

組込みソフトウェア上流開発環境を日本で

渡辺 政彦／キャッツ(株)



バブルがはじけて以降、我が国のメーカはソフト開発ツールや開発支援環境の研究開発をやめて、ひたすら外国任せになってしまった。補正予算の膨大な開発費も戦略商品開発にまわされているがいかなものか。開発ツールや環境の研究開発は皆が協力しやすく、失敗が少ない。こういう分野の研究は工作機械の開発と同じで地味だがうまくいったときの影響力が大きい。こんな分野をもっと一度きちんとしようではないか。進め方としてはGNU、LINUXなどというオープンソースなども参考になる。

21世紀の組込み業界はホットに

自動車では1993年型トヨタ製「ソアラ」には32個のマイコンが搭載されている。2000年には1990年の5倍のマイコンが搭載されるという。またITS(高度道路交通システム)では60兆円もの市場が誕生し、その大部分がソフトウェアになるという。携帯電話では1993年頃に25Kゲート数、数百Kバイトのメモリ規模が、2001年では数十万～数百万ゲート数、数十メガバイトの規模になると予測されている。デジタル家電では2000年には1997年の4倍のコード行数、5.5～16倍の開発工数が必要になるという。

組込みソフトウェアはなぜ難しいのか

組込みシステムの特徴の1つに、ハードウェアが可変であることが挙げられる。ウィンテルのような共通プラットフォームはない。組込みソフトウェア



アのバージョンアップはシステムモデルアップであり、ハードウェアの変更もあり得る。特定のアルゴリズム(たとえば圧縮・伸長)をRISC(Reduced Instruction Set Computer)上にC言語で実装するか、CISC(Complex Instruction Set Computer)にアセンブラ言語で実装するか、DSP(Digital Signal Processor)にするか、ユーザロジックでゲートにするかといった選択肢が用意され、どれを選択するかは生産量や単価の影響を受ける。

さまざまなリソース面での制約を受ける。リソースとは物理的な実装面積であったり、ROM/RAMのサイズであったり、処理速度や消費電力の場合もある。待機時間での電力消費を下げるためにマイコンのクロック数を制御することもある。

ソフトウェアのリビジョンアップ版をCD-ROMやネットワークで配布といったことがしづらいものである。組込みソフトウェアは機器に組み込まれて出荷されるため、出荷後の改修には多大なコストを強いられる。ユーザにキーボード、ディスプレイと基本OSの知識を前提に求めることはできない。ソフトウェアの動作不良はリセットして再度立ち上げるといったリセット文化を押しつけることはできない。製品のリコールとなるのである。

道具開発は重要だ

日本の土木建築技術は世界に誇る

べきものであろう。トンネルを掘るといったことを考えてみても、新しい道具の開発が生産性向上以上に過去不可能であったことを可能にしている。青函トンネル、ドーバ海峡トンネルにも日本の土木技術が採用されたと聞く。実際に掘ってみなければどのような地質が分からず、多くの人命を奪ってきた。これが弾性波探査装置の開発により、地質構造を実際に掘らなくても分かるようになった。湧水軟弱地盤を硬化剤注入といった手法・装置開発により軟弱地盤でのトンネルを可能とした。大断面掘削機の開発により硬い断面に直径8メートルもの掘削が可能となった。

日本生まれのワープロソフトが苦戦している。理由の1つに開発環境が米国製であることが挙げられる。マイクロソフトのOS上で動作するビジネス統合アプリケーション(Microsoft Office)のようなものを開発する場合、使用する開発環境もマイクロソフトが提供する統合開発環境(Microsoft Visual Studio)を使用することになる。最新Officeにある機能の一部が開発環境である最新Visual Studioでは作れないことがある。Officeのライバルはこの段階で遅れをとることになる。

