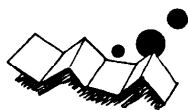


解説



ソフトウェア工学における品質管理(QC) と品質保証(QA)[†]

菅野 文友^{††}

1. ま え が き

製品の生産においては、品質保証 (Quality Assurance: QA) が重要なことは常識である。QA とは、消費者が安心して、満足して買うことができ、それを使用して安心感、満足感をもち、しかも長く使用することができるという、品質を保証することである。この QA こそ、品質管理 (Quality Control: QC) の真髄である。そして、QC の基礎は、統計的管理状態の確保にある。この統計的管理状態こそ、信頼性確保の基本的な問題である。

巷間ややもすれば、極端にソフトウェアの特殊性を云々することによって、QC や QA との絶縁を強調しようとする徒輩が存在している。そのような言動を発する面々の多くは、QC や QA の実態を知らずに、いわば喰わず嫌いの偏見を露呈しているに過ぎない。

現在の我が国における QC/QA 活動の躍進に鑑み、既往の成果を踏まえた諸種の理念や技法の適用を勇敢に意図することが、いわゆるソフトウェア生産の危機を乗り切る有力な眼目の一つであろう。虚心に同一視して推進することによって、はじめて、本来の確固とした異質の様相がクローズ・アップしてくるものであることを忘れてはならない。

ここでの所論は、固有技術としてのソフトウェアに対する共通技術としての品質管理 (品質保証および信頼性にも関連) の寸見に過ぎない。製品としてのソフトウェアの生産を、品質の面から管理してゆく手だては、ようやく始ったばかりである。今後の多角的な進展が切望されるものである。

2. 品質管理とは

品質管理とは、顧客の要求する品質——それは単な

る品質仕様だけでなく、製品の働き、寿命、使用の経済性ならびに安全性などを指す——を確保することができるよう、企業の品質目標を定め、これを合理的かつ経済的に実現するための、すべての活動の全体をいう。すなわち、品質という企業の重要目標を達成させるための経営の道具である。

この活動は、経営者のリーダーシップのもとに、調査、開発、設計、施設、購買、製造、販売、輸送、据付け、修理サービスなどの生産・販売の全活動において、遂行しなければならない品質に関係する諸業務を明確に規定し、それらを責任もって遂行する体制を作り、合理的な管理の手法を採用して進められるものである。したがって、品質管理は、特定の専門家だけによって遂行され得るものではなく、経営幹部から第一線の従業員に至るまでの全員が、それぞれ責任を分担して進められる全社的な活動を必要とするもので、今日品質管理とえば、この全社の品質管理 (Company Wide QC) あるいは総合的品質管理 (Total QC) を指しているのである。

品質管理を合理的に遂行するためには、各業務に対し、それぞれ目標を定め、実現のために、計画の策定、実施、結果の評価、処理といういわゆる Plan-Do-Check-Action の管理のサイクルを回転させつつ、管理の対象の安定化と向上を計り、かつ統計的方法を中心とする科学的方法を十分に活用することが必要である。統計的方法は多くの事象を解析し、診断し、推測するうえに必要なにして欠くことのできない方法なのである。

また、これからの QC は、製品責任 (product liability) を予防する計画を折り込んで推進する必要がある。このためには、PL 関係の法規その他の情報をよく研究し、欠陥製品の出荷予防対策を進めるとともに、ラベル、製品説明書や注意書などについても過失のないように十分な注意を払わなくてはならない。

[†] Quality Control and Quality Assurance for Software Engineering by Ayatomo KANNO (Dept. of Computer Science, Faculty of Engineering, Iwate University).

