

製品品質の決定要因としての組み込みソフトウェアと組み込みソフトウェア・クライシス

西 康晴*

1. はじめに

“一寸の虫にも五分の魂”という表現がある。小さくてもおろそかにできない、という意味をわざわざ記すまでもないだろう。近年、工業製品は小型化や軽量化が顕著になる一方で、高機能化や高性能化などが進み、外見からは想像もできないほど複雑になってきている。

携帯電話を例にとってみよう。1993年には通話機能や電話帳機能しか持っていなかったが、1995年にはパケット通信によるネットワーク接続に対応し、1999年にはiモードを始めとするインターネット関連機能を備えるようになった。2000年にはカメラの搭載による画像処理機能が追加され、現在では映像の録画や再生が可能になっている。さらには電子マネー機能により金銭の授受を管理する機種もある。わずか10年で、劇的な高度化が進んだと言えよう。また携帯電話以外にも、家電製品やOA機器、車載機器など市場ニーズの変化が多様化しスピード化した製品群では、同じように高度化が進んでいる。一寸の虫が気が付くと一寸法師になっているのだ。

こうした製品の高度化を支えている“魂”が「組み込みソフトウェア (embedded software)」である。その昔ハードウェアを制御していたリレー回路が高度化して特定半導体による電子制御になり、汎用のマイクロプロセッサ (CPU や ECU) 上で動作するソフトウェア制御に進化するという歴史を辿ってきた。そのためファームウエ

アや制御ソフトウェア、埋め込みソフトウェアとも呼ばれる。近年では制御だけでなく製品間のネットワーク通信や、携帯電話における電子マネー機能といった付加機能の実装、またコストダウンの要請によりスイッチなどのユーザインタフェースの置き換えなど、製品の機能の多くをソフトウェアが実現している。経済産業省が実施した調査^[1]によると、製品開発組織の半数以上では、全体の開発費の3割以上のコストを組み込みソフトウェア開発に投下しているほどである。五分の魂どころか、中枢を担うようになったと言っても過言ではないだろう。

その結果、製品全体の品質、すなわち性能や信頼性、操作性などが組み込みソフトウェアに依存するようになってしまった。しかし残念ながら組み込みソフトウェアの品質、特に信頼性は十分確保されているとは言えない。携帯電話や家電機器、カーナビゲーションシステムなど組み込みソフトウェアの不具合が市場に流出した事例は多く報道されている^[2]。

前述の産業調査によると^[1]、出荷後の品質問題の1/3が組み込みソフトウェアに起因している。しかしより危ぶむべきは、従業員1万人以下の企業で部門内・全社を問わず組織的な品質管理活動を実施していない組織が3割を超えるという現状である。1000人以下の企業では半数を超えるほどである。その一方で、製品の高機能化・高性能化・複雑化が進み、コスト削減の波にさらされ、開発期間は短縮するばかりである。すなわち日本の製造業の品質管理の大きなテーマは、組み込みソフトウェアの品質や信頼性の向上に他ならない。言い換えると「組み込みソフトウェア・クライシス」からの脱却なのである。我々は真剣に正面から、このテーマに取り組みねばならないのだ。

*電気通信大学 電気通信学部 システム工学科
連絡先：〒182-8585 東京都調布市調布ヶ丘 1-5-1
(勤務先)

2. 組込みソフトウェア製品において 高い利益率を得る品質向上戦略

組込みソフトウェアの品質の向上は重要なテーマである。しかし市場の状況や顧客価値といった経営的側面について考察せずに取り組みと、過剰品質となって利益率が低下したり、デル・コンピュータに代表されるBTO(Built To Order)企業との価格競争に敗れることとなる。組込みソフトウェアの品質を向上して日本の製造業の産業競争力を強化するには、高い技術力を有効に活かしコスト競争に陥らないようにする戦略が重要なのだ。本章では組込みソフトウェアが大きな役割を担うような複雑な製品を対象に、製品の品質と顧客価値という観点から、すり合わせ型と単純組立型という両極の製品アーキテクチャについて、Christensen^[3]と藤本^[4]を基に論ずる。それによって、精緻な品質技術を高い利益率に結びつけるために必要な3つの開発力を提示する。