上流を狙おう

組込みソフトウェアの下流開発環境は整備されつつある。不具合の工程別分類をみるとソフトウェア設計



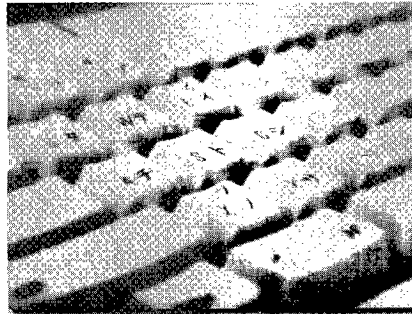
(38%), システム設計 (30%), コーディング (21%), 仕様検討 (10%), その他 (1%) という結果がある。実に8割近くの不具合が上流工程に起因している。

組込みソフトウェアは木と似ている。根 (実装) の部分はさまざまなプロセッサの選択肢がある。幹 (プログラミング) の部分になるとC言語により、さまざまなプロセッサ間の移植性の問題を解決する。枝葉 (分析・設計) の部分になってくると、さまざまなRTOS (Real Time Operating System), 独自の分析・設計手法となる。枝葉 (上流) の部分はそれぞれの組込みソフトウェアにより姿形は変わってくる。1つの携帯電話に組み入れられるソフトウェアにおいても、ウェブブラウザやメーラ部分とOSI (Open Systems Interconnection) 通信プロトコルのLayer1~3を司る部分では同じC言語で実装されたにせよ、分析・設計方法は異なってくる。ウェブブラウザやメーラ部分にはUML (Unified Modeling Language), OSI通信プロトコルにはSDL (Specification and Description Language) を用いて表記するであろう。物理層はユーザロジックによりゲート化の検討となると、コ・デザインを意識することになる。

ハードウェア設計にHDL (Hardware Description Language) が多用されている。ハードウェア設計が回路図で行うことが主流であった時代に、米国ではHDLによる設計の研究・開発を行っていた。日本はHDLによる論理合成結果のゲート数が現実的ではないとの見方が大勢を占めた。これが今日のハードウェア論理設計ツールに国産のものが極端に少ないといった状況を引き起こしている。戦略的なシステムLSIを作ろうとして、それを作るための環境が意図的に操作される危険性を孕んでいることになる。EDA (Electronic Design Automation) ツールはLSI設計の他にソフトウェア側の検証もできるような試みをコ・シミュレーションで行っている。コ・シミュレーションはソフトウェア設計者にとっては粒度の細かいCのインストラクションレベルが、ハードウェア設計者にとっ

てはCPUのモデルとして最大の粒度として認識され、そのギャップが明らかになってきている。海外のEDAツールベンダによるHDLをより抽象度の高い表現方式でモデル化する試みを始めている。IPをRTLレベルでの流通からさらに抽象度の高い表記で部品化を進めるものである。この分野において、国産では東芝の仕様記述言語SpecC言語、ツールとしてVisual Specが発表されている。

情報通信時代の主役となるソフトウェア分析・設計においても、ハードウェア設計時代の歴史を繰り返すことは避けなくてはならない。OMG (Object Management Group) によるUMLやITU (International Telecommunication Union)



によるSDLが欧米にて提言され、その支援ツールが続々と欧米で誕生している。この分野において、国産ではキャッツのEHSTM (拡張階層化状態遷移表) 設計手法があり、ツールとしてはZIPCが発表されている。EHSTM設計手法は設計時のモレ・ヌケを防止するために、欧米で使用されている状態遷移図ではなく、事象と状態のすべての組合せを記述できる状態遷移表をベースにしている。HCP (Hierarchical and Compact Description Chart), PAD (Problem Analysis Diagram) といったフローチャートだけが日本生まれの技術として取り返しのつかないことが起きる。

RTOSでは日本製のμITRONが日本で組込みのデファクトになっている。このように組込み分野ではちらほらと日本製の開発環境、ツールが出現し

ている。この火を絶やしてはならない。開発ツール、開発プラットフォーム、ターゲットプラットフォームが海外製で埋められてしまえば、作れば作るほど海外にロイヤリティが流れ、作ったものは価格競争の波を被り、利益の薄い構造となってしまう。これではさらに生産性・品質を向上させるための研究・開発費は捻出できないことになり、悪循環が起こってくる。

