組込みソフトウェア品質特性

CATS社
渡辺　政彦

本稿では組込みソフトウェア品質について、CATS社のツールを交えながら考えてみたいと思います。組込みソフトウェアに関する品質の前に、他の分野で品質についてどのような取組みが行われているかを見てみます。例えば最近良く耳にするQOL（Quality Of Life）やQoS（Quality Of Service）について簡単に触れたいと思います。同業者同士よりも異業種の方と会話をしていた時の方が、行き詰まりを打破するようなひらめきを得た経験が皆さんもあるかと思います。遺伝的アルゴリズム（GA：Genetic Algorithms）は生物の世界にある遺伝の法則をまねて作られたものですし、焼きなまし法（SA：Simulated Annealing）とは高温で溶融状態にある物質を徐々に冷却し、欠陥の少ない結晶などの低エネルギー状態を得る方法です。このようなアイディアは、ソフト屋さん同士がどんなに議論しても出てこない発想ではないでしょうか。

医療の世界では、QOL（生活の質）を導入することで、病気にかかって施術後の生活の質が、施術前とかわらないように、出来る限り機能を温存させる治療法を積極的に施すようになっています[1]。身近な例で言えば、従来は虫歯で歯を失ってしまうと義歯かブリッジですが、現在はインプラントがあります。インプラントは、何より安定感に優れているので、自分の歯と同じように何でも噛め、味覚を楽しめます。また、口の中の違和感もなく、審美性も良く、発音障害もありません。食事中などに外れたりしないため、人前でも自信を持って食べたり、笑ったり、会話を楽しめるなどQOLが飛躍的に高まるということです[2]。健康関連QOL（HRQOL：Health Related Quality of Life）を測定する包括的な尺度を提供する健康調査票にSF-36があります[3]。SF-36は８つの尺度36項目で構成されています（図１参照）。このような

図１　SF-36　4

何らかの尺度があってこそ、生活の質を守るという主観的なテーマを、客観的に科学的に進歩させることができるのではないでしょうか。

通信の世界、特にマルチメディア通信を行う場合に音声や動画の品質が問題になります。例えば、現状のインターネットで音声や動画を伝送する場合、声が途切れたり、画像がコマ送り状態になったりすることがあります。このため、ある特定の通信のための帯域を確保し、一定の通信速度を保証することをQoS（サービス品質）と呼びます。ネットワーク上でやりとりされる通信には様々な種類があり、それぞれに求められる品質は異なります（表１参照）[5]。常に最高の品質を追求することは大切ですが、ビジネス上の制約である納期、コストに応じた最適な品質を用意できることが重要です。例えば、郵便が「速達」、「書留」、「配達証明」、「内容証明」等、料金に応じた郵便サービス品質があるように。

表１　マルチメディア情報通信とQoSとの関係

AV QoSレベル	Audio品質	Video品質	帯域	アプリケーション例
1	電話音声	ビデオ・クリップ（1フレーム/秒）	9.6～14.4kbps	マルチメディア・ホームページ
2	電話音声	準動画ビデオ・クリップ	28.8kbps	一人の人物の映像付きスピーチ
3	AM放送	1/4画面擬似動画	28.8～64kbps	マラソン中継
4	AM放送	1/4画面準動画	128kbps	自動車レース映画プレビュー
5	FM放送	1/4画面完全動画 VTR品質準動画	500kbps	映画プレビュー映画
6	音楽CD	VTR品質完全動画	1.5Mbps	映画スポーツ中継
7	音楽CD	テレビ放送品質動画	6Mbps	映画スポーツ中継

ユビキタス時代では、インターネットに接続される組込み機器が多くなります。こうなると機器単体の性能だけでなく、インターネット回線を含めた性能を、機器利用者の立場から十分に検討する必要があります。インターネット上でデータを送受信する際のリソース（CPU、バッファ、帯域などの通信に関するハードウェアやソフトウェアなどの資源）を確保し、QoSを保証するためのプロトコルにＲＳＶＰ（Resource reSerVation Protocol：リソース予約プロトコル）があります。インターネットはルータが相互に回線で結ばれた水平構造をしており、高額な交換機を使わないため、伝送コストが低くなっています。反面、回線品質が一定せず、伝送途中にデータの欠落遅延が起こり得るおそれが常にあります[6]。RSVPでは、送信側パソコンに搭載した「リクエスター」というソフトウェアが、平均及び最大伝送容量等の必要な回線容量をルータ側に指示します。指示を受けたル