2.1 すり合わせ型製品アーキテクチャを基にした 品質向上による顧客価値の増大

新たなカテゴリの製品が生みだされると、性能や信頼性、デザインといった(広義の)品質は顧客の潜在的期待価値を当初下回っていることが多い。そこで開発組織は期待価値を満たすべく、品質の向上に熱心に取り組む。例えばデジタルカメラの開発では、画質の向上に取り組むべく当初80万画素しかなかったCCDの画素数を数百万に向上させた。こうした品質向上の取り組みは、顧客の期待価値を上回るまで、もしくは技術的限界に達するまで、企業間の競争という側面を持ちながら続けられる。

この競争から脱落しないためには、個々の部品の品質の向上もさることながら、多数の部品の品質向上を積み重ね、相互作用をまとめ上げる高い設計開発力が必要となる。こうした高度な技術力を要する製品アーキテクチャは「すり合わせ型(インテグラル型)」と呼ばれる。乗用車は、すり合わせ型製品アーキテクチャの代表例である。すり合わせ型製品アーキテクチャは日本の製造業のお家芸であり、品質を向上させるために極めて重要な技術体系であると言ってよいだろう。

しかし品質が顧客の期待価値に十分近づくと、品質向上競争に勝ち残ることが顧客価値を高めることを必ずしも意味しなくなる。魅力品質が当たり前品質に変化したと表現してもよいだろう。引き続き品質向上競争の土俵に残る開発組織は、過剰品質やオーバーシューティング

と呼ばれる現象に陥り、利益率や投資対顧客満足度の伸びの鈍化に直面する。もしくはより高い品質を顧客から期待される上級グレードの製品に注力せざるを得なくなる。

2.2 新機能追加による顧客価値の増大

一方、製品開発競争そのものは品質向上競争の様相を改め、二つに変質する。一つは新たな機能を追加することで絶対的顧客価値を高める競争であり、もう一つは価格を下げることで相対的価値を高める競争である。デスクトップ・パーソナルコンピュータ(PC)の開発を例にとると、前者はSONYに代表されるAV機能の付加に当たり、後者はデル・コンピュータに代表される低価格化に当たる。

こうした競争の変質は、製品アーキテクチャをも変化させていく。引き続きオーバーシューティング製品や上級グレード製品を開発していく場合は、単にすり合わせ型の設計開発を続けていけばよい。しかし絶対的顧客価値を高めるために新たな機能を追加していくと、徐々に製品が肥大化し複雑になり、すり合わせ作業が莫大なものになってしまう。結果として品質が低下した製品を市場に投入するか、開発期間が長期化し機会損失を被ることとなる。そこで新機能追加を容易にするため、製品アーキテクチャはすり合わせ型から、複数の機能コンポーネントの単純な組み立て作業だけで製品が完成するような「単純組み立て型(モジュラー型)」に移行していく。PCは、単純組み立て型製品アーキテクチャの代表例である。

とはいえ製品の基本的付加価値を与える機能は引き続き(顧客の期待価値を下回らない程度の)高品質が求められる。また追加された新機能についても期待価値を上回るまでは品質向上競争が発生する。携帯電話を例にとると、通話品質やアドレス帳といった基本機能には高品質を求められると同時に、顧客価値向上のための新機能である“着メロ”(着信時音楽演奏機能)にも和音数や演奏時間、音源品質などの競争が発生している。したがって全体としては単純組み立て型製品アーキテクチャに移行するが、新機能コンポーネントについては品質向上競争に勝ち抜くため、すり合わせ型製品アーキテクチャとなる。

