

# 自動車組み込みソフトウェアの現状と動向

菅 沼 賢 治

(株)デンソー 電子技術 1 部

環境・安全・快適性への対応が求められている自動車システムの高機能化に伴い、自動車組み込みソフトウェアはその規模が増大化した複雑化している。自動車組み込みソフトウェア開発では増大する業務量に対応するために、ソフトウェアアーキテクチャ、ソフトウェアプロセス、開発環境などのソフトウェア開発技術の観点からの総合的な取り組みが必要となっている。本論文では、パワートレイン ECU ソフトウェアを事例に、自動車組み込みソフトウェアの特徴と課題について報告する。

## Technology Trends in Automotive Embedded Software

Kenji Suganuma

Electronics Engineering Department 1, DENSO CORPORATION

The automotive embedded software has been explosively expanding its size and becoming complex due to the new functions of automotive control systems required for environment management, safety, comfort and convenience. Under the circumstance, the efficiency of the automotive embedded software development is to be improved. The viewpoints of software engineering, such as architecture, software process and development environments, should be taken into account for overall improvement activities. This paper describes the features and problems to be solved in the industry for the software through an example of Power train ECU software developments in DENSO.

### 1. はじめに

自動車業界では、環境・安全・快適性への追求から機能の大規模化や複雑化が進む一方で、厳しい事業環境からますます開発期間短縮の要求が高まっている。これらを背景として、自動車用ソフトウェアを取り巻く環境は大きな変革期を迎え、市場競争力を確保していくためにソフトウェア開発技術にも大きな変化が求められている。

従来、自動車用システム開発においてソフトウェア技術はあまり重視されてこなかった。自動車用システム開発は元々システム開発プロセスを主体としており、その中のプログラムは最後の最後で仕様変更の帳尻をあわす便利な道具でしかなかった。ソフトウェア開発者側からみれば次から次へと行われる仕様変更に対応するだけで新しい技術の導入どころではなかった面も否めないが、高い信頼性と低コストを要求されるソフトウェア開発に新しい技術を適用することに慎重にならざるを得ないことも、その導入を遅らす一因となっていた。

しかしながら、1990年代から自動車用ソフトウェアの規模は飛躍的に増大した。それまでの少人数の開発から、数十人のチームで行われるソフトウェア開発も珍しくなくなってきた。このような状況下では、単に技術的な面だけでなく、組織管理的な面からも様々な問題が発生する。このために、我々は1990年代に入っていくつかの取り組みを本格化してきた。

本稿では、自動車組み込みソフトウェアについて解説した上で、これらの活動とその基本的な考え方を紹介するとともに、自動車組み込みソフトウェアにおける今後の課題を整理してみたい。

### 2. 自動車組み込みソフトウェアの動向と特徴

#### (1) 自動車組み込みソフトウェアとは

自動車組み込みソフトウェアとは、自動車に搭載されている様々な制御システムを制御する ECU (Electronic Control Unit) と呼ばれる車載コンピュータに組み込まれるソフトウェアである。自動車用制

御システムは、エンジンやトランスミッションなどの駆動系を制御するパワートレイン制御、サスペンションなどの走行系を制御するシャシ制御、ドアやエアコンを制御するボデー制御、ナビゲーションや自動車電話を制御する情報通信、の各分野に分類される<sup>[1]</sup>。それぞれは ECU によって制御され、ほぼ全ての ECU にマイコンが搭載されている（図 1）。



図1. カーエレクトロニクス製品群

図 2 にカーエレクトロニクスの歴史を示す。自動車にエレクトロニクスが用いられたのは、1970 年代に米国で施行されたマスキー法と呼ばれる排ガス規制を契機としている。その後、マイコンが用いられ自動車組み込みソフトウェア時代の幕開けとなった。



図2. カーエレクトロニクスの歴史

個々の ECU は実装技術を主体としたエレクトロニクスの発展に伴い機能を拡大していったが、1990 年代より車載ネットワークで機能統合が始まっている。車載ネットワークでは、制御系、ボデー系、情報系の各系をゲートウェイで結び、車両統合システムへと発展している。