1 http://www.ncc.go.jp/jp/ncc-cis/pub/events/990605b.html#04
2 http://www.isahai.jp/implant.htm
3 http://www.i-hope.jp/tool/index.html
4 図の目盛は筆者が適当に入れたものです。
5 通信プロトコル辞典：アスキー出版局：1996：ISBN4-7561-0269-7
6 http://www.bell.jp/pancho/terminology/hyper-dictionary/ohbun/ohbun-r/rsvp.html

ータは基準を満たす回線を選択し、さらに先のルータに伝令します。その結果として、情報を蓄積したサーバから利用者端末まで基準を満たすルートが確保できます。この方式だと帯域は保証されますが、RSVPが普及するにつれて回線を奪い合うことになり、回線不足を招くことが指摘されています[7]。

ソフトウェアの品質に関しては、「ISO/IEC9126ソフトウェア品質特性」という国際規格があります。ISO/IEC9126[8]では品質特性を6つの特性に分類し、さらに特性に対して21の品質副特性を定義しています（図２参照）。ISO/IEC9126は、医療関係におけるQOLでのSF-36のように、品質を測定する包括的な尺度といえるでしょう。

ISO/IEC 9126に関する詳細な説明は他の文献[10]を参考にしてください。これから、ISO/IEC 9126の品質特性を参考に、組込みソフトウェアに関する品質とCATS社の取組みについて述べます。