2.3 単純組立型製品アーキテクチャを基にした 低価格化による顧客価値の増大と低利益率化

一方低価格化を行う場合、すり合わせ型の製品アーキテクチャは部品間の相互調整作業やまとめ作業が多く必

要なため、コストがかさみ開発スピードも遅くなってしまふ。そこでこちらも、単純組み立て型の製品アーキテクチャを採用するようになる。単純組立型の製品の品質はすり合わせ型より低いものの、顧客の期待価値はそこそこのレベルで満たしている。したがって単純組み立て型のアーキテクチャに従うと、低い技術力でそこそこの満足度の製品が開発できるため、多くの企業が市場に参入する。するとコスト競争が発生し、疲弊しきるまで企業体力の勝負となってしまう。いわゆるBTO (Built To Order) 企業に代表される最適調達・最適組み立て・最適オペレーションによる低利益率開発を得意とする組織は競争に勝ち残るが、技術力に溺れ市場変化に気付かず高コストの開発を続ける組織には撤退ないし組織改革という選択肢が眼前に突きつけられることとなる。もちろん、すり合わせ型で製品を開発してきた組織が最適オペレーション型の組織に変貌するのは、生易しいことではない。したがって撤退が現実のものとなってしまう。

それでも、ごく一部の組織は技術力を活かし高い利益率を獲得し続ける。それは、単純組み立て型製品において製品の機能や品質を大きく左右する寡占的な高付加価値コンポーネントを開発する組織である。PCを例にとると、PC本体、そしてメモリなどの価格競争的コンポーネントの開発組織の利益率は低いが、インテルのCPUの利益率は高い。こうした高付加価値コンポーネントはすり合わせ型製品アーキテクチャを採用し、他の組織の追撃を許さぬよう性能や信頼性など品質向上にいそむこととなる。

2.4 高い利益率を得るための品質向上戦略

本章の議論を製品アーキテクチャに着目してまとめると、以下ようになる。顧客が価値を感じる機能において、顧客要求よりも製品品質が下回っている場合は、すり合わせ型製品アーキテクチャを採用して迅速に品質を向上させる組織が競争に生き残る。一方、顧客要求を製品品質が上回ると、新機能追加および低価格化を実現するために、全体として単純組み立て型製品アーキテクチャに移行する。ここで組織をドラスティックに改革し、単純組み立て型アーキテクチャ製品の開発に特化するとコスト競争から撤退せずすむ。しかし対照的に、新機能コンポーネントや高付加価値コンポーネントに焦点を当て、すり合わせ型製品アーキテクチャを採用し品質向上に注力する組織は高い利益率を得る。

したがってコスト競争に陥らず高い利益率を得るためには、3つの開発力を鍛えることで品質を向上させていく必要がある。一つ目は、新機能コンポーネントや低付加

価値コンポーネントを迅速に組み込めるよう全体的視座に立った「アーキテクチャ構築力」である。二つ目は、中長期的な顧客価値の変化を敏感に察知して追加すべき高付加価値コンポーネントを見抜く「顧客価値創造力」である。もう一つは、高付加価値コンポーネントを迅速に高い品質でコストをかけず開発できる総合的な「すり合わせ力」である。この3つが、高い品質技術を持つ開発組織を過剰品質に陥らせず、高い利益率を達成する品質向上戦略である。特に、組み込みソフトウェアが大きな役割を担うような複雑な製品であるほど、3つの開発力が利益率に寄与する割合は高い。

3. 組み込みソフトウェアの品質向上に必要なとされる3つの技術

組み込みソフトウェアは高度に複雑な製品の中核を担う一方で、自分自身が高度に複雑な構造を備えている。そのため前章で提示した3つの開発力の向上が重要となる。本章ではアーキテクチャ構築力、顧客価値創造力、すり合わせ力という3つの開発力に対応させながら、組み込みソフトウェアの品質向上に必要な技術について述べる。

3.1 組み込みソフトウェアのアーキテクチャ構築力

組み込みソフトウェアのアーキテクチャ構築力には、大きく分けて2つの種類がある。ハードウェアとの相互依存関係についてのアーキテクチャと、ソフトウェア内部についてのアーキテクチャである。組織によっても異なるが、前者を定める開発フェーズをシステム設計、後者を定める開発フェーズをソフトウェア基本設計と呼ぶ。