今後は、ITS 等、社会インフラともネットワークで結ばれさらに規模を増大していくことが計画されて

いる。従って、技術的にも汎用のソフトウェア技術との境界がなくなり、さらに高度な技術導入が必要となることが予想されている。

## (2) 自動車組み込みソフトウェアの特徴

特に自動車分野でのソフトウェアは、以下のような特徴を持っている。

- ・ハードリアルタイム性
- ・省リソース
- ・高い品質要求
- ・自動車メーカーと ECU サプライヤによる分担開発
- ・頻繁な仕様変更とバリエーションの多さ

はじめの二つは組み込みソフトウェアに共通する特徴であるが、それ以外のものは自動車分野に特有のものと考えられる。

### ■ ハードリアルタイム性

ハードリアルタイムシステムとは、リアルタイムシステムのなかでも特に厳格に時間制約を守ることが必要なシステムである<sup>[2]</sup>。一定時間内に処理を完了しない場合、その処理自体に意味がなくなってしまうり、場合によっては危険が生ずることになる。自動車の運行を司る制御システムでは、搭乗者を危険にさらすような致命的な故障を全て回避する必要があり、このハードリアルタイム性が求められる。

多くの場合、自動車用の制御システムは、エンジンなどの機械を制御対象としている。例えば、ガソリン噴射や点火などタイミングが重要であり、期限を過ぎて処理を行うことに意味のないことが多い。また、ブレーキを電子的に制御するブレーキバイワイヤ等、搭乗者を危険にさらすような致命的なフェールを回避しなければならないシステムでは、より厳格なハードリアルタイム性が要求される。最近では、このようなシステムをネットワークで構成するための技術として、TTP/C、FlexRay など、自動車分野でも時分割多重 (TDMA) 方式による通信プロトコルが欧州を中心に提案されている。

### ■ 省リソース

自動車用 ECU に搭載されるマイコンは、高い信頼性要求やコストの面から、そのリソースが厳しく制限される。通常、自動車は厳寒の地から灼熱の地での使用が想定され、また 10 年以上の耐久性が要求されるため、使用するマイコンにも同等の品質が要求される。また、マイコン単価も非常に低く抑える必要がある。このため、マイコンのリソースを効率良く利用するた

めの様々な工夫が必要となっている。

### ■ 高い品質要求

自動車組み込みソフトウェアで最も重要な品質特性 (ISO/IEC9126) は、信頼性である。自動車の運行時に障害が発生することは通常許されず、仮に障害が発生した場合でも致命的な故障とならない様な仕組みが必要である。すでに触れたが、限られたリソースを有効に活用するために、効率性も重要な品質特性である。

### ■ 自動車メーカーと ECU サプライヤによる分担開発

自動車用制御システムの開発は、自動車メーカーと、制御システムや ECU のサプライヤとの共同で行われる。

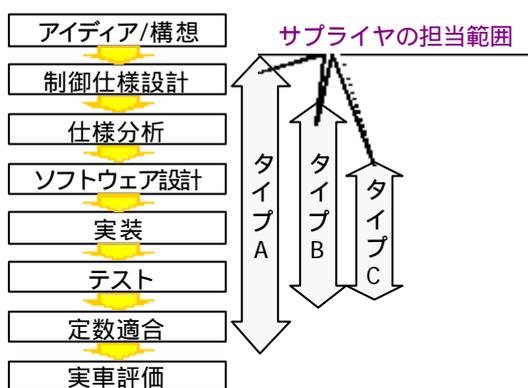


図3. 制御システム開発プロセスと分担開発の例

自動車メーカーは複数のサプライヤとの共同開発を行うと同時に、大手のサプライヤも複数の自動車メーカーとの共同開発を行っている。この場合の役割分担は、対象となるシステムに応じた様々なパターンが存在する。相互の情報伝達を正確にかつ迅速におこなうために、要求仕様はそれぞれのパターンに応じた適切なものとする必要がある。

### ■ 頻繁な仕様変更とバリエーションの多さ

車両開発は何段階かの試作を積み重ねることで進められる。各試作フェーズでは実機での検証を行い、より精密な制御仕様が固められていく。実機検証で見られた不具合はその都度修正され、さらに検証が進められる。このため、自動車組み込みソフトウェア開発では、頻繁な仕様変更に対応することが求められる。