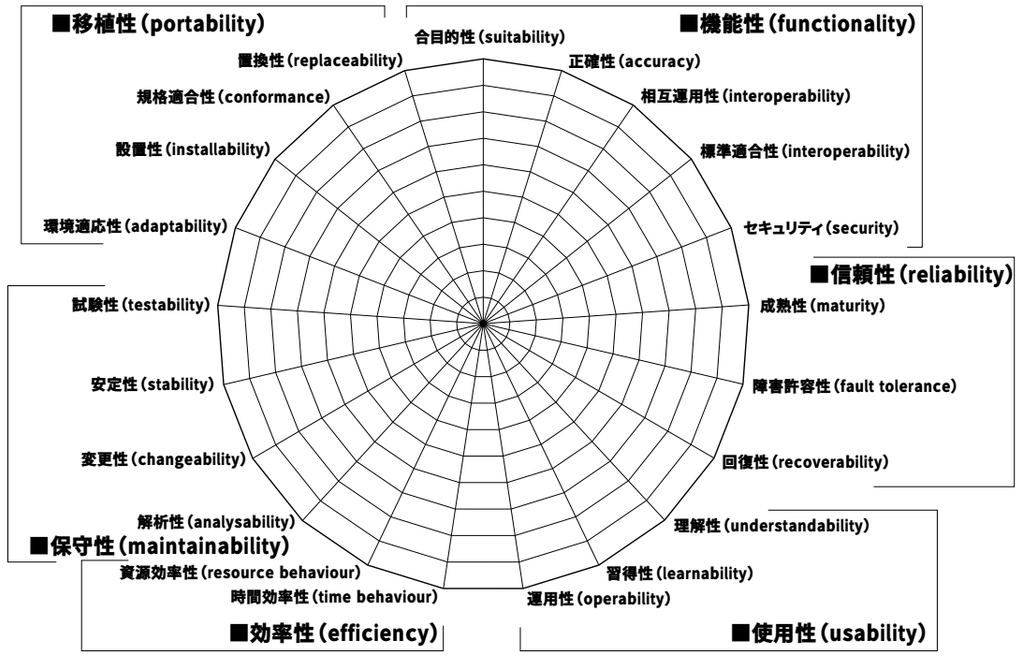

図２　ISO/IEC9126評価指標　9

「機能性」品質特性は、目的から求められる必要な機能の実装度合いです。機能性品質特性には５つの副特性があります。「合目的性」、「正確性」、「相互運用性」、「標準適合性」、「セキュリティ」です。ユーザ要求と設計仕様との合致性度合いを「合目的性」で示し、設計仕様が正確にプログラムコードとして実装されているかを「正確性」で示します。従来「合目的性」を検証するには設計レビューか、プログラムコードまで工程を進めた（下流に下りた）後に、システムテストを行うことで実現していました。設計レビューは人手による目視検証のため、品質がエンジニア個人のスキルに大きく依存します。プログラムコード実装工程でユーザ要求の検証を行うため、フィードバック工数が増大します。このため、QCD（Quality, Cost, Delivery）が低下します。そこで現在はユーザ要求をモデル化し、このモデルを実行することで「合目的性」の検証を上流工程で行うようにして、QCDを改善します。このようなモデルを中心に開発を進める方式を「モデルベース開発」と呼び、従来のコードを中心に開発を進める方式を「コードベース開発」と呼びます。V字モデルにおける妥当性確認（Validation）が「合目的性」、検証（Verification）が「正確性」ととらえる事が出来ます（図３参照）。

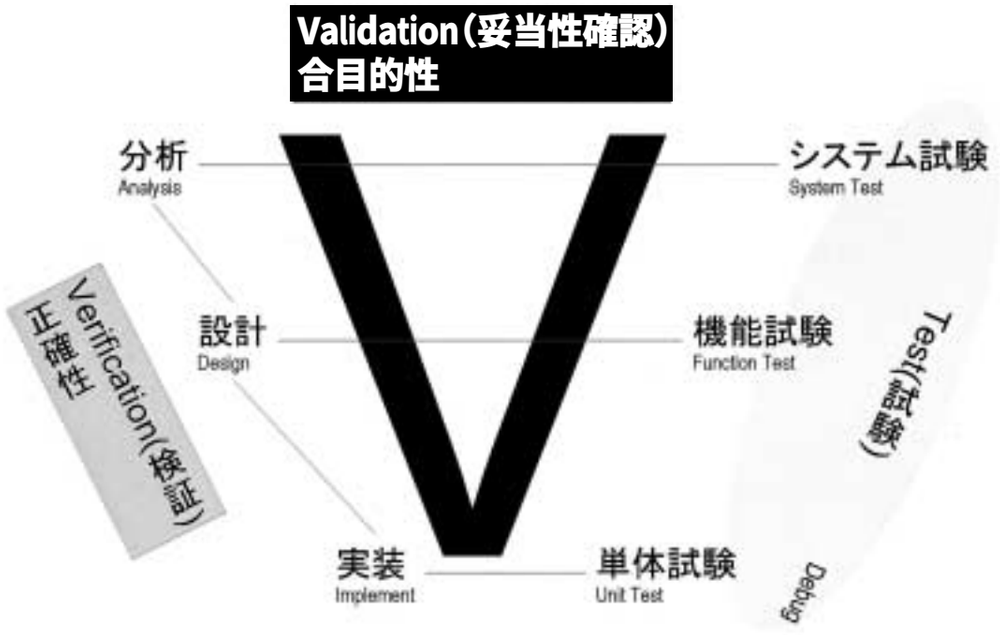

図３　V字モデルと機能性品質特性

ユーザ要求が比較的設計者寄りの場合、ユーザ要求を設計モデルで表現し、「合目的性」の検証を設計工程で行います。設計モデルから実装モデルへのマッピングは、現状ではほぼ100%近くを自動コード生成機能により実現できます。これにより「正確性」品質副特性を飛躍的に向上できます。ZIPC[11]を開発に適用したシステムでは全コードの97.5%を自動的生成コードが占めた実績があります[12]。ユーザ要求が企画や利用者側寄りの場合だと「分析モデル」が必要となります。分析モデルをUML（Unified Modeling Language）で表記するには、Konesa-RealTimeやIBM社のRoseがあります。UMLが苦手とするHMI（Human Machine Interface）の分析モデルにはDrawrial、Altia社のAltia Designが利用できます。分析モデルを実行することで「合目的性」の検証を分析工程で行います。「正確性」品質副特性を向上させる自動コード生成については、DrawrialではZIPCの状態遷移表に変換し、ZIPCの自動コード生成機能に接続します。Altia Designの場合はDeepScreenにより自動コード生成が可能です。Konesa-RealTimeはRTOSラッパクラスを用いて、アクティブオブジェクトをタスクにマッピングし、コードを自動生成します。ZIPCはRoseのステートチャートを読み込み状態遷移表に自動変換し、ZIPCのコード生成機能を用いてCコードを生成します。XModelink[13]はRoseのクラスをインポートし、SystemCの構造図エディタにマッピングしSystemCコードを自動生成します。モデルベース開発環境では、「正確性」を設計仕様とプログ