システム設計では相互依存関係を考慮しながら、製品全体で実現すべき機能を主にメカニカル部、エレクトロニクス部、ソフトウェア部の3つ、およびデジタルカメラや半導体検査装置における光学部といった製品固有の技術要素に切り分けていく。同時に、それぞれのバランスや相互作用を考慮しつつ、全体としての大きさや重さ、消費電力、熱の流れなどが最適になるよう調整していく。

したがってメカ部、エレキ部、ソフト部、製品固有技術部が切り分け後の作業を円滑に進められるよう独立性の高いアーキテクチャを構築しながら、製品全体の最適設計のためのすり合わせ作業を行うという高度な技術力が必要となる。しかしメカとエレキ、エレキとソフトなど一部に関するシステム設計の専門書やツールは存在するものの、メカ、エレキ、ソフトの全てを俯瞰した技術分野として確立されているわけではない。システム設計は日本の高い技術力が活かせる分野であると考えられる

ので、産官学を挙げて技術発展に取り組む必要があるだろう。

一方、ソフトウェア内部のアーキテクチャ構築は注目を集めており、先端的な開発組織では盛んに議論され適用されている。ただし企業情報システムのようなエンタープライズ・ソフトウェア開発と異なり、組み込みソフトウェアのアーキテクチャを構築する際に重要なのは、単一のソフトウェアにおけるアーキテクチャではない。むしろ多くの派生製品、要求品質の異なる製品グレード、中長期的な製品の世代といった製品ファミリを体系的に整理できるアーキテクチャ群を構築する技術である。

これはプロダクトライン・エンジニアリング^[5]と呼ばれ、議論や適用が始められている。また典型的なアーキテクチャの種類をデザインパターン^[6]として蓄積し適用する技術も成熟してきている。今後一層多くの技術者への普及を強く推進しなければならない。

3.2 組み込みソフトウェアの顧客価値創造力

組み込みソフトウェアの顧客価値創造力に必要なのは、マーケティング技術との融合である。プロダクトアウトによって機能を追加したり品質向上を図るだけでは、新たな顧客価値は創造できない。既存の価値の枠組みを抜け出すことはできず収斂してしまうが、技術陣の独りよがりや抜け出すことはできないからである。その一方でマーケットインの名の元に、技術の進化や深化を見据えずに市場のニーズを分析しても、競争に生き残れるスピードと付加価値を両立できる製品は開発できない。製品の方向性を具体的に定めることができないからである。

重要なのは、要素技術の進化や深化を見定めた上で、多種多様な顧客価値について豊富な洞察を行い、顧客に新たな価値観やライフスタイルを提示することである。Web ブラウザやメール機能のついたインターネット冷蔵庫がほとんど普及しなかったのは、家庭の主婦のライフスタイルに合致しなかったからである。カメラ付き携帯電話がこれほど普及したのは、携帯電話を買えばデジタルカメラが付いてくるからではなく、身の回りの出来事や心の変化を画像として家族や友人に送るというコミュニケーションスタイルが提示されたからであろう。両者の違いは技術力ではなく、顧客価値創造力に他ならない。技術陣がこのような顧客価値について豊富な洞察を行うことは容易ではないが、従来から顧客満足について議論を続けてきた品質技術者であればマーケティング技術と製品技術の架け橋になりうるであろうし、なることが期待される。そのためには品質管理界において、より濃密に顧客価値と融合した深遠なる「質」概念の拡張をより

深く議論していかなければならない。

3.3 組み込みソフトウェアのすり合わせ力

組み込みソフトウェアのすり合わせ力にまず必要なのは V&V 技術 (verification and validation, 検証技術) である。ソフトウェア開発において最も特徴的な要因の一つは、汎用部品であるはずのオープン・コンポーネントの複雑性が高く信頼性が低いため、多くのすり合わせ作業が必要となる点である。まずすり合わせ作業が必要かどうかを判定するために、テストを始めとした V&V 技術を駆使することでコンポーネントの信頼性を迅速かつ的確に評価する必要がある。またすり合わせ作業中に頻繁に実施されるトライアル・アンド・エラーにも、レビューやインスペクションといった V&V 技術が必須である。そしてすり合わせ作業を実施した後にも、やはり V&V 技術を活用してきちんとすり合っているかどうかを判定しなければならない。しかし組み込みソフトウェア開発における V&V 技術は、十分成熟しているとはいえない。4 章では、V&V 技術について詳細に述べる。