また、仕向地やオプション設定によるバリエーションも多い。自動車は世界中の国々に輸出されるため、その国々の法規制等に対応する必要がある。また、同

じ車両においても、オプション設定があり、これら全てに対応する必要もある。製品によっては年間に出荷されるソフトウェアの本数は数千本にも上る例もある。

## 3. 自動車組み込みソフトウェア開発における新たな取り組み

急激に大規模化・複雑化していくソフトウェア開発は、爆発的な業務量の増大を引き起こす。自動車組み込みソフトウェアでも1990年代後半より、その規模は従来に比べ急激に増大している。限られた経営資源でソフトウェア開発を行うには、効率化が必須である。このような状況下で、1990年代に入ってエンジンとトランスミッションを制御するパワートレイン ECU ソフトウェア開発を対象に新たな手法を導入する取り組みを行ってきた。ここでは、その事例をもとに、具体的な取り組みの考え方とその概要について述べてみたい。

### (1) 取り組みの背景

新たな取り組みが必要となった背景には、市場の要求から来る外的な要因と、それらに対応する上での開発組織に内在する要因とがある。

#### ■ 外的な要因

… ソフトウェアの大規模化・複雑化、統合化

パワートレイン制御システムでは、法規制の強化や車の運動特性に対する制御性の向上を目的とした機能統合が進んでいる。これに伴いパワートレイン ECU のソフトウェア規模も増大している。図4に示す様にプログラムサイズは1メガバイトに迫っており、ソースコードの総ライン数はすでに数万行を越えている。さらに、機能統合に向け、ネットワーク化に対応するための新しい技術導入も必要となっている。

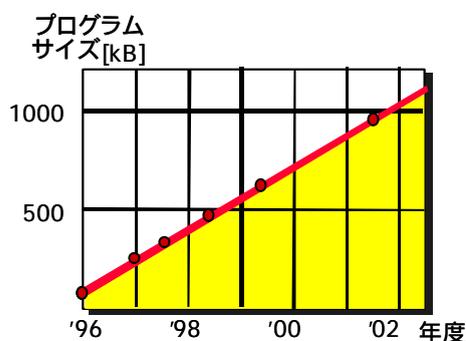


図4. パワートレイン ECU でのプログラムサイズ

“ 開発期間短縮への要求と市場競争力の確保

厳しい事業環境の中、タイムリーに商品を市場に投入する必要があるのは自動車業界も例外ではない。市場競争力を確保していくためには、自動車メーカの厳しい開発期間短縮に耐えていかなければならない。

■ 内的な要因

“ ハードウェア製品に依存した品質保証体制

開発プロセスや品質保証の仕組みはハードウェア製品を対象に長年かけて構築されてきた。しかし、新しい技術であるがゆえにソフトウェアの開発プロセスへの適用は始まったばかりである。

“ 開発者個人に依存したソフトウェア開発

従来の自動車組み込みソフトウェア開発は、規模が比較的小さく個人或いは数人といった小規模の開発チームで行われる時代が長かった。このため設計情報やノウハウ、開発プロセスは開発者個人個人に依存している部分が多い。

“ デバッグ中心の品質確保

設計プロセスが十分に確立しておらず、とにかくものを作って動作を確認しながら修正を施すといったデバッグ中心の品質確保が主体となっている。

(2) 自動車組み込みソフトウェア開発に求められるもの

自動車組み込みソフトウェア開発は、「類似多品種の大量生産型ソフトウェア開発」と言える。様々な仕向け地やオプション設定に対応した類似のソフトウェアを開発するとともに、長期的には新しい機能追加とともにインクリメンタルな開発が繰り返される。このような特徴を持つソフトウェア開発を高生産性と高品質を両立させかつ組織的に行うためには、徹底的な再利用が有効と考えられる。

