

所属·役職：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

勤務先住所（〒　　）＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

勤務先 TEL：＿＿＿＿＿-＿＿＿＿＿-＿＿＿＿＿　（内線＿＿＿＿＿）

勤務先 FAX：＿＿＿＿＿-＿＿＿＿＿-＿＿＿＿＿

自宅住所：（〒　　）＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

自宅 TEL：＿＿＿＿＿-＿＿＿＿＿-＿＿＿＿＿

資料送付先 & 連絡先（どちらかにチェック）　□勤務先　□自宅

.....

SEA 入会申込書（賛助会員：年会費 1 口 100,000 円，何口でも可）　94-03

会社·団体名：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

代表者氏名：＿＿＿＿＿＿＿＿＿＿＿＿　(ふりがな：＿＿＿＿＿＿＿＿＿＿＿＿)

連絡担当者：＿＿＿＿＿＿＿＿＿＿＿＿　(ふりがな：＿＿＿＿＿＿＿＿＿＿＿＿)

所属·役職：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

住　所：（〒　　）＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

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

申込口数：＿＿＿＿＿口

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