^{††} 岩手大学工学部情報工学科

表-1 QC 七つ道具

項番	区 分	説 明
1	特 性 要 因 図	原因を整理する
2	パ レ ー ト 図	重要な問題を見つける
3	チェ ック シー ト	簡単にデータをとる
4	ヒ ス ト グ ラ ム	分布の姿を目でみる
5	散 布 図	アクションと結果の対応をみる
6	管 理 図	グラフに幅をつける
7	グ ラ フ	目にものをいわせる

3. 品質管理の定石的手法

古くから、品質管理の場で日常的に使用されている手法としては、表-1に示すような品質管理七つ道具がある。また、それに加えて、最近提唱されているものとして、表-2に示す新品質管理七つ道具がある。いずれも、固有技術のそれぞれの特色的様相と密に結合して、それなりの効果を発揮しているものである。

また、ソフトウェア生産が人間の集団としての活動によって営まれる以上、生産職場におけるチームワークの成果を効果的に発揚するための職場小集団活動の重要性が認識されよう。従来の品質管理活動では、QCサークルと呼称するグループ活動が著名である。

QCサークルとは、同じ職場内で品質管理活動を自主的に行う小グループである。この小グループは、全社的品質管理活動の一環として自己啓発、相互啓発を行い、QC手法を活用して職場の管理、改善を継続的に全員参加で行う。全社的品質管理活動の一環として行うQCサークル活動の基本理念はつぎのとおりである。

- (1) 企業の体質改善・発展に寄与する。
- (2) 人間性を尊重して、生きがいのある明るい職場をつくる。
- (3) 人間の能力を発揮し、無限の可能性を引き出す。

ソフトウェア生産の場においても、このような活動の活発化が基本的課題である。ソフトウェアの場合には、開発から保全までの各段階において、ソフトウェア生産に特有な諸要素を加えて推進する際には、従来のQCサークルとは若干異質の点も考慮して、たとえば、「QCユニット」等といった活動を精力的に盛りあげることも効果的である。

4. ソフトウェアの品質管理と信頼性向上

4.1 ソフトウェアと品質管理

表-2 新 QC 七つ道具

項番	区 分	説 明
1	連 関 図 法	複雑な要因のからみ合う問題(事象)について、その因果関係を明らかにすることにより、適切な解決策を見出すのに役立つ方法。
2	K J 法	混沌とした状態の中から収集した言語データを相互の親和性によって統合し、解決すべき問題を明確にする方法。
3	系 統 図 法	目的を果たす最適手段を、系統的に追求していく方法。
4	マトリックス図法	多元的思考により、問題点を明確にしていく方法。
5	マトリックス・データ解析法	マトリックス図に配列された多くの数値データを、見通しよく整理する方法。
6	【過程決定計画図法】 PDPC (Process Decision Program Chart) 法	事態の進展とともに、いろいろな結果が想定される問題について、望ましい結果に至るプロセスを定める方法。
7	アロー・ダイアグラム法	最適の日程計画をたて、効率よく進捗を管理する方法。

ソフトウェアの汎用性に対処するための標準化のあり方については、保全性と拡張性を適切に具備したソフトウェアの生産が強く望まれている。

ソフトウェアを製品として扱う立場からすると、品質管理の活用はごくあたりまえの話である。すなわち、ソフトウェアの生産管理についても、工程管理、原価管理、品質管理の三要素から成る。もちろん、この三要素は完全に独立ではなく、微妙な絡み合いの効果を呈している。この品質管理の対象としてのソフトウェアは、コンピュータのプログラムとそれに関連するドキュメントをもって構成されると考える。ソフトウェアのQCに対して考えねばならないものは、まず、その品質の評価尺度である。どんな物差しで品質をはかるかである。

次に大事なことは、計画段階から設計、製造、検査、運用、保全を経て廃棄に至るまでのそれぞれの過程に焦点を合わせた方法論の定式化である。さらに、品質向上の常道である「源流の清め」に対応したデザイン・レビュー(設計審査)の問題を、しっかりと処理しつづけなければならないのである。

そして、こういった活動の効果は、明確かつタイムリに、稼働実績の着実な把握とその多角的な活用を図ることによって具体化される。

また、ソフトウェア生産のほとんどは人間そのものの行動と直結しているだけに、モチベーションの向上、ヒューマン・エラーのリカバリ、チームワークの発揚、ドキュメンテーションの効果発揮、その他もろもろの人間臭い問題が陸続と輩出していることにな