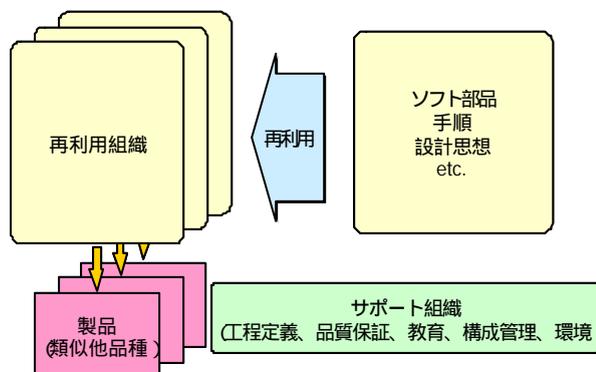


図5. 再利用の考え方

再利用は、単にコードレベルに留まらずさらに利用

価値の高いプロセス、設計情報、ノウハウ、コンセプトといった全ての技術情報が対象となる。また、再利用を組織的に行うために、優れた技術情報を集約し、全体組織の利用を促進する機能も必要となる。

(3) 組み込みソフトウェア開発手法

再利用をベースとした新たな取り組みを導入する上で、技術的な観点からパラダイムシフトが必要である。従来の小規模ソフトウェア開発では、サイズや性能達成のための技術が重要で、主眼は言語ベースの実装技術であった。しかしながら、大規模化にともない再利用を促進する上では、拡張性や保守性を考慮した分析や設計など上流工程が重視されるべきである。

これらを実現するために、オブジェクト指向技術を適用し、部品化構造を持ったソフトウェアアーキテクチャを導入している。

	小規模ソフト	大規模ソフト
狙い	サイズ・性能の達成	拡張性、保守性
開発技術	アセンブラ C言語	分析・設計手法 工学的なソフト構造
観点	コードチューニング	標準化を睨んだ技術戦略

図6. 規模による開発技術の違い

最近になって、組み込みソフトウェアへのオブジェクト指向の適用事例が報告されるようになってきた<sup>[3]</sup>。現在、一般に用いられている方法論は1990年代後半から確立されてきたUMLの適用が主流となっている。しかしながら、取り組みの初期はまだいくつかの方法論が乱立している状況であった。このため、当時リアルタイムシステムに比較的適していると考えられたOMT法とMVCアーキテクチャをベースに多少修正を施した方法論を適用している。

また、このようなパラダイムシフトを実現するためには、部品化構造を持ったソフトウェアアーキテクチャを定義するだけでは不十分で、組織として均一な開発を定着させる仕掛けが必要となる。仕事の仕方と役割分担を定義するためにソフトウェアプロセスの再構築が必要となる。また、技術やプロセスを組織的に支援する開発環境の導入も並行して必要となる。

■ アーキテクチャ

再利用のためのアーキテクチャは、オブジェクト指向技術をベースとした再利用単位での階層化・部品化の考え方により構成されている。

前述したように、自動車組み込みソフトウェア開発は、インクリメンタルな繰り返しにより行われるといった特徴を持っている。従って、再利用性を高めるためには、変更に着目した単位での部品化を行う戦略が有効である。ソフトウェア部品の単位としては、主に変更やバリエーションの単位を想定している。変更は、アプリケーションの制御アルゴリズムやバリエーション展開に対してのもの、ハードウェア部品の変更には依存しているものがある。前者は頻繁に行われる傾向を持ち、後者は比較的安定しているといった特徴もっている。

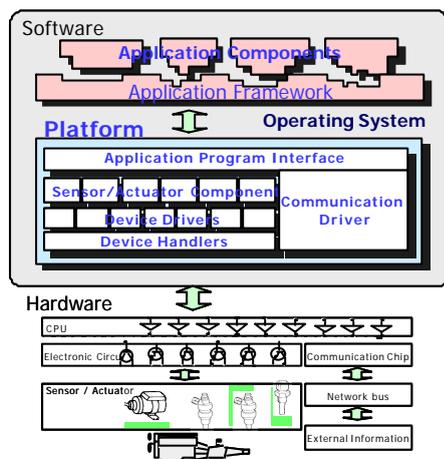


図7. パワートレイン ECU のソフトウェアアーキテクチャ

図7は、パワートレイン ECU ソフトウェアでのアーキテクチャの一例である。全体はハードウェア部品構成をちょうど鏡で写像した構成としており、制御対象と駆動部分に対応した部分をそれぞれアプリケーションとプラットフォームとしている。