7 http://www.sound-zaidan.com/01rs2.4-2.5c.PDF
8 対応JIS規格はJIS X 0129
9 図の目盛は筆者が適当に入れたものです。
10 http://www.cam.hi-ho.ne.jp/adamosute/kyotu/iso9126.htm
11 第17回神奈川工業技術開発大賞受賞製品
12 ZIPC WATCHERS Vol3.「ZIPCを適用した防災システムの開発」
13 第11回LSIオブ・ザ・イヤー設計環境／開発ツール部門グランプリ受賞製品

ラム間で測定する必要はなくなりつつあります。代わりに分析モデルと設計モデル間についての「正確性」品質を測る指標がこれから必要になります。

「相互運用性」は他のシステムとの間で協調動作できるかどうかといった尺度、「標準適合性」はMPEGやVoIP等の標準規格にどれだけ対応できているかといった尺度です。「セキュリティ」はおサイフケータイ（iモードFeliCa）のように携帯電話がお金やクレジットカード、会員証などに早変わりする時代には重要な品質特性となります。個人情報保護法が2005年４月から施行されます。違反すると最高６ヶ月以下の懲役または30万円以下の罰金です。個人情報漏洩賠償の相場は、メールアドレスだけだと4000円、これに氏名、住所、本籍まで漏洩すると30万円を上回るそうです。2003年の個人情報漏洩被害者総数は約155万人で、このモデルで試算すると、280億円の損害賠償リスクになるそうです[14]。

「信頼性」品質特性は、実装している機能があらゆる条件の下で、機能要件を満たして正常動作し続けることができる度合いです。機能性とはユーザ要求を満足する機能になっているかということです。ユーザもしくは企画が要求する電子レンジが何なのかをモデル化します[15]（図４参照）。このモデルを実行して「合目的性」を検証するわけです。この電子レンジの信頼性はどうでしょう？あらゆる条件下で「合目的性」を持った電子レンジが機能し続けるのでしょうか？信頼性を得るための技法として状態遷移表があります。図４のステートチャートで記述されたモデルを状態遷移表で記述したモデルが表２です。「調理中」に「ボタン押下」した場合の挙動がステートチャートにはありません。つまり、あらゆる条件下で動作することが保障されていない信頼性の低い電子レンジということになります。CATS社は1992年に状態遷移表を拡張階層化した設計手法を公開しています。ZIPCはRose以外にもTMW社のMATLAB/Stateflow、I-Logix社のStatemateMAGNUMのステートチャートを読み込み、状態遷移表を自動生成することで「信頼性」品質特性を向上させる機能を持っています。ステートチャートモデルから状態遷移表モデルにすることで、モデルを網羅的に「状態と事象の組合せ」として見ることができます。「信頼性」はモデル化という作業の過程で作りこまれるものです。分析・設計段階において、いかにあらゆる条件の下で、機能要件を満たして正常動作し続けることができるように、分析・設計しておくかが信頼性の鍵なのです。