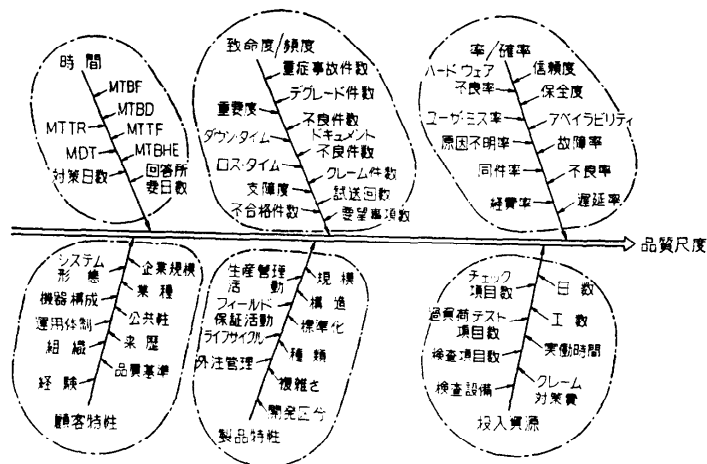


図-1 ソフトウェア品質尺度因子

4.2 品質評価測度の設定

品質管理活動の対象とするソフトウェアの品質は、一義的には、決まらない。必然的に多面的・多次元の尺度であり、また、T.P.O. によつての変移性も高い。そして基本的には、ソフトウェアを利用するユーザによつて決まる。つまり「フィジカル」なものと同時に「メンタル」な様相を考慮しなければならない。同時に、ソフトウェアを生産している工程の各ポイントに対して、中間製品としての評価測度を明確に設定しておくことが大切であり、それによつてはじめて適切なデザイン・レビューの実をあげることができる。現実には評価測度を考える際には、その製品が管理プログラム、言語処理プログラム、ユーティリティ・プログラムを含むオペレーティング・システム (OS)、すなわちコンピュータ・システムの基本ソフトウェアなのか、問題向きに設定されたプログラムとしてのアプリケーション・パッケージなのか、あるいはまた個々のユーザの業務を処理するためのユーザズ・プログラムなのかによつて、きめ細かな特有性の配慮を必要とするとともに、それらに共通した諸問題、たとえば、共通的なエラー・リカバリ (フル・ブルーフ、フェイル・セーフ、フェイル・ソフトリのすべてを含む) などについても、矛盾のない対処方を吟味しておくことが大切である。

ソフトウェアにおける品質評価尺度因子に関する具体的な事例としては、図-1 に示すものがあげられる。この図-1 の上側は、時間、致命度/頻度、率/確率、といった区分にしたがつての品質尺度のいろいろであ

る。また、下側には、このような品質尺度を設定するに際して考慮すべき諸条件を、顧客特性、製品特性、および投入資源、といった分類にしたがつて示している。

4.3 デザインレビュー(設計審査)の重要性

ソフトウェアの品質管理では、デザイン・レビューの重要性が大いに強調される。その対象は、中間製品としての各種のドキュメントである。製品の特色を明示した基本仕様書、受決仕様書と同一機能を明示するとともに、取扱説明書の原型ともな

る機能仕様書、その機能を具体化するためのしかけ—すなわち、アルゴリズムを明示した設計仕様書、さらにそのアルゴリズムを実現するに際しての手順を示した製造仕様書 (詳細な流れ図等) のそれぞれについて、入念なデザイン・レビューが大切である。

岡目八目、三人寄れば文珠の知恵、亀の甲より年の功、といった形で他人のフンドシを大いに借りまくり、縦から横から斜めから徹底した吟味を意欲的に行うと同時に、その結果の記録を着実にフォローアップすることが必要である。プロジェクト的な仕事の要領は、入念な取敢りにある。ソフトウェア・エンジニアリングの場合でも、最も大切なのは仕様を明確にする設計段階であることを忘れてはならない。