エンジン制御は 2-30 年の歴史をもっており、基本的な制御方式はすでに成熟している。このため、アプリケーションの制御の基幹部分はアプリケーションフレームワークとして構成し、比較的ホットな部分をアプリケーションコンポーネントとして括り出している。従って、制御の基本的なことをあまり考慮しなくても、部品を組み込むだけでシステムの構築が可能となっている。

プラットフォームは、センサー・アクチュエータ部品、それらインターフェースを取る ECU 電子回路のドライバ部品、マイコンおよび周辺デバイスのドライバ部品といった、アプリケーション比べ比較的安定した部品から構成され、API を通してアプリケーション

にサービスを提供する。また、アプリケーションからみたときの通信を隠蔽するための機構も導入している。

## ■ プロセス

ソフトウェア規模の増大に伴い、必然的に少人数開発からチーム開発に移行する。この場合、組織の効率性を高めるためには、分業体制を構築しそれぞれの専門性を高めることが重要である。

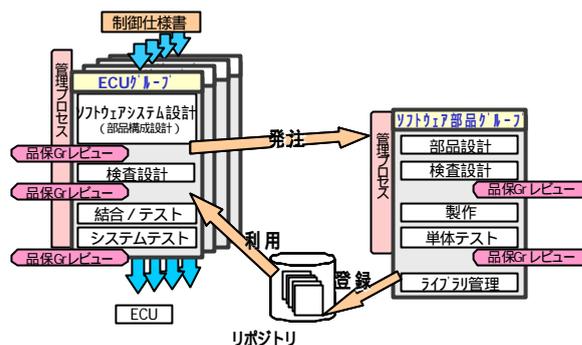


図8. 部品化による分業体制

パワートレイン ECU ソフトウェア開発組織に対し、図8に示すような組織の枠組みを構築した。従来、コードをベースに開発を行っていた ECU 開発チームに対し、原則としてコーディングを禁止し、部品によるソフトウェア開発を促進させている。一方、部品を開発するチームは一つに集結させ、必要となる技術やノウハウを集約させた。これにより導入当初は多少の混乱をみせたが、現在では部品による開発が徐々に進行している。

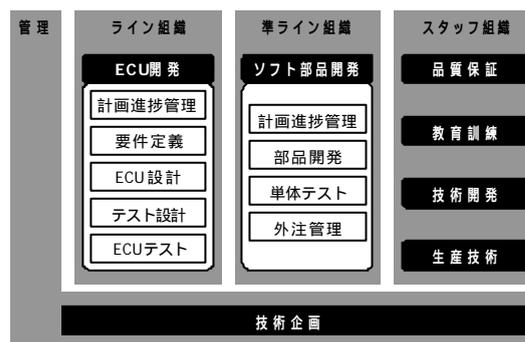


図9. ソフトウェアプロセス

図9にソフトウェアプロセスの概要を示す。ソフトウェアプロセスは SLCP-JCF94<sup>[4]</sup>を基本としている。ソフトウェアプロセスとして、ライン組織、準ライン組織に対応した開発プロセスとともに、サポートを行うスタッフ組織に対する各プロセス、組織全体に対応

