

解説



4. ソフトウェア工学における標準化動向

4.1 今後の課題†

東 基 衛** 菅 忠 義††
松 原 友 夫†††

1. はじめに

今日、コンピュータ利用分野の拡大は目覚ましく、われわれは社会生活と日常生活の両面で大きくコンピュータの影響を受けている。たとえばパーソナルコンピュータやワードプロセッサは、すでに企業内で広く利用され、銀行のオンラインシステムや航空あるいは鉄道の座席予約も馴染み深い。また、自動車や家庭電化製品のように、マイクロコンピュータを組み込んだ製品もわれわれの身近に見られるようになってきている。

コンピュータを使うには、ソフトウェアが必要不可欠である。ソフトウェアは、これまでコンピュータ部門の専門家が作るものとされてきた。このような場合には、作ったソフトウェアに不具合があっても後で直すことができる。しかし、大量に販売されるパソコン用ソフトウェアや、自動車のように大量生産される製品に組み込まれるマイコンのソフトウェア、あるいは、社会的に大きな影響をもつ大規模システムのソフトウェアなどいわゆるソフトウェア製品は、特に最初から品質の良いものを低コストで生産しなければならない。つまり、ソフトウェア製品は、近代的な工業製品として考えられ、その生産性、品質向上は、限られたコンピュータ部門の問題から、広く製品の開発、生産技術、生産管理、品質管理部門などの問題へと拡大してきているといえよう。

近代的工業製品の生産に不可欠なのが標準化である。製品の規格の標準化は利用者に多くの便益をもたらし、また、部品や作業方法、工具の標準化は生産者

に品質向上とコストダウンによる利益をもたらす。ソフトウェア製品生産の場合も例外ではない。

ISO/TC 97/SC 7 に寄せられる新しい活動項目の提案が年を追って活発になり、1984年の第12回 ISO/TC 97/SC 7 トロント国際会議で、SC 7 がそのスコープ (Scope) を「Software Development and System Documentation」と改訂して、ソフトウェア工学を扱うことを鮮明にしてからは一層活発になってきているのも偶然でない。ソフトウェアの品質保証とか、消費者保護といった視点から標準化の問題が取り上げられるようになってきたのは、ソフトウェアの工業製品としての重要性が認められてきたなよりの証拠であると考えられる。本稿では、SC 7 の最近の活動を中心に、ソフトウェア工学における標準化動向を紹介する。

2. 規格化領域の体系化 (Reference Model)

前述のように SC 7 には、年を追うごとにソフトウェア工学、特に製品としてのソフトウェアに関連した新規活動項目の提案が、活発に寄せられるようになってきた。

これらの提案の中には、フランスが提案した「ソフトウェア開発技法」のように国際標準としての必要性からみて疑問のあるものや、ほかの規格との関連が十分に検討されていないものも見受けられた。

一方、標準化のための活動は、専任の人がいるわけではなく、また、特にソフトウェア工学の標準化は、受益者がメーカーよりはむしろ利用者、消費者であるということから、ボランティア的色彩も強いので参加する人の工数にも限りがある。

このような状況から、計画的に標準化活動を進めていく必要性が強く感じられるようになり、日本は、1985年ロンドンで行われた第13回の SC 7 国際会議において、運営委員会 (Steering Committee) の設置と活

† Fupule Work by Motoei AZUMA (Waseda University), Tadayoshi KAN (Department of Physics Faculty of Science, Gakushuin University) and Tomoo MATSUBARA (Japan Information Service Industry Association).

** 早稲田大学

†† 学習院大学

††† 情産協

動領域を示すレファレンスモデル (Reference Model) の作成、及びその領域からテーマを取り上げていく際の優先順位を決めるためのガイドラインの作成、ならびに長期計画の立案を提案した。この提案は、ただちに臨時の検討グループ (Ad Hoc Group) を構成して検討され、その結果が本会議に提出された。

本会議では、フランスが、日本が運営委員会を設置して SC7 を支配しようとしている、という理由で強く反対をしたため、運営委員会は設置せず必要に応じて臨時の検討グループを設けることにするとともにレファレンスモデルなどの検討を行う作業グループを設置する決議が採択された。しかし、後に ISO の規約に基づく技術上の理由から、レファレンスモデルを新しい活動項目とすることで投票を行い 1986 年 7 月に承認された。その結果、カナダを主査として正式に作業グループ 5 (WG5) が設置された。