要は、デザイン・レビューによる「先憂後楽」こそ、コスト有効性の立場での信頼性向上方策の一大必須要素であると銘記すべきである。

デザイン・レビュー時の留意事項は、表-3 に示すとおりである。また、デザイン・レビューの効果を表-4 に示す。さらに、ソフトウェアのデザイン・レビューにおける表現上、および内容上、の指摘事項を例示すると、それぞれ表-5 および表-6 に示すようなものがある。

4.4 ソフトウェアの検査の問題

検査部門は、設計部門とは独立に、機能仕様書に基づいた検査仕様書を明確に設定するとともに、メーカーとしての生産側における唯一のユーザ的評価を行い得る立場を堅持しなければならない。エラー・リカバリ、過負荷処理、保安全性、拡張性などの面での確かさを確認するとともに、そのソフトウェアについての

表-3 デザイン・レビュー時の留意事項

項番	項目	理由/内容
1	動機づけ	デザイン・レビューを設計者が積極的実施するためには、デザイン・レビューの効果を設計者が直接感ずることが重要な要素となる。
2	計画的なデザイン・レビューの実施	審査員として、他システムのデザイン・レビューに参加する場合は、システム単位、プロジェクト単位の工数費与率の低下をもたらすので、事前に工数計画が必要である。
3	手法の確立	デザイン・レビューの効果を上げるためには、十分に体系づけられた手法を取り入れて、ソフトウェア生産に、より適切なデザイン・レビューを実施することが大切である。
4	規格化の推進	デザイン・レビューの早期実施を実現するための、仕組を設定することが大事である。
5	技術者の育成	デザイン・レビューの実施を通して、OJTにより、技術力およびデザイン・レビューに精通した技術者を育成するようすすめる。
6	人材の登録	専門技術者のリストアップが必要。これによって、構成メンバーの選択が容易にできるようにする。
7	データバンクの確立	デザイン・レビューの効果を上げるための重要な要素となるフィールドデータを効果的に集約し、活用できる体制が必要である。
8	評価の確立	デザイン・レビュー結果の実用的な評価法を、確立する必要がある。

表-5 表現上の指摘事項の例

項番	区分	項目
1	基本不良	表現形式の規格不遵守
2		表現の不統一
3	表現不良	文字的誤り
4		文章曖昧
5	努力不足	誤字/脱字
6		誤記入
7		記述不足
8	技術不足	記載もれ
9		内容の誤り

PL (product liability) 的方策を熟慮し、設定しておくことが大切である。特にこの際、ドキュメントの面での品質確認を十分に実施しておくことが肝要である。

また、検査部門の重要な業務の一つとして、稼働実績の把握とフィードバックがある。その際、設計不良や製造不良に対する吟味と同時に、検査不良に対しても謙虚な反省の記録を積み上げてゆかねばならない。諸般の失敗の記録をデータバンクとして活用し、他山の石として見直してゆくことは、すべての製品にとって、品質向上に際しての共通の定石として心すべき要点である。そして、ソフトウェアにとっては、特にその即効的な教訓が得られることが多い。

4.5 稼働実績の把握

表-4 デザイン・レビューの効果

項番	区分	項目
1	信頼性	設計不良の早期撲滅による品質向上
2		設計者間の情報伝達の徹底 (誤解に基づく不良の撲滅)
3		他システムからの関連技術の反映、フィールド・データの反映
4		システムの問題点の把握の徹底
5	管理の強化	中間製品の品質の把握と改善
6		品質向上のための適切な指示
7		次工程への問題点のフィード・フォワード
8	削減・低減	ドキュメント試送回数の低減
9		検査工数の削減
10		ドキュメント修正作業に必要とする設計工数の削減