した品質保証プロセスの各プロセスを定義している。

… **開発プロセス**

開発プロセスは V 字のリファレンスモデルを定義している。図 10 に示すように、上流を ECU 開発、下流をソフトウェア部品開発と大きく分業を行う。

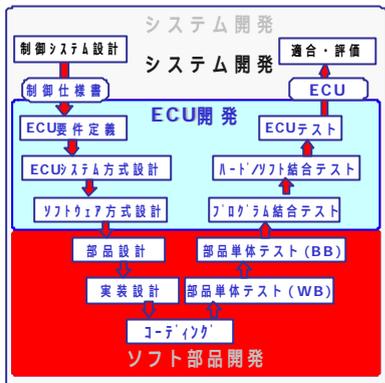


図10. 開発プロセス

開発プロセスの詳細を構築するにあたり、ソフトウェアのアーキテクチャに関連させ設計及びテストの体系化を行っている。図 11 はその概念を示したものである。

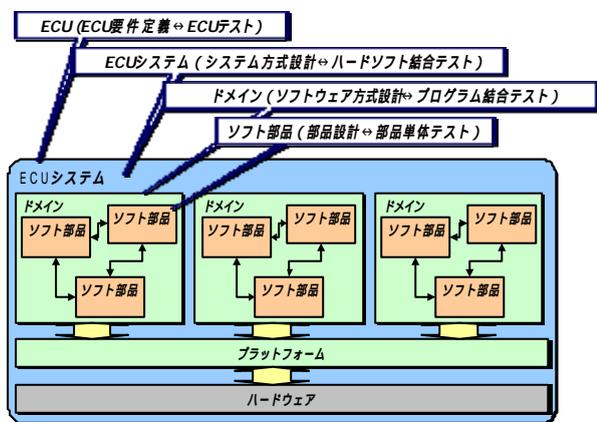


図11. アーキテクチャと工程の関係

ECU はその構成により、ECU、ECU システム、ドメイン、ソフトウェア部品の各階層に分離できる。実際の開発では、これらの単位で設計及びテストが行われる。パワートレイン ECU ソフトウェア ECU 開発では、要件からハード・ソフトのシステム設計から順次部品構成まで設計がなされた後、エンジン制御ドメイン、トランスミッション制御ドメイン、診断システムドメイン等のアプリケーションドメイン毎の部品結合を行い、さらにドメインユニットと ECU ハードウェアを結合し、システム統合する方法が取られる。

このような理由から、各階層に対応した工程を設定し、それぞれ設計要素と対応するテスト要素を定義している。

… **品質保証プロセス**

他方、分業を進めると情報伝達のミスに基づく品質の低下が懸念される。これは工程及び工程間の情報を厳密に定義し、管理することで防止する方策をとっている。このためソフトウェア品質保証プロセスを定義し、その上で品質保証グループによるプロセス監査を実施している。

品質保証プロセスは、主に、レビュー、テスト、部品品質管理の 3 要素から構成される。レビューは、特に設計工程での品質の作り込みを目的に、各工程の実施と完了を特定するために行う。テストは階層化された設計情報に対応して体系化している。また、部品品質は特に重点管理対象としており、再利用に向けドキュメントレビュー、単体テスト、及び部品メトリクスによる管理を行っている。

… **プロセスの定着**

高生産性・高品質の両立といった組織目標を達成するために、プロセスを定着させていくことが重要である。このため、生産性・品質の指標と並行して、CMM によるプロセス診断及び改善を導入している。

とかくこの手の指標を導入すると、レベル到達そのものが目的と化してしまいがちである。また、本来の目的を組織全体に周知徹底することもことのほか困難である。推進側と開発側とにねじれが生じない様、地道なコミュニケーションが重要である。

■ **開発環境**

開発環境は、ソフトウェアプロセスを支援するためのものと位置づけている。実際のツール導入にあたっては、プロセス毎の必要性に応じ自社に担保しておくべき領域は自社開発を行い、その他は最適な汎用ツールを利用する方針としている。自動車組み込みソフトウェアでのソフトウェア部品を結合し ECU を開発するプロセスでは、汎用のツールに適したものがないのが実状である。また、開発環境は組織としての開発を高いレベルで均一にし、個々の開発者が全体像を詳細に知らなくても開発を実行できるためのものでなければならない。このような事情から、プロセス全体をサポートするために個々のツールを統合しツールチェーンを構築する必要もある。このような領域に対しては独自のツールを自社開発している。