レファレンスモデルは、いまだその原案も十分に固まっていない段階なので、今後どのような方向に進んでゆくかは明らかでない。

3. 優先順位について

前述のように、非常に多くの標準化要求項目がある現在、そのどれから手懸けていくべきかは大変に重要な問題である。日本は、そのためのガイドラインの設定に向けて努力を行っているが、その基本的な考え方は次のようなものである。

1) 受益者の人数

規格の標準化によって利益を受ける人の人数、あるいは、標準化が行われないことによって不利益を受ける人の人数は、優先順位決定の際にまず考慮すべきことである。このことは、広く一般大衆が利用する商品に関する規格を第一に取り上げるべきことを意味する。たとえば、家庭電気製品などのように大衆向けの製品に組み込まれたソフトウェアや、職場や家庭内で広く使用されるパーソナルコンピュータのために開発され、大量に販売される、いわゆる流通ソフトウェアの規格は第一に優先されるべきであろう。

2) 教 育

規格の教育目的による利用も軽視することはできない。社会のニーズに応じて、高等学校や大学あるいは専門学校などで、コンピュータ教育が盛んに行われるようになってきた。しかし、教える内容が学校ごとに異なるのでは、卒業生を受け入れる側の企業などからみると具合が悪い。プログラムの流れ図の JIS など

は、その意味でこれまでそれなりの役割を果たしてきたと評価できる。しかし、ソフトウェア工学が急速に進歩していろいろな技術が使われている現在、教育目的からも基本となる技術の標準化が必要な時期にきているといえよう。

3) 消費者保護

消費者保護の立場に立つ標準化は、上記の受益者の人数という視点と一致する面が多いが、すでによく知られているように、消費者は十分に知らされていないこと、及び一人一人の立場は弱いということからも考えられるべきである。

4) 社会的影響度

実際に稼動するシステムの数はそれほど多くはないが、そのシステムが稼動することにより社会に与える影響が非常に大きいシステムがある。また、国民の税金を用いて開発されるシステムのように、失敗や無駄使いが許されないシステムもある。アメリカ政府が、特に国防省関係や NASA (航空宇宙局) のシステムのためのソフトウェアの信頼性や品質に心を配って、その標準化の推進に熱心なのはその一例である。

5) 技術的成熟度

ソフトウェア工学は、急速に発展中の技術分野である。あまりに未成熟な技術を標準化し、すぐに改訂を余儀なくされるようでも具合が悪いし、さらに勝れた技術が開発されないと限らない。また、新しい技術の発展を阻害するようなことがあってはならない。しかし、一方でそのタイミングが遅れると似たような技術がたくさんでまわって、標準化が困難になったり、機会を失うようになり、利用者に多大の迷惑をかけたり、大きな社会的損失をまねくことになる。

以上のような各種の視点からの基本的な枠組みを基に、適当に分類し、チェックリストのようなものを作り、ガイドラインとすることが必要視されている。

4. ソフトウェアの評価基準

1980 年にオランダのハーグで行われた第 8 回の ISO/TC 97/SC7 国際会議において、フランスは、ワーニエ法に基づくソフトウェアの設計技法の標準化を提案した。これは、ワーニエ法の標準化に執念を燃やしているフランスが、すでに国際標準となった ISO 6593 「Program Flow for Processing Sequential File in Terms of Record Groups」の検討に際して、ワーニエ法に基づくプログラムを標準にしようとして果たせなかったため、形を変えて提案してきたものである。