表-6 内容上の検討事項の例

項番	区分	項目
1	記述	曖昧、不明確
2		不適切 (もっとよい表現がある)
3	仕様	機能欠落
4		システムと基本的に矛盾
5		仕様不適切
6		使用上の必要条件の検討不足
7	インタフェース	ハードウェアとのインタフェース不良
8		ほかのプログラムとのインタフェース不良
9		同じプログラム内でのインタフェース不良
10		ドキュメント間の矛盾
11	その他	全般

一般にコンピュータ・ユーザが、自分の使いたいソフトウェアの機能や性能を、仕様として事前に明示することは難しい。したがって、ある程度の要望は開陳し得ても、その他の特性は、メーカーまかせになることが多い。しかし、実際にそのソフトウェアを使ってみると、あらためていろいろと潜在的な要望が顕在化してくるのが普通である。したがって、ソフトウェアの出荷 (リリース) に際しては、稼働実績とそれに基づくユーザの声を適切に反映するように、バージョン・アップ (機能の追加や性能向上の実現) とリビジョン・アップ (不具合の修正) の実施時期を計画しておかねばならない。こういった稼働実績の把握に際しては、受動的な態度では駄目で、能動的なアプローチを意識しなければならない。特に、コンピュータの場合には、ハードウェアのトラブルやユーザの運用形態などとの複雑な絡み合いがあり、表面的な事実の裏に

隠された意外な真実を即座に見抜くことはなかなか難しい。

常日頃から、稼働実績把握のネットワークを陰に陽に張りめぐらしておくとともに、把握できた真実にもとづく対処とフィードバックを、タイムリに実施しつづける習練を積み上げることが必要である。そしてそのネットワーク自体についても、常に見直しを怠らず、意欲的に更新・改廃してゆく心構えがポイントである。

こういった稼働実績の把握に際しては、信頼性工学の場で広く適用されている解析技法、すなわち、FME (C) A, FTA, 分散分析などを大いに活用し、データに語らしめ、データに学ぶことが大切である。図-2には、クレームのFTAを例示する。

5. ヒューマン・ファクタの重要性

由来ソフトウェアに含まれる不良(バグ)には、いわゆる高級・高度のものはあまりない。その不良のほとんどは、ケアレスミス、ボンヤリミス、ウッカリミス、いわゆる“ポカ”の類である。ソフトウェアほど、その生産に関与した人間の特性、つまり共通の特性と個人的特性がにじみでているものは数少ない。極言すれば、ソフトウェアの「品質の管理」の多くは、その生産にたずさわる「要員の管理」でもある。人間くらい安易に誤りを犯し、しかもその誤りを安易に忘却できるものは少ない。先人の失敗を何度も繰り返し、ほかから注意されてもあまり気にせず、気がついたときはもう手遅れ、といった様相が、人間活動の実態であろう。そこを何とかしてゆくのが、ソフトウェア生産活動の重要な鍵である。

6. プロジェクト・エンジニアリングのあり方

ソフトウェアにおける生産の様相は、部分的には在来の製品を極力活用するとしても、全体としては、何らかの意味での新製品を設定している。こういった見地から、計画、設計、製造、検査、運用、保全といったそれぞれの段階に対応して、ソフトウェア品質保証のための諸方策を、プロジェクト・エンジニアリングとして、表-7に示すようにまとめることができる。