モノ作りに活気を与えよう

世界に名をとどろかす日本の製造業。自動車、携帯電話、デジタル家電、FA。この分野における組込みソフトウェアの比重が高まってきている。組込みソフトウェアはPC、WS、汎用機上に構築されるアプリケーションソフトウェアとは異なった性質を持っている。日本の組込みシステムから、組込みソフトウェア上流開発環境を発信すべきである。組込みソフトウェアの上流開発環境を構築するということで以下のことを進めることができる。

- (1) 国内製造業の生産性・品質向上
- (2) 組込みソフトウェアの分類
- (3) 産学連携体制の確立
- (4) ベンチャー企業育成

国はベンチャー企業に国内製造業の生産性・品質を向上させる開発環境・ツールを作成するための費用を援助する。国は最初のツール開発イニシャルコストに予算を使うだけである。ベンチャー企業はその後ツールの売上や、ツールに関するサービスで自らを成長させる。これは国のベンチャー育成政策にも適ったものであり、新しい雇用を生み出す。ベンチャー企業は国内製造業の現状問題点とその分類 (パターンとプロセス) を進めなくてはならない。これらなくして生産性・品質を向上させる開発環境・ツールは作れない。競争の激しい組込み業界では企業機密の問題になるが、ベンチャー企業が各製造業企業と守秘義務を締結することで回避できる。ベンチャー企業の役割は製造業の現在の開発・保守における現

場レベルの問題点を抽出する。それを自社だけでなく、大学、国家機関の研究所にテーマとして提出する。大学・研究機関はIT革命時代の製造業における組込みソフトウェアをテーマとして貢献する。またベンチャー企業は大学や研究機関の研究成果をツールとして製造業に提供し、理論と実践にあるギャップを明確にし、それを大学、研究機関にフィードバックする。製造業の現場では論文を読み、理解し、実践に適用する時間をかけることが難しい。ベンチャー企業からツールとコンサルティングといったサービスを提供されることで適用する機会が増える。開発されたツールは世界的なブランド力を持つ日本の製造業が適用したということでグローバルスタンダードに結びつきやすくなる。日本の組込み産業界の生産性・品質が向

上し、産学連携およびベンチャー企業の育成につながる仕掛けとなる。開発したツールそのものを販売する方法もあるが、GNUやLinuxといった戦略で一気にデファクトスタンダードにすることも考えられる。ベンチャー企業はツールサポートサービスにより収益を上げることになる。

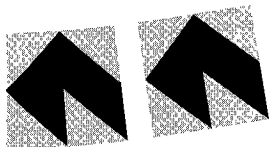
歴史は繰り返す

20世紀の産業革命の偉業は、製造業における肉体労働の生産性を50倍に引き上げたことである。21世紀に期待されるのは知識労働者、ホワイトカラーの生産性を向上させることとなる。ソフトウェアは知的労働が集約されたもの。ITが21世紀を変えていくといわれている。そのITの一翼を担うソフトウェア生産労働は、エンジニア個人のスキルに依存する職人の集団

に大きく依存し、生産性革命はまだ起こってはいない。フルスクリーンエディタが登場し、ベンチャーという業種がなくなったぐらいのものか。

1849年、一攫千金と東海岸、ペルーから大勢の人達が西部にやってきた。彼らを49er (フォーティ・ナイナーズ)と呼び、サンフランシスコのフットボールチームの名前にもなっている。東から西への道々でシャベルを売って大金持ちになった人がいる。金を掘る際の作業用パンツを作って大儲けしたのが後のリーバイスジーンズ。さて、時は2000年。情報通信産業はこれからがゴールドラッシュ。情報通信で一旗上げようとしている輩に「道具」「作業環境」で儲けることは、歴史は繰り返すことが真実だとすれば、確実なことだと思いがいがなものか。

(2000.1.8)