Boehm et al. MODEL

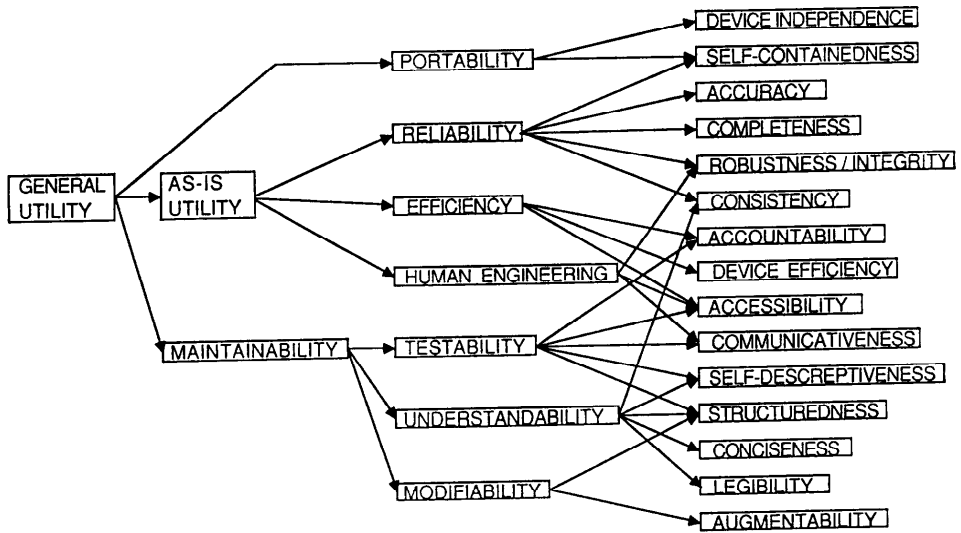


図-1 ベームの品質モデル

しかし、特定の手法の標準化を嫌うイギリスと日本は、これに反対し、イギリスは、技法を標準化するためにはその技法の優劣を判断するための評価基準が必要不可欠であることを主張し、ソフトウェア開発技法評価基準 (Criteria for Evaluation of Software Development Methods) を提案した。

さらに、日本は、技法の優劣を評価するにはその出力としてのソフトウェアの優劣を評価することが先決問題であることを主張して、ソフトウェア評価基準 (Criteria for Evaluation of Software) を提案した。

これらの提案は、参加各国の支持を得て採用されることになったが、その後具体的な提案がないまま推移した。この間、日本は、ソフトウェアの品質に関する参考文献を提出したが、ほかの標準原案の審議で時間がなかった。

ソフトウェアに品質という概念がでてきたことは、比較的新しい。ソフトウェアの歴史の初期においては、プログラムにはバグがなければいい、ということから、プログラミングとデバグの2工

McCall et al. MODEL

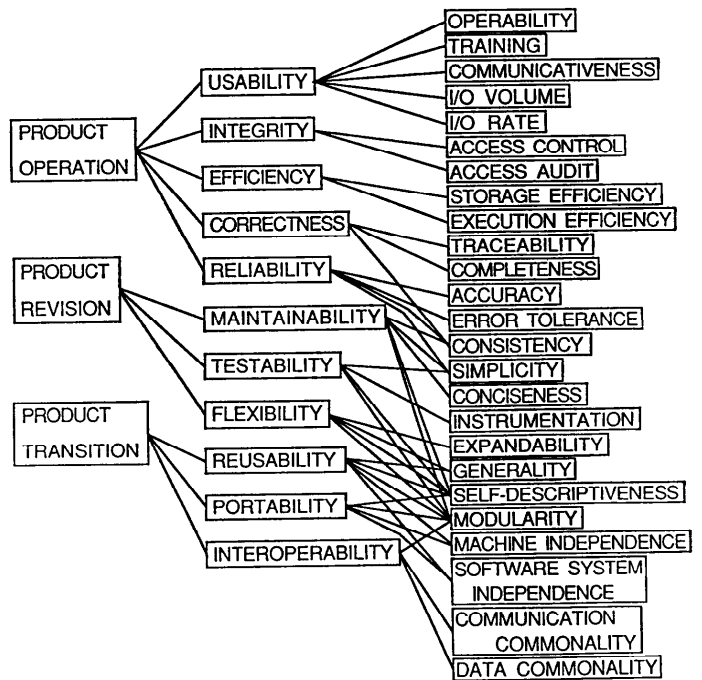


図-2 マコールの品質モデル

程のみが存在した。

しかし、次第にソフトウェアの規模が大きくなるにつれて、ソフトウェアの信頼性が重視されるようにな

ってきた。1968年に、NATO(北大西洋条約機構)の主催で行われた高信頼性ソフトウェアの国際会議は、ソフトウェア工学の発端となったものとして名高い。その後、ソフトウェア工学の進展によって、良いソフトウェアの条件は、信頼性が高いことだけではないということが広く知られるようになってきた。

このような背景から、アメリカのTRW社のベーム(B. W. Boehm)は、アメリカ国防省の要請に応じて研究を行い、1976年にサンフランシスコで行われた第2回のソフトウェア工学国際会議でその成果を発表した¹⁾。ベームのモデルは、図-1に示すようなものである。