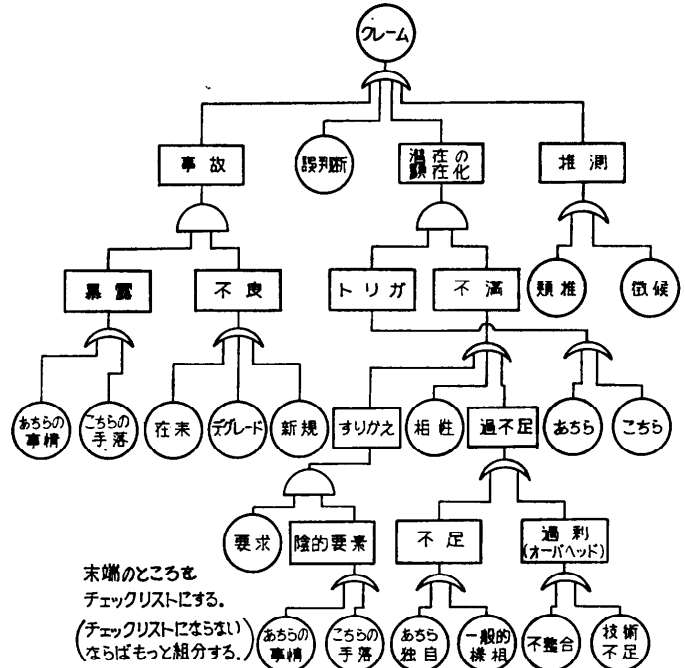


図-2 クレームの FTA

7. 具体的な推進に際しての注意事項

1) 正しい TQC を

品質管理は、部分的に、局部的に、汗水たらしても、ちっとも効果はない。直接部門と間接部門のいかんを問わず、社内と外注を問わず、技術部門と事務部門(こういったわけ方も今や時代遅れであろう)を問わず、いろいろな角度から全社の品質管理 (Total Quality Control: TQC) を推し進めてゆかねばならない。

いたずらに手を抜いたり省略したりせず、基本的な定石をしっかりとふまえ、奇道によらず、正道を着実に踏みしめてゆくことが、関係者全員の相互信頼感の発揚により、各部門での成果の相乗的な成果を具現する道である。

2) 原価管理・工程管理・品質管理の三位一体

たとえば、工程管理がうまくゆかずには、信頼性の高い製品は生まれ得ない。「一事が万事」とおり、あらゆるところがうまくゆくことが第一である。原価と工程と品質という3つのものの管理がうまくゆくことが、この全体の協調は成果としての信頼性向上につながる。要は、バランスのとれた管理が最も効果的である。

3) データ使いのデータ溺れ

表-7 プロジェクト・エンジニアリングとしてのソフトウェア・エンジニアリング

項番	段階	ポイント	備考
1	計画	1. 工程管理計画 2. 原価管理計画 3. 要員管理計画 4. 品質管理計画 5. 更新・改変・廃棄計画 6. 周辺システムとの交互作用の予測	
2	設計	1. 利用者を加えたデザイン・レビュー 2. ユーザ・クレームの吸収 3. 拡張性と保全性の感与 4. 製造段階における作業性の考慮	数回 情報ネットワークの活用 技法活用
3	製造	1. モチベーション 2. 機会損失の排除 3. 健康管理 4. 品質管理(狭義)	参画意欲と達成感 心身
4	検査	1. エラー・リカバリ確認 2. 保全性確認 3. 取扱説明書確認 4. 事故時の周辺への影響予測と措置の確認	
5	運用	1. 稼働実績データ把握 2. 拡張性のニーズ把握 3. 周辺との協調性維持 4. 改廃ニーズの把握と伝達	+, -の両方向
6	保全	1. 稼働実績のフィードバック 2. 保全要員のモチベーション 3. 改廃時の周辺への影響対策措置 4. 運用との協調と事前対策活用	
7	全般	1. あらゆる副次的効果や反作用への配慮 2. 広範な条件下でのコスト有効性の保持 3. タイミングのよい意志決定 4. グローバルな存在意義の確認 5. ドキュメント管理の徹底	他分野の知恵を活用

同じ稼働実績といっても、周辺条件には、かなりの差があり、データの数値のバラツキの度合もさまざまなものが多い。

データそのものの持つ意味、データの有効性(有効桁数も含めて)、データの有効領域、その他についていろいろと入念な吟味が大切である。そのデータの背景を確認せずに安直な解析を行った結果よりも、常識的な判断の方がたしかな場合も少なくない。