ツールと方法論を受け入れる 組織文化の育成も必要

羽生田 栄一／オージス総研

ここでは、オブジェクト指向の観点から渡辺さんの主張を補足してみたいと思います。

私の職場では同僚も含め、顧客に対してオブジェクト指向でのシステム開発のコンサルおよびSIを提供しているのですが、ここ1年ほどの家電や自動車を含む組込み業界におけるオブジェクト技術への取り組みへの熱心さは驚くべきものがあります。つい数年前までは、組込み業界でのオブジェクト技術に対するイメージは、遅い・重い・分わからないの3拍子揃った時期尚早という意見だったといえるでしょう。それがこの1~2年でがらっと変わりました。C++での開発に切り替えるので手伝ってほしいというのが多いですが、いくつかより深刻な相談もあります。いままでCやアセンブラで職人芸的に組込みソフトを作っていた部署から、ぜひオブジェクト指向技術を導入して、シリーズ化され次々に

短サイクルで投入される製品向けのソフトウェアをより体系的に再利用できる部分はしつつ製造し、しかも技術が特定の個人に従属せずエンジニア間で共有できるような開発体制を作りたいので協力してほしい、というものです。つまり、現場サイドからの現状を打開するための止むに止まれぬ声というのが聞こえてくるのです。

実は渡辺さんへの応答記事を依頼されたとき私はたまたまネットニュースで「携帯電話の爆発的な普及の影で携帯電話のトラブルが相次いでいる。iモード、cdmaOneを問わず、各地で障害が起き、機器メーカーの初歩的ミスによる“不良品”まで市中に出回った」という記事に出くわして(よく読むと端末側組込みソフトだけでなくサーバ側の問題も多いのですが)日本の製造業らしくないなと思っていたところでした。つまり、インターネットや通

信を中心にビジネスの圧力が組込み製品のマーケット特性をも急速に変貌させつつあり、いままでの属人的な設計製造プロセスでは対応できない時点に差しかかっているということです。しかし、未だに100万、1000万という個数のオーダーで掃ける組込み製品の既得権益からメーカーの購買部門が力を持ち、製品のトータルな開発プロセスでのコストと生産性を考える前に、その時点で局所的に最安値の組込みチップを購入が決めてしまい、ファミリー製品であるにもかかわらずソフトは全部作り直し、という冗談のような話しも体質として解消しているとはいえないようです。こうした事実から感じるのは、オブジェクト指向開発用のツールを導入すれば問題は解決するのか、ということです。ツール特に上流レベルで定義された設計モデルから自動・半自動のコード生成ツールを使ってバグを混入しない



一定水準のソフトを製造させる、というのは生産性・信頼性を上げる際に非常に効果的な方法です。しかし、安直にツール導入だけで安心してほしくないのです。ある抽象度で記述された設計モデルを定義できる人を用意するだけでなく、知識が1人に偏らないように、そして分散してチーム開発できるようにそのモデルを共有しコミュニケーションの際の共通言語として利用していけるような組織文化がないと、体系的な開発方法論として組織の中で定着していきません。そうした心構えが果たしてツールを導入しただけで整うのかという危惧です。

幸い、オブジェクト開発技法も成熟してきており、情報システムだけでなくリアルタイム制御系そして組込み系といわれるシステムにも適用可能なものが徐々に現れてきてきました。たとえば、ノキアでは4, 5年前から携帯電話端末の設計にOctopusという実時間オブジェクト技術を適用し成果を挙げていますし、リアルタイムUMLといったモデリング技法やユースケースマップといった個別技術も積みあがってきています。それらをペー

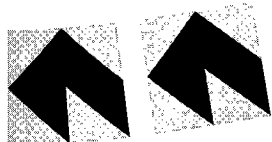
スにすれば、開発ツールと方法論はなんとかがんばって作り上げていくことができるでしょう。その点には不安を感じていません。あとは、そのような体系化を指向したエンジニアリングの発想をいかに製造業という組織文化に定着させていくかということになるでしょう。