さらに、翌年の1977年にマコール(Ma-Call)らは、図-2のような品質モデルを含む、RADC(Rome Air Development Center)レポートを発行した²⁾。このマコールらのモデルに基づいて、ミュリーン(G. Murine)は、品質計測の方法(Software Quality Metrics)を開発した³⁾。

これらの成果をふまえ、日本でもSQMAT⁴⁾、SQUALAS⁵⁾などいろいろな技術が発表され、実用化されるようになり、機が熟してきたことから、1985年2月西ドイツのミュンヘンで行われたSC7/WG3の国際会議に先立って、日本は、図-3のようなSQMATの7種類の品質要求尺度と、マコール及びミュリーンのモデルのクライテリアに準拠した23種の品質設計尺度を叩き台として提出した。

ミュンヘンの国際会議では、同時にアメリカから提出された資料も含めて討議を行った結果、まず、品質を評価するための視点としての品質特性を標準化することになり、KJ法のように品質特性を表すと思われる単語をすべて紙に書いて壁に貼って検討を行った。この結果、次の7つの品質特性が選定された。

- 機能性 (Functionality)
- 信頼性 (Reliability)
- 使いやすさ (Ease of Use)
- 時間経済性 (Time Economy)
- 資源経済性 (Resource Economy)

品質要求尺度と品質設計尺度との関係

| 品質設計尺度 | 品質要求尺度 | | | | | | | |
|-----------------|-------------|-------------|-------------|-------------|-----------------------|--------|----------------------------|-------------|
| | 正 確 性 | 信 頼 性 | 保 守 性 | 柔 軟 性 | 使 い や す さ | 効 率 | セ キ ュ リ テ ィ | 接 続 性 |
| 追跡可能性 | ○ | | | | | | | |
| 完全性 | ○ | | | | | | | |
| 一貫性 | ○ | ○ | ○ | | | | | |
| 単純性 | | ○ | ○ | | | | | |
| 誤り許容性 | | ○ | | | | | | |
| 精 度 | ○ | | | | | | | |
| モジュール性 | | | ○ | ○ | | | | ○ |
| 自己記述性 | | | ○ | ○ | | | | |
| 簡 潔 性 | | | ○ | | | | | |
| 計 測 性 | | | ○ | | | | | |
| 汎 用 性 | | | | ○ | | | | |
| 拡 張 性 | | | | ○ | | | | |
| マシン独立性 | | | | ○ | | | | |
| ソフトウェアシステム独立性 | | | | ○ | | | | |
| 学 習 性 | | | | | ○ | | | |
| 対 話 性 | | | | | ○ | | | |
| 操 作 性 | | | | | ○ | | | |
| 実 行 効 率 | | | | | | ○ | | |
| 記 憶 効 率 | | | | | | ○ | | |
| アクセス管理 | | | | | | | ○ | |
| アクセス監査 | | | | | | | ○ | |
| データ共通性 | | | | | | | | ○ |
| 通信手順・インタフェース共通性 | | | | | | | | ○ |

図-3 SQMATの品質尺度

- 保守性 (Maintainability)
- 移植性 (Portability)

この結果、規格の名称も“Software Quality Evaluation Criteria”から、“Software Quality Characteristics”に変更された。続く1985年8月のロンドン及び1986年5月の東京国際会議では、規格文章の検討が行われた。また、東京国際会議では、日本の提案により品質特性のほか、レーティングレベル、メトリクスなどの用語の定義も組み込まれることになった。これらの結果を組み入れた規格原案の編集作業は日本が担当することになった。

このような状況に対応して、日本国内では、1986年度より新たに INSTAC の ISO/TC 97/SC 7 に対応したソフトウェア開発とシステムの文書化委員会の中に、WG 4「評価」作業グループ(主査 東)を設置、品質評価をはじめとする、ソフトウェアに関する各種の評価の標準化に関する調査研究を開始した。WG 4では、上記の7つの品質特性を詳細化した第2レベルの品質特性を定義する作業を行った。また、この検討の過程で上記の7つの品質特性のうち、

時間経済性 (Time Economy)

資源経済性 (Resource Economy)

を統合して、

効 率 性 (Efficiency)