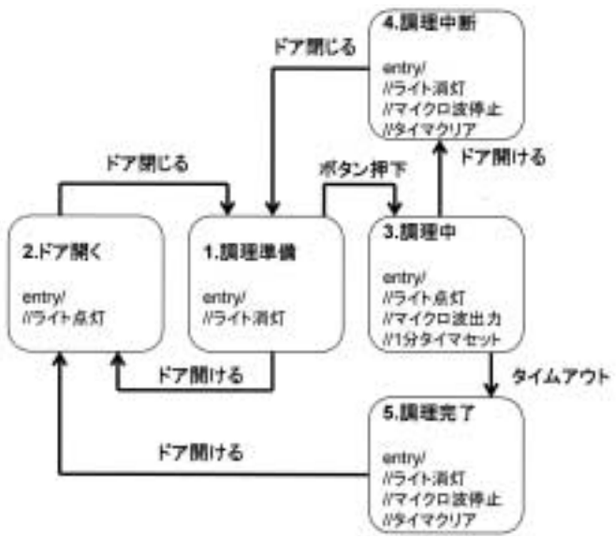

図４　電子レンジモデル　ステートチャート

事象＼状態	調理準備	調理中	調理完了	調理中断	ドア開く
ボタン押下	=>調理中	✸	✸	/	/
ドア開ける	=>ドア開く	=>調理中断	=>ドア開く	×	×
ドア閉じる	×	×	×	=>調理準備	=>調理準備
タイムアウト	/	=>調理完了	×	/	×

表２　電子レンジモデル　状態遷移表

このようにして「信頼性」を作りこんだ製品を外から測る指標として、「成熟性」、「障害許容性」、「回復性」があります。
「成熟性」とは潜在バグが少ない枯れた状態になっているかを示す尺度です。「成熟性」を評価する指標として平均故障間隔（MTBF：Mean Time Between Failure）があります。

平均故障間隔 =
ソフトウェア(システム)の総稼動時間 / 故障回数

「障害許容性」は、組込みシステムの場合、想定外の事象やデータが起こっても、ハングアップしない設計になっているかという指標です。表２の電子レンジで調理中にボタン押下されて何も反応しない故障と、押下された際に電子レンジが暴走して何のボタンを押下してもウンともスンとも言わなくなり、電源リセットする以外に道がないのでは障害許容性は低いことになります。
「回復性」とは障害時の回復・復旧の能力です。これを測る指標として稼働率があります。

稼働率 = MTBF /（MTBF + MTTR）
平均復旧時間（MTTR: Mean Time To Repair）= 修理時間の和 / 故障回数

「使用性」品質特性は、使い勝手の良さを評価するものです。「理解性」は機能が理解しやすいように提供できているかの指標です。エレベータドアを開閉するボタン「開」「閉」は理解しやすい一方で、誤操作を多発させます。「習得性」は機能を使いこなすまでの時間です。最近の携帯電話では入力予測変換機能が搭載され、簡単にメイルを作成することができ、「習得性」が向上しています。「運用性」はソフトウェアを陳腐化せずに運用できるかの指標です。カーナビでは、無線で最新の地図に更新できるなど「運用性」が向上しています。ZIPCやDrawrialで、ビジュアルプロトタイピングシミュレーションを何度も繰り返し行うことで「使用性」を向上させることができます。以前はこのシミュレーションを行うために、プログラミングを必要としていましたが、現在はモデルベース開発によりモデルでシミュレーションが可能となり、QCDを向上できます。

「効率性」は組込みソフトウェアの特徴的な品質特性と言えます。「効率性」を測る指標に「時間効率性」と「資源効率性」があります。「資源効率性」の向上は組込み製品には重要な要素です。メモリサイズの削減は、実装面積のサイズを小さくすることができます。軽薄短小を求める製品ではメモリサイズは重要です。また、ROM/RAM容量を減らしたりするこ

14 朝日新聞2004年9月1日朝刊1面
15 Executable UML:Mellor, Balcer:Addison Wesley,2002, ISBN 0-201-74804-5