むやみやたらにデータを使いまくるのは、時には、かえってデータに溺れ、データに引きずりまわされてしまう懸念がある。

4) メモリサイズに執着すべからず

プログラムの設定に関しては、極力占有領域を減らすように心がけることは常識的なアプローチであろう。しかし、その度が過ぎると、回路モジュール設計時におけるいわゆるマジック回路の形成のようなものになり、信頼性の立場からは逆効果を招来する。

まさに、過ぎたるは及ばざるが如しである。事実、これまで、プログラムのメモリ・サイズ縮減を実現しようとする努力のために、どれだけの人力と時間と経費が失われていったかを考えると空恐しくなる。プログラムを格納するメモリの大きさを気にせず(といっても、勿論限界はある)、信頼性の高いプログラムを設定することこそ、今後の正道であろう。

要は、全体としてのコストの有効性を常に確認しつつゆくことである。

以上のような見地から、ソフトウェアの品質管理のあり方をまとめてみると、次のような15項目で表現できよう。

◎ ソフトウェア品質管理の15のポイント

1) 職場小集団活動の活発化, 2) 個人管理の徹底, 3) 参画意欲の刺激, 4) 達成感の体得, 5) ドキュメンテーションの徹底, 6) 設計不良や検査不良の記録化, 7) OJT の徹底, 8) QC 七つ道具の日常使用, 9) デザイン・レビューの推進, 10) 工数管理票の活用, 11) アロー・ダイアグラムの設定と更新, 12) 品質事前把握技術の活用, 13) フィールド・データバンクの設定・更新, 14) バグ予測管理の徹底, 15) 監理チームの活用。

8. む す び

局部的な最適化をいろいろと独立に行っても、それが大局的な最適化を保証することにならないのは、ソフトウェア工学においても同様である。今後とも、しばらくの間はさまざまな試行錯誤が試みられよう。次第にそれが収束して定式化されてゆくことが望ましいが、その実現時点についての予測は至難である。正直な話、“日暮れて道遠し”の感も強い。

しかし何とかしなければならぬ。従来、生産管理技術者の参加が少なく、それだけに品質管理的な認識がほかの工学ほど顕著ではなかった上に、もっぱら理学的な立場からのアタックがなされていたソフトウェア品質管理の問題について、今後大いに参画意欲と達成感をもった品質管理エンジニアの介入・参画が望まれるところである。このことは、メインフレーム・メーカーであっても、ソフトウェア・ハウスでも、商品としてのソフトウェアを生産する立場では、すべて共通の課題である。

要するに、ソフトウェアの QA 方策の第1は、ドキュメンテーションにある。その場合、標準化、図表化、更新改廃規則設定、に注目することが大切である。第2は、それに対するデザイン・レビューである。そのためのしかけをしっかりと設定し、しつこくきちんと遵守し、関連部署とのつき合いよろしく推進することが重要である。第3は、限界テストである。特に、高トラヒック、タイミング、エラー・リカバリ、に注意する必要がある。第4は、エージングである。その際、ランダムに実施し、記録を明確にし、信頼性工学におけるコンポーネント・アワーの理念を活用すべきである。第5は、デグレード防止対策である。そのためには、関連モジュール体系を明確に活用し、評価変移に対応し、適切な廃棄の意志決定が肝要である。

参考文献

- 1) 菅野文友：ソフトウェア・エンジニアリング，p. 453，日科技連出版社，東京（1979）。
- 2) Gunther, R. C. (水野幸男監訳)：ソフトウェア・プロダクト・エンジニアリング，p. 391，ジャテック出版，東京（1979）。
- 3) 菅野文友：情報やデータのメカニズム，p. 211，日科技連出版社，東京（1978）。
- 4) 菅野文友：ヒューマン・エラーのメカニズム，p. 232，日科技連出版社，東京（1980）。
- 5) QC手法開発部会（編）：管理者スタッフの新QC七つ道具，p. 238，日科技連出版社，東京（1979）。
- 6) 日本規格協会（編）：JIS ハンドブック品質管理1980，p. 673，日本規格協会，東京（1980）。

（昭和55年6月3日受付）