またすり合わせ作業には、高い組織能力が必要となる^[4]。特に「きめ細かいチームワークや連携調整を得意とする組織能力」が重要である。組織能力を高める取り組みのことをソフトウェア開発ではプロセス改善と呼び、CMMI^[7]というフレームワークを採用している例が多い^[8]。CMMI には段階表現と連続表現の 2 つがある。段階表現は組織のレベルを 5 段階に分けて改善を促すもので、ISO9000s と概ね同等のレベル 3 から定量的なメトリクスの測定を行うレベル 4、定量的測定結果を用いて改善を進めていくレベル 5 とプロセスが成熟していくモデルである。そしてレベルに応じて取り組むべきプロセス改善項目が定義されている。例えばレベル 2 には構成管理が必要であり、レベル 3 には教育 (組織トレーニング) が必要となる。連続表現は、レベルにとらわれず必要な部分からプロセス改善を行うモデルである。

CMMI などを基にプロセス改善の取り組みを進めている組織は多いが、TQM で提示されているような改善のための諸概念を適切に理解している組織ばかりかどうかは疑問であると言わざるを得ない。マニュアル主義に陥ってプロセス改善の推進部門と現場が乖離する組織もあれば、定量的にメトリクスを測定しているものの改善に結びつかない組織もある。レベル取得という手段が目的化してしまう組織も散見される。しかしこうした組織においてもハードウェア部門に目を向けると、全社の品質保証部門がきちんと機能し TQM を推進していることがある。原因はソフトウェア部門において TQM の浸透が遅

れていることだけでなく、ハードウェアの品質保証部門がソフトウェアの品質技術を十分理解していないこともあると考えられる。組込みソフトウェア開発組織に必要なのは、ハードウェアの品質保証部門と二人三脚でTQMを推進し、ハードウェア・ソフトウェア統合型品質マネジメントシステムを構築することであろう。さもないと、せっかくハードウェアで蓄積した改善のノウハウをいかすことができず、いつまで経っても組込みソフトウェア・クライシスから抜け出すことはできない。

4. 組込みソフトウェアの信頼性保証

組込みソフトウェアの品質向上には、前章で列挙したような多くの技術が必要となる。さらに1章で述べた「組込みソフトウェア・クライシス」から脱却するためには、顧客が安心して製品を用いるための品質保証の仕組みや技術が必要である。本章では特に、組込みソフトウェアの信頼性の保証の核となるV&V技術について述べる。

4.1 組込みソフトウェアの信頼性保証の核となる

V&V技術

ソフトウェア開発の最も大きな特徴は、大量生産工程が無いことである。そのため信頼性を全て設計（およびプログラミング）工程で作り込まねばならない。工程そのものも人間であるため、非常にばらつきが大きい。さらにソフトウェアは論理的実体であるため目に見えず、品質を如実に示す特性（メトリクス）を定めるのが非常に難しいという特徴がある。したがってハードウェアの設計工程から類推できるように、抜き取り検査を始めとした統計的手法による信頼性の保証は非常に難しい。

したがって信頼性を保証するには、以下の3点に頼らざるを得ない：

1. 特定の不具合を検出する静的解析
2. 要求事項や設計、プログラムを机上で検討するレビューや精査（インスペクション）
3. 実際に動作させて不具合を検出するテスト

以上を総称してV&V技術と呼ぶ。組込みソフトウェアの信頼性を保証するためには、核となるV&V技術が成熟していることが必要となる。

4.2 未成熟なV&V技術

静的解析では、特定の不具合そのものや不具合の可能性の高い部分が固有の解析技法で検出できる。代表的な例^[9]としては、制御フロー解析、データフロー解析、複雑度解析、カバレッジ解析などが挙げられる。静的解析