そのためには、トップダウンの流れも重要ですが、現場サイドからノウハウを公共化していく努力も重要だと思われます。従来のQCサークルが製造プロセス改善だとすれば、個々のエンジニアの開発ノウハウをUMLモデルとともにパターン記述として共有するような文化が組織内に作り出せれば、優秀なエンジニアの育成とともに設計プロセス改善の手段としても有効に機能するのではないのでしょうか。特に、組込み分野では、標準的なオブジェクト設計技法でカバーできない、独自RTOSや独自開発環境上での、タスク分割、メモリ管理、パフォーマンスチューニング、フットプリントの極小化、Cやアセンブラでの実装、ハードと込みでのテスト技法等々さまざまな実践的なノウハウが必要ですから、それらがモデルやパターンとして

知識化され共有されれば、ツールの利用と相俟って生産性と信頼性の高い組込みソフトウェアの開発に貢献するでしょう。

今はまさに新しい設計・製造スタイルを統合的な開発環境と共に作っていく過渡期なのでしょう。渡辺さんがいわれるような「ツール・開発環境」を体系的な視点のもとで導入するチャンスではないでしょうか。組込みエンジニアの皆さんの置かれた、メモリ制約とスピード、開発期間のプレッシャーの中でのご苦労はよく分かります。だからこそ、もっと人間的な環境の中でより創造的な開発に取り組んでほしいと願う次第です。渡辺さんのツール開発提言もその理不尽さを解消したいという気持ちからのものだと感じました。私からは、ツールを適用する体系的な開発プロセスの定着とモデル・パターンを通したノウハウの共有の制度化という点を補足しました。

(2000.1.18)



コストと品質・信頼性のトレードオフを再考しよう

高田 広章／豊橋技術科学大学 情報工学系

日本が率先して組込みシステム分野の研究・開発を進めるべきというのは私自身の主張でもあり¹⁾、渡辺氏の主張にはほぼ全面的に賛成である。また、ツールの導入だけでは不十分で、開発プロセスの体系化が必要という羽生田氏の指摘もきわめて重要な指摘である。そこで、お2人の主張をサポートする意味も込めて、お2人も触れられている組込みシステムの品質・信頼性の問題について議論したい。

日本の工業製品の品質・信頼性が高いことは、確かであろう。元気のな

い日本の工業界を活気付けるには、このアドバンテージを維持していくことが重要と思われる。ここでは、日本の工業界がいかにしてこのアドバンテージを維持していくかという視点から議論する。

いうまでもなく、組込みシステムには一般に高い品質・信頼性が求められる。機器や機械を制御するという性質上、システムの誤動作が大きなことゆえに直結する。たとえば、家庭用の炊飯器であっても、システムの誤動作が火災につながる可能性がないとはいえない。また、渡辺氏も指摘して

いる通り、出荷後の改修に高いコストがかかるのも、品質・信頼性が求められる理由の1つである。

品質・信頼性を確保するために、組込みシステムのソフトウェア開発は、保守的な面が強い。システム全体が見えていないと品質・信頼性が確保できないという理由で、OSから始まり、システムを構成するすべてのソフトウェアを内製すべきと考えている技術者も多い(ツールは組込み機器上で動作するソフトウェアではないので、除外して考えてほしい)。やや時代遅れに思えるかもしれないが、それが品

質・信頼性の確保に重要な役割を果たしてきたのは確かであろう。一方で、組込みシステムの高機能化・複合化・ネットワーク化が進行している。機器がインターネット接続されたり、Javaのサポートが必要になってくると、すべてのソフトウェアを内製しては、タイムリーに製品を出すことが難しくなる。ドッグイヤー（さらにはマウスイヤー）という言葉があるように、きわめて速い技術の進歩の一方で、羽生田氏が例を挙げられている通り、システムの品質・信頼性が犠牲になっているのは、否定できない事実であろう。多くの組込みソフトウェア技術者が、この問題に悩んでいる。