#### (4) 取り組みの成果

活動の目的は、ソフトウェアの大規模化に向けた再利用による高生産性・高品質の両立であった。

現在までの生産性の効果はほぼ当初の目標を達成する見込みである。

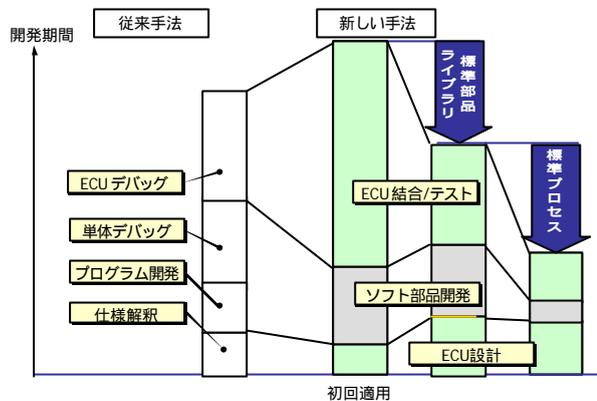


図12. 新手法の生産性向上効果

但し、当然ではあるが、図12に示してあるように、効果を出すに至るまでに初期投資が必要となる。投資に見合うだけの効果が見込めるかは、対象となるアプリケーションの特質に依存する。繰り返しになるが、パワートレイン ECU ソフトウェアは、類似多品種の大量生産型のソフトウェアであった。故に効果を見通すことができた。また、経営的な視点で見たときには、この初期投資に耐えられるだけの経営資源に余力があるかも重要なファクターとなる。同じ目的を持つ自動車組み込みソフトウェア開発では、概ね同様の手法が適用可能であると考えられるが、適用にあたっては投資に対しどの程度の効果が得られるかの慎重な見積もりが必要である。

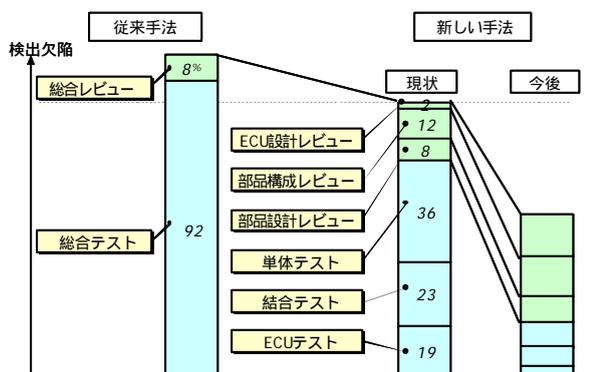


図13. 工程による不具合検出能力

一方、品質に関しては混入不具合件数と検出能力について効果が見られた。今後はさらに上流での欠陥の

検出能力を高めることが課題である。

#### 4. 今後に向けての課題

数年にわたり高生産性と高品質の両立に向けた新たな取り組みを行ってきたが、現実には残されている課題も多い。以下ではそれらを集約するとともに将来に向けた課題を総括してみたい。

##### ■ 開発現場への定着

仕事の仕方を大きく変えることには、やはり大きな困難を伴う。組織的ソフトウェア開発を行う上でソフトウェア部品による標準化やプロセスの定義が有効であるが、それを担うのは一人一人の人材である。現状を変えることへの抵抗や、スキルの問題から、有効な定着を図ることは地道で継続的な活動が必要である。そのためには技術的な取り組みに加え、人材育成の仕組み作りも重要な課題である。

##### ■ 継続的な改善活動の推進と体制作り

継続的な改善についても、地道な活動が必要となる。CMMでも指摘されているように目標を達成するための改善を有効に進めるためには、管理の定量化が必要となる。最終的な生産性や品質の目標に対し、実際の現場において計測可能でかつ上位の目標との関連付けが可能な指標を特定していく必要がある。いくつかの取り組みを試みてはいるが、未だ経験的な予測の域を出していない。例えば品質の定量化ひとつとっても直接最終品質に結びつくようなメトリックが存在しておらず、データを積み上げて経験則を構築する以外の道はなさそうである。

##### ■ オープン化を前提とした組み込みソフトウェア開発手法の確立

組み込みソフトウェア分野においても、オープン化が大きな流れとなっている<sup>[5]</sup>。自動車組み込みソフトウェアにおいても、リアルタイム OS など、外部から調達された部品を組み合わせるソフトウェアを構築するケースが増えている。この場合、ソフトウェア全体の品質保証の手法と同時に、外部から調達された部品品質を正確に評価する手法の確立が必要となる。自動車組み込みソフトウェアに搭載するためには、単に性能が満足するだけでは不十分で、信頼性などの高い品質と低コストの厳しい要求を満足しなければならない。品質とコストのトレードオフを定量的に評価し、ソフトウェア部品の正当な価値を評価することが今後の課題となるであろう。