とするほうが良いことが明らかになり、1987年3月シナトルで行われた SC 7/WG 3 の国際会議で提案し、採用された。

現在検討中の品質特性及び品質評価プロセスは、**図-4** に示すようなものである。

たとえば、効率性という品質特性の下位に時間経済性、及び資源経済性という第2レベルの副品質特性 (Sub Quality Characteristics) があり、時間経済性はさらに応答時間 (Rosponce Time) などいくつかの第3レベルの副品質特性から成り立つ。応答時間をどう測るか、たとえば“改行”キーを押してから最初の文字を表示するまでを秒単位で、がメトリクスである。その結果 0.5 秒とすると、0.5 秒が大変良いか、良いか、駄目かなどはシステムの目的によって異なる。これをランク付けするのがレーティングレベルである。その結果を集計して総合的に合否判定を行うための基準が評価基準である。

現在は、副品質特性を検討中であるが、第一段階としてはここまでを標準とする予定である。しかしこれらは、今後 DIS, ISO と段階を踏んでゆかなければならないので、紆余曲折が予想される。

さらに、将来はメトリクスを順次整備してゆく予定である。国際標準としては、メトリクスまでで、レーティングレベルや、評価基準は、個々のプロジェクトごとに設定すべきものと考えられている。

5. その他の検討を開始した事項の紹介

先に述べたように、レファレンスモデルの作成作業は、現在 SC 7/WG 5 で進められている最中であるが、ソフトウェアの評価基準以外にも、レファレンスモデルの完成を待たずに、いくつかの新しい作業項目が並行して取り上げられ、標準原案作成の作業が進行中である。以下にその主なものを、紹介する。

5.1 ソフトウェア開発方法の評価基準

1985年2月に西ドイツのミュンヘンで行われた SC 7/WG 3 の国際会議では、ソフトウェア開発方法の評価基準についても検討が行われた。現在までにワーニエ法、ジャクソン法などをはじめとして、数多くのソフトウェア設計技法が紹介されている。それらの中には、コンサルティングの商品として販売されているものも少なくない。

このような状況において、その顧客、すなわち、その技法を導入しようとしている人々にとって、適切な評価基準を提供することは意義が大きいというわけである。

ミュンヘン会議では、ソフトウェア開発手法もまたある種のソフトウェアであるということから、その評

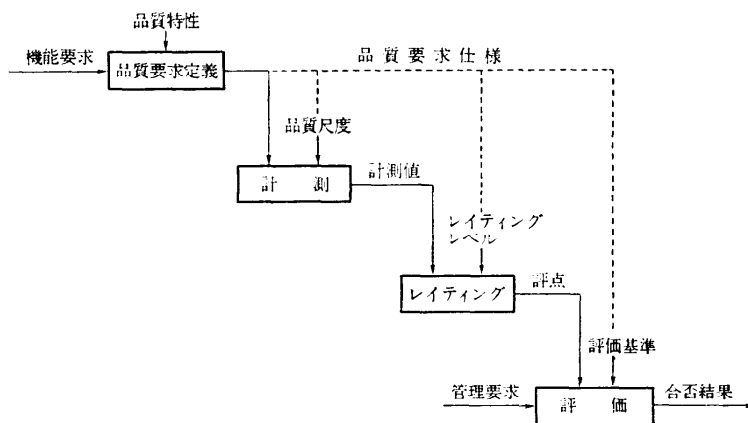


図-4 品質評価プロセス

価基準はソフトウェア品質評価基準にできるだけ合わせてゆこうということになった。規格原案の作成作業は、イギリスが担当しているが、現在のところソフトウェア品質評価基準の標準化作業の進展をみている状態である。

5.2 ソフトウェア開発方法

前述のようにフランスは、ソフトウェア開発技法の標準化を提案し、ワーニエ法を標準原案として提出した。しかし、特定の技法を標準とすることには、他の参加国が強く反対し、ソフトウェア開発技法の備えるべき条件、あるいは、データ構造指向、オブジェクト指向などの種類別のメタ技法のようなものならば標準化してもよいという妥協案を示した。

この妥協案を受けてフランスは、再度原案を作成して提案することとなったが、その後毎年の国際会議に提出された原案は、当初からのワーニエ法を基にしたものから一歩も出ていない。