米国での例を1つ紹介すると、ここ数年NASAが推進している一連の火星探査計画（Mars Pathfinderが有名）では、探査機をいかに安く作るかが重要なテーマとなっている。それまでのNASAの考え方であれば、それ専用に新たに開発したであろうものを、今回はできる限り市販品を使って開発している（たとえば、Mars PathfinderにはVMEバスが使われている）。その結果、コストは一桁下がったと聞いているが、その一方で、報道されている通りいくつかの探査機が失敗に終わっており、信頼性は確実に下がっているものと



思われる。もちろん、品質低下の原因はソフトウェアだけではないが、機器制御におけるソフトウェアの役割が急速に増している現状を考えると、ソフトウェアがシステム全体のコストと品質を大きく左右するようになるのは間違いない。

ソフトウェアのコスト削減（言い換えると、生産性向上）と品質・信頼性向上を両立させるための取組みとして、渡辺氏の指摘される優れた上流開発環境の開発・利用や、羽生田氏の指摘される開発プロセスの体系化は、必須のものである。

しかし、私には、日本の工業界がこれまで通り高い品質・信頼性の機器・機械を開発していくためには、それだけではまだ不十分であると思われる。すなわち、高い品質・信頼性をユーザにアピールし、それに対するコストに理解を求めることが不可欠である。そのためには、品質・信頼性を

分かりやすく提示する指標や規準が不可欠である。日本製の自動車は、海外においても信頼性が高いことで評価されていると聞く。これは、自動車産業にとっては幸運なことであるが、高い信頼性を何らかの指標化して消費者にアピールしていく努力をしなければ、その優位性は薄れていくものと思われる。

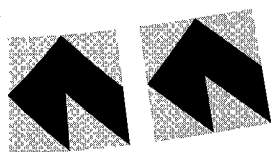
もう1つの観点として、日本では、信頼性を上げるというと部品の信頼性を上げることだけが重視され、システム全体の信頼性という視点が欠けているという指摘がある。やや乱暴な例ではあるが、上で述べた火星探査計画に関していえば、コストが一桁下がれば、成功率がたとえ半分になっても、コストパフォーマンスは上がっているとみることできる。個々の部品の信頼性確保も重要であるが、システム全体の信頼性を確保する技術についての研究・開発が望まれる。

以上、まとまりのない文章となったが、コストと品質・信頼性のトレードオフについて再考するきっかけとなれば幸いである。

参考文献

- 1) 高田広章他: インタラクティブエッセイ, 「組込みソフトウェア分野における産学間のギャップを埋めよう!」, 情報処理, Vol.40, No.5 (May 1999).

(2000.1.21)



再コメント

渡辺 政彦／キャッツ(株)

羽生田氏からは体系化された開発プロセスとモデル・パターンの共有の重要性を補足いただきました。高田先生からはコストと品質・信頼性のトレードオフを補足いただきました。両氏のコメントに対して再コメントをします。

開発プロセスとモデル・パターンは経営者をまきこんで

情報システムの構築には経営者サイドの参加が不可欠です。どのよう

な企業戦略に基づくかがシステム構築の基本概念になるからです。商品をどのように管理し、それらの情報をどこまでオープンさせ、その成果をどのように人事考課にするかなどは経営者抜きではできません。一方、モノ作りの方はどうでしょうか。経営者からはいつまでにいくら儲かるものを作れといった一方的な指示がくるだけではないでしょうか。こうなると技術現場としての自由な権限範囲は有効なツールと環境を整備す

るだけとなります。羽生田氏が指摘される安直なツール導入という形になってしまいます。特に上流の開発ツール導入となると設計の方法が根幹から変わることもあるのです。その際、それが成功するか否かは誰にも分からないものです。そこに現場を知らない管理職や経営者が登場し、その投資額に見合った数値を出せなければ、責任は導入したおまえにあるなどと言われてしまつては、日本全体に広がりつつある現状維持の無