## ■ 世界標準化への対応

1990年代に入り、自動車組み込みソフトウェアの分野でも標準化の活動が活発化している。

例えば、車載ネットワークシステムでのソフトウェア部品の流通をさせるためのリアルタイム OS・通信プロトコル・ネットワークマネージメントの仕様作りが OSEK/VDX プロジェクト<sup>[6]</sup>により行われている。OSEK/VDX の仕様は、CAN に代表される CSMA/CD 方式を前提とした構成となっていたが、最近では TDMA 方式へと検討の主流が移行している。

開発手法ではモデルベース開発の適用とその標準化が始まっている。モデルベース開発とは、従来アセンブラから C 言語へ、さらにソフトウェア部品へと開発のレベルを転換してきた様に、それをさらにモデルにまで引き上げた開発手法である。背景には、モデルそのもので自動車システムのアルゴリズム開発を行うことによる開発そのものの効率化と、自動車業界全体で仕様記述を標準化することによるプロセスの効率化がある。

モデルの記述方法としては、エンジン制御などのフィードバック制御系ではブロック線図モデルが、ボデー制御などでのシーケンシャル制御系では状態遷移モデルが、それぞれ有望とされている。自動車業界標準化の動きとしては、MAAB (MathWorks Automotive Advisory Board) を中心に進められているモデリングガイドラインの策定<sup>[7]</sup>が一例としてあげられる。一方、自動車組み込みソフトウェアのシステム記述方法としては UML が有望である。自動車分野では、Automotive UML の適用検討が始まっている<sup>[8]</sup>。

今後は、単独で製品開発を行うのではなく、世界の標準化動向を見据えたソフトウェア開発が求められている。

## 5. おわりに

本稿では、パワートレイン ECU ソフトウェアを事例に、自動車組み込みソフトウェアの特徴や課題について報告した。しかし、組み込み系のソフトウェアは日本のソフトウェア産業の中で重要な分野であるにも関わらず、特に自動車組み込みソフトウェア分野での学術的な研究や公開の場での議論があまり活発になされてこなかったように思われる。この分野の特殊性から学術的なアプローチができないのではないかと、の先入観や業界の閉鎖性から、我々現場サイドからの情報公開があまり行われてこなかったことにも一

因があると考えられる。このような機会をきっかけに、層の厚い議論・研究が発展することを期待してやまない。

## 6. 謝辞

今回の発表の場を提供して下さった、南山大学の青山教授に感謝いたします。

## 参考文献

- [1] 水谷 集治 (監修) カーエレクトロニクス研究会 (編著): 新カーエレクトロニクス、山海堂 1992
- [2] 高田 広章: ハードリアルタイムサポートへ向けて - ITRON ハードリアルタイムサポート研究会中間報告 - 16th July 1997
- [3] 杉浦、赤坂、福富、井上、金澤: 特集「組み込み機器開発への導入がすすむオブジェクト指向の本格活用テクニック」、Interface Mar. 2001, pp.95-177
- [4] 大野 豊 (監修): システム開発取引の共通フレーム SLCP-JCF94、通産資料調査会、1994
- [5] 青山 幹雄: 連載ソフトウェア新時代、第四回組み込みソフトウェアの転機、情報処理、Vol.39, No7, pp.588-590 (July 1998)
- [6] OSEK/VDX Operating System, Version 2.1 release candidate 1, 18. January 2000 (<http://www.osek-vdx.org/>)
- [7] Controller Style Guidelines for Production Intent Using MATLAB, Simulink and Stateflow, MathWorks Automotive Advisory Board (MAAB) April 2001 ([http://www.mathworks.com/products/control\\_design/maab.shtml](http://www.mathworks.com/products/control_design/maab.shtml))
- [8] Daimler Chrysler, Research and Technology, AutomotiveUML, (<http://www.automotive-uml.de/>)