したがって、SC7の主要な参加国である、アメリカ、イギリス、カナダ、西ドイツ及び日本は、毎年これに反対し、現在に至るも叩き台として採用された原案はない。

5.3 ソフトウェア開発用図記号

1986年5月東京で行われたISO/TC97/SC7国際会議でカナダは、新しくソフトウェア開発用図記号(Symbols for Software Development)標準化の提案の趣旨説明を行った。説明によると、データフロー指向の設計技法におけるDFD(Data Flow Diagram)などを意図しており、当面考えられるだけでも17種くらいあるという。

これに対して日本は、データフローはすでにJIS化されているISO 5807(Program Flow, Network and Resource Chart)の中に組み込まれており、DFDとは記号が異なるが、新たにDFDを標準化すると同じISOに同一名称同一目的の二種類の規格が制定されることになり混乱を招く恐れがあることを指摘し、また、17種の中には、状態せんい図のように標準化の意義の大きいものも、あまりないものもあると考えられることから、一つ一つ個別に審議すべきであると提案した。DFD及び状態せんい図の例を図-5及び図-6に示す。

この提案は、各国の支持を得て、カナダが具体的な提案を準備することとなった。

日本は、とりえず状態せんい図に目標を絞ってテクニカルワーキングドキュメントの作成作業を独自に

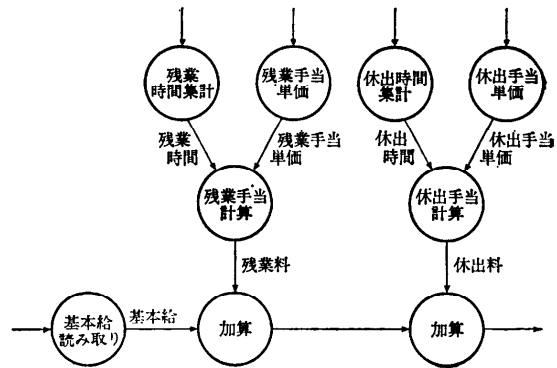


図-5 DFDの例

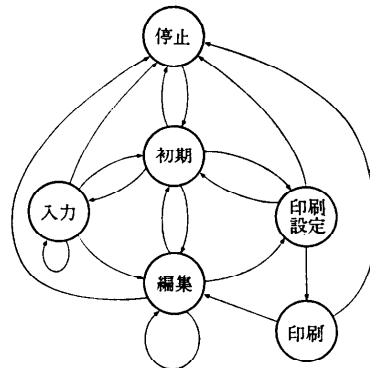


図-6 状態せんい図の例

行う方針を固め国内の委員会で作業を行っている。また、今後さらにどのような図記号を取り上げてゆくべきかについては、SC7/WG5のレファレンスモデルと関連して作業を行っている。

5.4 ソフトウェアメトリクス

ソフトウェアメトリクスも1986年5月に東京で行われたSC7の国際会議で、カナダにより提案された。まだ、投票の段階であり、正式に採択されたわけではないが、今後重要性を増してくる分野であると考えられている。

日本は、このテーマを重視して、前述のINSTAC/STDのWG4「評価」作業グループで、標準化のための調査研究作業を開始した。

メトリクスとしては、品質、規模、複雑さ、生産性などいろいろな分野があり、長期的になにを取り上げていくべきかについてはレファレンスモデルの作業グループと関連して作業を進めなければならないが、品質のメトリクスは現在すでに作業中であるため、当面規模、複雑さ及び生産性などを取り上げている。

表-1 IEEE/SESS 標準化状況

| | |
|--------|--|
| 1. 制定済 | Standard Glossary of Software Engineering Terminology (IEEE/729) |
| | Standard for Software Quality Assurance Plan (IEFE/730) |
| | Standard for Software Configuration Management Plans (IEEE/828) |
| | Standard for Software Test Documentation (IEEE/829) |
| | Guide to Software Requirement Specification (IEEE/830) |
| | Guide for Software Quality Assurance Plan (IEFE/983) |
| 2. 投票中 | Standard for Measures to Produce Reliable Software |
| | Guide to the Use of Standard for Measures to Produce Reliable Software |
| | Standard for Software Reviews and Audits |
| | Guide for Software Configuration Management |
| | Standard for Software Project Management Plan |
| | Standard for Software User Documentation |
| 3. 作業中 | Standard for Classification of Software Errors, Faults, and Failure |
| | Standard for Software Productivity Metrics |
| | Guide for Software Verification and Validation |
| | Standard for Software Software Quality Metrics |
| | Recommended Practice for Software Certification |
| | Standard for Software Life Cycle Process |