気力が現場に蔓延してしまうのです。自社の技術者に現在の問題点と解決策を求めるのは正しい姿です。しかし、その結果の責任を経営者が技術は分からないから、技術者に責任転嫁させることはあってはならないことです。

コストと品質・トレードオフには経営者をまきこんで

東海村臨界事故、国産H2ロケット失敗、羽生田氏の指摘にあった携帯電話の欠陥商品の増加と日本の技術が揺らいでいます。高田先生が指摘するマウスイヤー並みの技術進歩と短納期による信頼性犠牲がおき、多くのソフトウェア技術者が悩んでいます。この悩みに耳を傾けてくれる会社役員はどの程度いるのでしょうか。高田先生が例題に出されたMars Pathfinderは大変分かりやすい例えです。一桁コストダウンするが、半数は失敗するだろうということを経営者がOKかNGかを判断するのです。コストはなるべく下げて、品質はなるべく上げよとオウムのように繰り返されても前には進まないのです。昔から優れた経営者は必ず自分の足で現場を回りました。ソフトウェアの現場を隈なく見ても、そこにはパソコンが並んでいて、最新のオフィスはパーティションでエンジニアを見回すこともできないかもしれません。せっかく、電子メールがあるのですから、現場のソフトウェアの問題を直接聞く手立てができています。

実現には国をまきこんで

きれいごとを並べ立てました。エンジニアは通常、従業員という枠組みの中に生きています。世の中これだけリストラが進む中で、経営者に結果責任をとれなどと諫言するのは難しいことでしょうし、それを笑って許す器の大きな経営者も少ないことでしょう。開発者が経営者といったベンチャー企業の場合、比較的動きはスムーズです。しかし最近困っ



たことが起きています。ベンチャーキャピタルにより、膨大な開発資金がベンチャー企業にバラまかれています。確かに高性能な製品を開発するには資金が必要です。しかし、誰のためのツール開発なのかを忘れがちになってしまうのです。ユーザのためのツールが投資家から資金を集めやすいためのツールになってしまうのです。国は経営者、技術者、企業の規模に関係なく、すべての納税者のために存在しなくてはならないと思います。そこで技術者がやりたくとも、経営者からすると不安なことに国が予算を使うべきだと思うのです。効果があるかないかも分からぬことに血税を使えるかといった論議はあるでしょう。しかし、半導体製造プロセスは官民一体となることができているのですから、組込みソフトの開発プロセスにも適用できるのではないのでしょうか。こうなると技術者と経営者の関係が、企業と国の関係にも見られるのでしょうか。日本国にはもしかして組込みソフトを知る役人はいないのかもしれませんが。

開発プロセスと品質を開発ツールで具現化したい

設計のための開発手法・ツールを1つにすることは、現実的ではありません。さまざまな製品形態によりそ

れらは変わります。それらを有効に使うにはプロセスの整備が重要であることは羽生田氏の指摘する通りでしょう。また、高田先生が言われるように品質についての指標化も開発プロセスにとって重要なことです。今回の製品は品質の指標値をいくつにするから、そのための開発プロセスにより、ある開発手法・ツールを使用していこうといった世界を21世紀には作り上げたいと思う次第です。抽象度の高い話しを具体的なツールもなく話しをするのはどうも哲学的、宗教的に思えて、科学的に思えないと私個人は考えてしまいます。人間の想像力は無から何かを生じさせるよりも、現在あるものに対してアレ・コレと発想がたくさん出るように思えるのです。そこで、品質・信頼性・コストや開発プロセスを具現化する上流開発ツールを作り、それをユーザのために進化させていくプロジェクトを実現したいと考えています。組込みソフトの現場からは動いた、動かないといった動かすことがメインの業務から動くのは当たり前で、何をどのような品質・信頼性で創作するかをメイン業務にするような改革をしていきたいと思っています。

(2000.1.23)



議論の続きは、次のURLをご覧ください。 <http://www.ipsj.or.jp/magazine/interessay.html>