とで製品コストを削減できます。消費者は品質と性能と機能が同じなら安いほうを買うのは万国共通です。ソフトウェア機能を実行する際に、時間的にどのくらいかかるのかを示すのが「時間効率性」です。「時間効率性」には時間制約を守れないと機能として成り立たないハードリアルタイムと、多少遅れがあっても機能自身には問題がないソフトリアルタイムがあります。ハードリアルタイムはむやみやたらと速く実行すれば良いというものではなく、一定時間内の実行を保証するものです。一定時間内の制御（ハードリアルタイム）だけをしていればよかった機器に、信頼性向上のため、監視機能（ソフトリアルタイム）を追加することで、マルチタスク構成とするケースが多くなっています。マルチタスク構成になると複数のタスクが動作するので、一定の時間保証をするためにリアルタイムスケジュール理論などを用いて「時間効率性」を保証します。さらに組込みシステムが複数のブロックから構成される組込みシステムでは、各ブロックの時間効率性を上げるとともに、ブロックを接続するバスの時間効率性を上げる工夫も必要になります。こうなると通信のところでお話したQoSを考慮する必要があります。携帯電話でメイルをしながら音楽を聴くなどの○○しながらの場合、それぞれの○○の時間効率は良くても、同時に使用した途端、両方の機能が使い物にならない時間効率では困ります。さらにモバイル環境で使用されるとなると消費電力も重要で、タフなアルゴリズムをハイクロックなCPUでぶん回すとあっという間に電池切れになってしまいます。「時間効率性」には機能実現時間指標を達成していても、その機能を使用できる時間がどのくらいなのかといった指標も組込みシステムの場合は必要です。XModelinkはUMLのクラス図モデルとSystemCの構造図モデルをクロスでモデリングすることで、「機能の合目的性」と「効率性」をアーキテクチャ設計によって、最適解を導出します。

CASEツールで「効率性」を向上させて、さらに毎年目標値を決めてさらにCASEツールを進化させることは、「効率性」に関して、熟練した職人がいなくとも、ツールを使えるエンジニアであれば、効率の良いコードを自動生成できます。EDA分野における自動配線と同じように、CASEツールも今後進化を続けることでしょう。ある実システムへのZIPC適用時のコード効率は、手書きの0.97倍のコードサイズであった実績があります（表３参照）。これからの組込みソフトウェア技術者は「効率性」をツールに任せて、他の品質特性に注力すべきです。

表３　コード効率 [16]

■コード効率		
システム名	社名	効果
アナログ電話機	NEC	ハンドアセブラと同等
WCDMA	NEC	ハンドCの1.17倍
火報システム	松下電工	0.97～1.28%倍
複写機	コニカ	コード効率ハンドコードと同等

従来の組込みソフトウェアは、効率だけを考えて設計、実装されることが多かったように思えます。「資源効率性」「時間効率性」が最優先され、そのために「相互運用性」、「セキュリティ」、「障害許容性」は犠牲にされました。しかし、組込み機器がインターネットに接続され、ソフトウェア開発規模が増大するにつれて、「効率性」最優先から「信頼性」、「使用性」、「保守性」にも目を向けざるを得ない状況です。組込みソフトウェアのアプリケーションは生命に関わるものから娯楽まで様々な種類があり、そのため求める品質は異なります。図5に示す品質特性では●印の品質傾向は「効率性」「保守性」「移植性」だけ重視型、■印の品質傾向は「機能性」「信頼性」だけ重視型、▲印の品質傾向は「信頼性」だけ犠牲型になっています。表１のQoSのように、アプリケーションドメイン品質レベルをISO/IEC 9126を指標に作成することが必要です。