は対象とする不具合に対する検出力は高いものの、あらゆる不具合に対する信頼性を保証することはできない。したがって信頼性の保証という観点では、あくまで補助的な手段に留まることとなる。

レビューやインスペクションは、ソフトウェアが動作可能な状態になる前、すなわち設計やプログラミングの工程で不具合を検出する技術である。開発の上流で不具合を検出し悪影響の伝播を防ぐために、極めて重要な役割を担う。しかし多くの組織では、経験の豊富な技術者を集めて不具合を指摘する、もしくは非体系的に蓄積されたチェックシートを用いているに留まっている。ソフトウェアの信頼性を確保するには非常に重要な技術であるにもかかわらず、活発に研究されているとは言い難い。本来は、書式や一貫性、整合性のチェック、論理展開や列挙、詳細化などに関する開発者の思考のトレース、過去に発生した不具合やスキルの高い技術者の経験を抽出・構造化・蓄積することによる不具合のパターンの認識、という3つの指摘が必要となるだろう。こうした不具合の指摘項目の設計の体系化を進め、勘と経験による指摘からいち早く脱せねばならない。

テストは、ソフトウェアを実際に動作させて不具合を検出する技術である。市場への不具合の流出を防ぎ信頼性を保証する最後の砦であると言える。ソフトウェアは非常に複雑な構造をとることが多いため、単なる動作確認としての単純作業ではなく、緻密かつ網羅的でありながら重点指向の高度な設計作業が必要となる。組込みソフトウェアは高い信頼性を要求されることが多いため、テスト工数はソフトウェア開発全体の3割から9割にも及ぶ。したがって、より危険な不具合を、より早く、より少ない工数で検出しなくてはならず、多岐に渡るテスト技術が必要となる。しかし信頼性を十分保証できるほど整備されていないのが現状である。また組込みソフトウェアに特徴的な、ハードウェアとの協調処理や割り込みなどのリアルタイム制御に関するテスト技術が、現場で適用可能なほど成熟していない点も大きな問題である。

また組込みソフトウェア開発のトレンドに目を向けると、Linuxを始めとするオープンソース・コンポーネントや、インド、中国に代表されるオフショア開発が挙げられる。しかしこちらにおいても、信頼性の保証が問題となっている。オープンソース・コンポーネントは世界中のボランティアにより開発されているため、体系的な信頼性の保証が行われていない傾向が高く、また自社向けに変更した際の保証の工数やコストが大きな負担となる。オフショア開発では文化的な差異を埋めるために、

定量的な信頼性の基準を定め詳細な作業指示を行う必要がある。しかしレビューやテストなどの V&V 技術が成熟していないため難しく、結果として信頼性問題が多発することが多いという問題を抱えている。

4.3 V&V 技術の動向

組込みソフトウェアに限らずソフトウェア全般の信頼性の低下が多く問題となっているため、V&V 技術に関する取り組みは近年徐々に目立つようになってきている。欧米では ISSTA(International symposium on Software Testing and Analysis)や Quality Week を始め V&V に関する国際会議が開催されている。国内も JaSST(Japan Symposium on Software Testing)や日科技連・SPC シンポジウムなどで議論が活発になりつつある。しかしまだまだ成熟というには遠いのが現状である。

自動車や航空機といった高信頼性を要求される産業でも、V&V 技術が未成熟だという状況はさほど変わっていない。しかし徐々に V&V 技術を高めようとする動向が見られる。例えば欧州の自動車業界では、C 言語によるソースコードの潜在不具合を検出するための 127 項目に渡るチェックリストを作成している。これは MISRA-C^[10]と呼ばれ、国際的・産業横断的な普及が進められつつある。米国の航空機業界では、DO-178B^[11]という基準でプログラムのロジックを適切に網羅してテストすることを定めている。しかしこれらは信頼性を保証する上で必要な観点のごく一部についての基準でしかない。信頼性を保証するための包括的な技術指針を早急に整備する必要があるだろう。