まだなにも方針が決まったわけではないが、考えられる叩き台としては、COCOMO⁹⁾のような関数型モデル、ファンクションポイントなどがあげられる。シヤトルで1987年3月に行われたSC7/WG3の国際会議では、ファンクションポイントだけに絞るよりも、なるべく広い範囲で検討していくことで合意した。

6. 他機関との関係

ISO/TC 97/SC 7 以外でも多くの機関がソフトウェア工学関連の標準化に熱心に取り組んでいる。最もアクティブに活動しているのが、IEEE 標準化委員会のソフトウェア工学分科会である。すでに標準となったもの、及び現在検討中の課題は、表-1 に示すとおりである。

アメリカでは、IEEE のほかにも NBS (国立標準化局, National Bureau of Standard) は、FIPS-PUB という一連の連邦情報処理標準を出版している。また、国防省は、独自に軍関係の標準として、MIL シリーズを発行している。これらの標準をベースにアメリカの国家標準が定められているようである。

アメリカ以外は、国内の状況はよく知られていないが、カナダ (SCC)、イギリス (BSI)、オランダ (NNM)、フランス (AFNOR) などが独自に標準化を進めているようである。特に最近カナダの意気込みが目立つ。

国際的には、IEC/TC 83 に IEEE の標準が持ち込まれるケースがでてきており、これは、投票の結果否決されたが、ISO と IEC の間の調整という新たな課

題を残した。この投票では、ISO/TC 97/SC 7 でアクティブな国がいずれもソフトウェア工学を ISO の課題として、反対投票をしたのが印象的であった。現在、このような問題を解決するために、ISO/TC 97 と IEC/TC 83 の共同委員会 (Joint Committee) を設立することが検討されている。

このほか、マンマシンインタフェースについては、ISO/TC 97/SC 18 及び ISO/TC 159 (人間工学)、品質保証については ISO/TC 176 (品質保証) が SC 7 と競合するという問題がでてきており、早期の調整が望まれている。

7. おわりに

ソフトウェアの歴史は、非常に短い。特に、産業活動としてのソフトウェア生産、及びその成果であるソフトウェア製品の歴史は浅い。

しかし、過去において標準規格の制定に失敗した製品の命運をいくつかみてきている。標準規格をもたないことによる消費者の不利益も見逃してはならない。歴史の浅いことを有利に生かして早期に標準化の推進を行うことが必要である。

一方、歴史が浅いということは、技術的に未成熟で、今後早いテンポで技術の進歩、革新が行われる可能性が高いことを意味する。標準化の推進が健全な技術の発展を阻害するようなことがあってはならない。また、反対に慎重に過ぎる余り、技術が枯れて陳腐化し、使われなくなったところ標準化されても無意味であ

る。アメリカが一時 SC7 から撤退したのも、SC7 が当時すでにあまり使われなくなった流れ図ばかり扱っていたためである。しかし、最近の SC7 の動向をみて、アメリカは再びこの分野に力をいれ始めてきている。

これまで日本は、その急激な生産性の向上により、しばしば経済活動のみに関心をもつという誤解を受けている。ソフトウェア工学の標準化活動は、教育や、消費者保護という側面ももっており、直接メーカの利益に結び付かないこともあるかもしれない。しかし、このような立場に立つ貢献が国際的な日本の正当な理解を助けるということを忘れてはならない。これまで SC7 において日本が果たしてきた役割は、国際的にも大きく評価されており、今後の日本の役割に寄せられる期待も大きい。

参 考 文 献

- 1) Boehm, B. W. et al.: Quantitative Evaluation of Software Quality, 2nd ICSE (1976).
- 2) RADC TR-77-369 (1977).
- 3) Murine, G. E. et al.; Applying Software Quality Metrics, ASQC Quality Congress Transaction (1983).
- 4) Sunazuka, T, et al.: SQMAT: Quality Assessment Technology, 8th ICSE (1985).
- 5) 日本 IBM: SQUALAS, ACCESS (1985).
- 6) 東 基衛他: ソフトウェアの品質計測/保証技術 (SQMAT), 品質, Vol. 16, No. 1.
(昭和 62 年 6 月 5 日 受付)