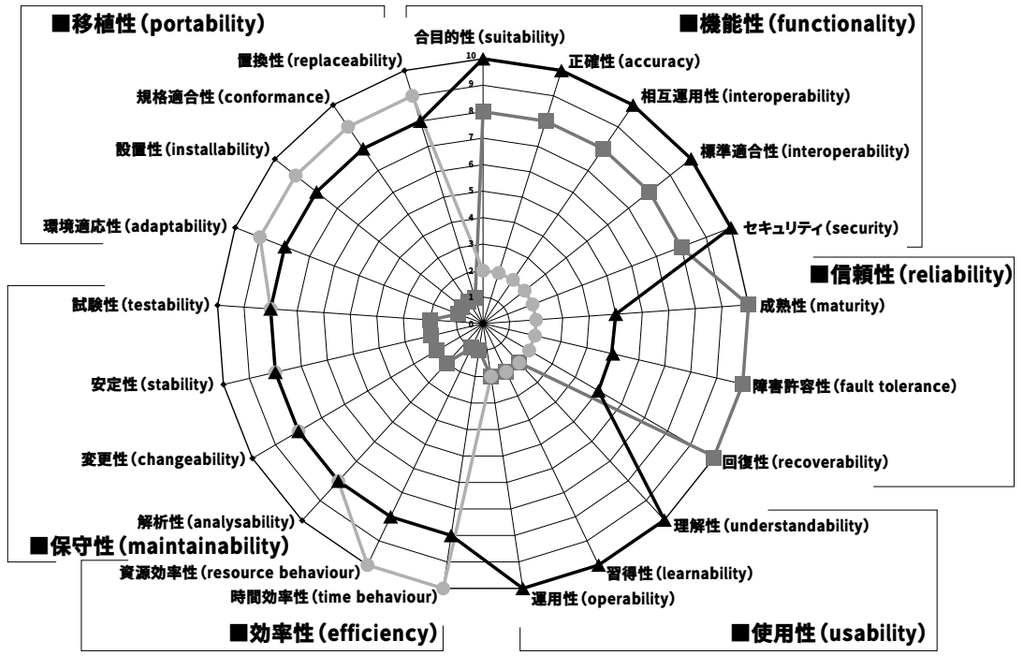

図５　組込みソフトウェア品質特性　指標

「保守性」には「解析性」、「変更性」、「安定性」、「試験性」の副特性があります。保守の場合は変更が簡単にできることが重要です。このためにはコードだけのバージョン管理ではなく、分析モデル、設計モデル、さらに各モデルに対するテストスクリプトの構成管理が必要です。特に試験の問題は早急に手を打つべき状況にあります。コスト、納期のシワ寄せが試験に起こっています。自動コード生成は実システム適用まで到達していますが、試験の自動化はまだこれからです。Perfect Pass[17]は、状態遷移設計モデルから自動的にテストスクリプトを生成します。これまで、主観的に「しっかり試験する」といわれていた世界を、「しっかり試験する」とはどういうことかを客観的な数値で示してくれるツールです。例えば、表4の状態モデルを完全に試験するには幾通りのパスをテストすることなのかといった客観的数字の裏づけが必要です。Perfect Passは表4の状態モデルからおよそ11時間かけて102億5474万2021ケースのテストスクリプトを自動的に生成します。100億を超えるテストケースは100人体制でも1人1億ケース、1日100ケースを実施したとして、100万日、つまりおよそ2700年もテスト完了に時間を要することになります。これではQCDのコスト、納期上現実的ではありません。しかし、この数字を管理者が把握できていないと、「がんばれ」の掛け声で結局のところ誰かがどこかでテストケー

[16] ZIPC WATCHERS Vol.1～Vol.7に詳細記事掲載

[17] ＩＰＡ：情報処理振興事業協会「重点領域情報技術開発事業」試験関連標準化「組み込みシステム用ソフトウェア評価項目抽出ツールの開発」

スをはしょることになります。この責任を開発者が取ることもなく、されど管理者も取ることもなければ、何の進歩ももたらしません。まずはどれだけ膨大なテストケースがあるのかを開発者と管理者が数値として共有し、その後、アプリケーション特性ごとにテストケースを削減するアルゴリズムを開発し、品質レベルに応じたテストケース自動生成を行うことが必要です。言うは易し、行なうは難しですが、すでに幾つかの企業では、ツールに品質パラメタを与えて、テストケースを削減する試みが行われているようです。

表４　状態遷移モデル

1 火災			わ			わ	
			災害	警告	お知らせ	ｻｰﾋﾞｽ	ﾃﾞﾓ
			入力	0	1	2	3	4
センタ	ｻｰﾋﾞｽわ	災害	0	／	=>0	=>0	=>0	／
		警告	1	=>1	／	=>1	=>1	／
		お知らせ	2	=>2	=>2	／	=>2	／
	ｻｰﾋﾞｽオフ		3	=>3	=>3	=>3	／	／
災害ﾚﾍﾞﾙ			4	警報ﾚﾍﾞﾙ設定（災害，火災） 火災発生通知	警報ﾚﾍﾞﾙ設定（災害，火災） 火災発生通知	警報ﾚﾍﾞﾙ設定（災害，火災） 火災発生通知	／	／
警告ﾚﾍﾞﾙ			5	／	警報ﾚﾍﾞﾙ設定（警告，火災） 火災発生通知	警報ﾚﾍﾞﾙ設定（警告，火災） 火災発生通知	／	／
お知らせﾚﾍﾞﾙ			6	／	／	警報ﾚﾍﾞﾙ設定（お知らせ，火災） 火災発生通知	／	／
センタ	停止		7	=>4	=>4	=>4	／	／
	復帰		8	／	／	／	／	=>わ（閉）