我が国においては、経済産業省の主導により情報処理推進機構に「ソフトウェア・エンジニアリング・センター」が設置され、産官学が集まって組込みソフトウェアの信頼性の向上について議論を始めたばかりである。また学会や技術者コミュニティも活発に活動を始めている。しかし上に述べたように、品質向上や信頼性保証という点においては問題が山積している。ハードウェアにおける豊富な品質向上の蓄積を活かし、組込みソフトウェアの信頼性保証技術を進化させ、世界に先駆けてハードウェア・ソフトウェア統合型品質マネジメントシステムの構築を目指すために、日本品質管理学会は積極的に活動すべきであると筆者は強く主張したい。

5. おわりに

組込みソフトウェアは、高度化し複雑化した製品の中核となっており、製品全体の品質が組込みソフトウェア

に依存するようになってしまった。しかし残念ながら組込みソフトウェアの品質、特に信頼性は十分確保されているとは言えない。いわば「組込みソフトウェア・クライシス」である。したがって日本の製造業の産業競争力を強化するには、組込みソフトウェアの品質を向上する必要がある。そこで重要なのは、高い技術力を有効に活かしてコスト競争に陥らないようにする戦略である。

本稿ではコスト競争に陥らず高い利益率を得るために組織が備えるべき開発力としてアーキテクチャ構築力、顧客価値創造力、すり合わせ力の 3 つを提示し、具体的な技術としてシステム設計、プロダクトライン、デザインパターン、マーケティング技術との融合、V&V 技術、ハードウェア・ソフトウェア統合型品質マネジメントシステムを推し進めることが有効であると述べた。また信頼性を保証するためには V&V 技術を成熟させ、ソフトウェアそのものの信頼性や、組込みソフトウェアに特有なハードウェア協調部、リアルタイム制御部、分散処理部の信頼性、そして最近注目されているオープンソースソフトウェアやオフショア開発における信頼性を確保する必要があると指摘した。

組込みソフトウェア製品という一寸法師に打ち出の小槌を振らせることができるかどうかは、組込みソフトウェアの品質向上について真剣に議論し、取り組むかどうかにかかっている。我が国の産業競争力を向上し、欧米やアジアとの熾烈な競争を中長期的に勝ち抜くには、産官学の総力を挙げて組込みソフトウェアの品質向上に取り組まなければならないのである。

参考文献

- [1] 組込みソフトウェア開発力強化推進委員会(2004): 「2004 年版組込みソフトウェア産業実態調査報告書」, 経済産業省.
- [2] 大豆生田崇志・井上理(2004): “携帯、家電に不具合相次ぐ”, 日経ビジネス, 7月5日, 6-7.
- [3] Christensen, C.(2003): *The Innovator's Solution*, Harvard Business School Publishing, MA.
- [4] 藤本隆宏(2004): 「日本のもの造り哲学」, 日本経済新聞社.
- [5] Clements, P. and Northrop, L.(2001): *Software Product Lines: Practices and Patterns*, Addison Wesley, MA.
- [6] Gamma, E., Helm, R., Johnson, R. and Vlissides, J.(1997): *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley, MA.

- [7] Chrissis, M., Konrad, M. and Shrum, S. (2003): *CMMI: Guidelines for Process Integration and Product Improvement*, Addison Wesley, MA.
- [8] 竹居智久(2004):“ デジタル家電のソフト開発不具合ゼロへの始めの一歩 ”, 日経エレクトロニクス, 7月5日, 57-64.
- [9] Software Engineering Coordinating Committee (2001): *Guide to the Software Engineering Body of Knowledge - SWEBOOK*, IEEE, CA.
- [10] The Motor Industry Software Reliability Association(1998): *Guidelines for the Use of the C Language in Vehicle Based Software*, The Motor Industry Research Association, Warwickshire.
- [11] the Radio Technical Commission for Aeronautics (1992): *DO-178B: Software Considerations in Airborne Systems and Equipment Certification*, RTCA, NW.

* * *

本論文は、「品質」(品質管理学会誌, “Journal of the Japanese Society for Quality control”), pp. 45-51, Vol.34, No.4, 2004. に掲載されています。