「移植性」には「環境適応性」、「設置性」、「規格適合性」、「置換性」の副特性があります。組込みソフトウェアは、アセンブラからC言語に移行することで、CPU依存から脱却しました。TCP/IPなどの通信適合性のためにミドルウェアやIPを導入し、RTOSも内製からITRONといった市販RTOSに移行しつつあります。今後はモデルレベルでの置換性などが検討され、モデルベース開発が加速すると思われます。通信ではOSI通信レイヤ構造のプレゼンテーション層をASN.1により抽象度の高い表記で仕様が記述され、実装モデルと分離することで「移植性」を高めています。ASN.1からC言語を自動生成する製品がASN.1ツールです。「移植性」についてはエンタープライズ系ではCORBA（Common Object Request Broker Architecture）やRMI（Remote Method Invocation）などのリモートマシン上のオブジェクトのメソッドを呼び出すフレームワークであり、コンポーネントレベルでの再利用（移植性）が進んでいます。組込みソフトウェアの場合、フレームワークそのもので「効率性」を追求しており、なかなか標準的なフレームワークが用意されて、そこで動作するコンポーネントが再利用される状況にはないようです。しかし、近頃は、自動車業界でのソウトウェアコンポーネントフレームワークがAUTOSARという団体で検討がはじまったようです[18]。

品質特性を上げることでコストや納期を削減できますが、品質特性を上げることでコスト、納期が犠牲になることがあります。モデルベース開発はQCD全てに良く効きます。しかし、「使用性」を向上させるための試行錯誤は、どれだけ行ってもきりはありません。コストと納期を考えてその中で最適な使用性を選定します。コストや納期は結果が分かりやすい、見えやすいものですが、品質はそれなりの尺度を持って計測していないと、結果が分かりにくく、見えにくいものです。品質向上をお題目のように唱えていても、科学的で客観的な数値指標を持たないと、品質を犠牲にしてコスト、納期を死守している状況を見過すことになります。

「モデルベース開発環境への移行」は開発部門のテーマととられがちですが、実は品質管理部門にとっても重要なテーマです。品質管理部門から開発部門への要求を、開発者はやたらと書類を書かされる手間のかかるもので、直接的に開発効率を上げてくれないことが多いと思っています。ところがモデルベース開発への移行には、開発者はモデリング技法のスキルが身につくので、取組むモチベーションが上がります。モデルベース開発ではCASEツールによる自動コード生成などの機能によりQCD向上の恩恵が受けられます。CASEツールで品質に関する計測を自動的に行い、ツールの機能向上することで組込みソフトウェアの品質特性向上することができます。製造ラインの品質管理は、オートメーション化とQC活動によって“MADE IN JAPAN”を世界に知らしめました。2004年アテネ・オリンピックで、日本は金メダル16個、銀メダル９個、銅メダル12個と、ロサンゼルスオリンピックを超える史上最多37個のメダルを獲得しました。福田日本代表選手団総監督は、2001年10月に完成したJISS（国立スポーツ科学センター）の活用が日本復活の施策の１つであったと述べております。JISSによって、集中してトレーニングに打ち込める環境が整備され、多角的分析によって導き出されたデータを現場に迅速にフィードバックできる体制が構築できたそうです[19]。もう「根性」だけでは、世界のトップアスリート同士の競争でメダルを獲ることはできません。科学的なデータを競技者と共有し、対策を施行することが必要です。さて次は、組込みソフトウェアの出番です。

以上

[18] http://www.autosar.org/find02.php
[19] http://www.joc.or.jp/stories/specialinterview/20040401fukuda02.html