

社会への提言

岐路に立つ組込みソフトウェア開発現場

～企業経営者や事業責任者への警鐘～

2012年6月28日

一般社団法人情報システム学会 企画委員会

提言検討チーム

1. 提言の主旨

現在の組込みソフトウェアの源は、制御用コンピュータのプログラム開発に端を発しています。限られた性能のマイクロコンピュータで厳しい稼働環境の中、高い信頼性と高いリアルタイム性を要求されました。そしてそこで培われた技術的なノウハウやアーキテクチャ、そしてソフトウェア開発技術者に埋め込まれた技術的価値観は、人を相手にする業務ソフトウェア（主にエンタープライズ・システム）とは少し別の世界として存続していました。その後コンピュータやネットワークの驚異的な性能向上とともに適用領域が飛躍的に広がり、社会的システムや企業の業務システムとネットワークを介して融合するほどになり、開発に携わる技術者も20万人とも30万人ともいわれるほどに成長しました。^[01]

本提言検討チームのメンバは業務ソフトウェアの世界で育ちました。しかし業務ソフトウェアの経験者として刮目すると、私達が過去に経験した失敗や苦勞と似たような問題で悪戦苦闘されているように見受けられます。そこでお互いに相互交流の必要があると感じたことが提言の発端です。

言うまでもなく組込みソフトウェアはこれからの産業社会のいたるところに利用され、「産業の知恵」として製品やサービスの競争力の源泉になりうるものと期待されています。これから蓋然性が高いと予想される大きな課題・問題に対して、早めに対策を講じると言う意味で、本提言は有益であると考えます。速やかに将来の展望を踏まえたうえで、組込みソフトウェア開発現場の改革・改善に着手すべきと考え提言いたします。

2. 組込みソフトウェアの「開発とメンテナンス」現場の声

組込みソフトウェア開発とメンテナンスの現場は、その製品の特徴によってソフトウェアの規模、要求性能、求められる品質がピンからキリまであります。例えば自動車に組み込まれているソフトウェアの規模は地方銀行の勘定系システムに近い規模になり、ソフトウェアにバグがあると人命に影響するほど重要な存在となっています。^[02]

その一方で家電製品によくある温度や照度に反応するセンサと連動して動作するような単純な機能のプログラムも、組込みソフトウェアの範疇に入ります。従って問題や課題を意識するレベルが全く異なり、関係者の問題認識が合わなくて、業界として問題を共有し協同で対策を講ずる気運がまだ薄いと感じられます。以下に提示する問題は事業的、技術的にトップレベルにある企業では克服されていても、大半の組込みソフトウェア関連企

業ではこれからの問題と考え、敢えて提言します。

2.1 プログラムのスパゲティ化が進み始めている

業務ソフトウェアの世界では当たり前ですが、プログラムは瑕疵や機能の追加・修正等部分的な改修を行うことが日常茶飯です。以前の制御用システムの組込みソフトウェアでは、いったん制作して十分な試験がなされた後は ROM に焼き付けられて固定化され、その製品寿命が尽きるまで安定的に劣化しないで稼働しておりました。しかし不揮発性のメモリや書き替え可能な CD が現れ、簡単にプログラムの更新が可能になってから、頻繁な変更や改善が要求される製品にも組込みソフトウェアが用いられるようになりました。その結果、ソフトウェアに対してメンテナンスの上にメンテナンスを重ね、保守容易性は確実に劣化し、瞬く間に理解しがたい複雑なプログラムに変わってきたと聞いています。

勿論再構築すれば良いのですが、投資費用と信頼性の低下及び即応性を考えると、決断がつかないまま難読なプログラムのメンテナンスという、現場の負荷が大きい作業が蓄積されていきます。そして最後は大きな障害が発生してようやく対策を講じるのが実態のようです。

2.2 大規模化にともない、プロジェクト運営が難しくなりトラブルを招く

小規模開発の現場ではまだ顕在化していませんが、開発対象のソフトウェア規模が大きくなるほど開発やメンテナンスの生産性は低下します。またそれに伴い、内在するバグを試験で発見できる確率も落ちるので品質の低下も免れません。その原因は、システムの複雑さとプロジェクト組織の効率低下にあります。^{注1}

この問題が深刻なのは、組込みソフトウェアの規模が年々大きくなっても、大規模プロジェクトの運営経験がない従来の開発体制、開発手法をそのまま踏襲してしまうことです。ここからが大規模で“要注意”という閾値は、もちろんありません。関係者が気付かないまま、大規模プロジェクトのトラブルの罠にはまるのです。何回かの失敗を重ね、大きなコストを費やしてから、ようやく大規模特有のプロジェクト運営の重要性を知ることになります。

2.3 組込みソフト開発現場の上位職や経営者層は機械設計技術者が多い

組込みソフトウェアの開発チームの上位職や経営者層には、機械製品の設計者や開発経験者が多いと聞きます。

注1：100万ステップの生産性は10万ステップの生産性の1/2～1/3であることが実績で明らかになっている。その要因は開発規模増大によるシステムの複雑化とプロジェクト組織の効率の低下にある。[03]

したがって本格的なソフトウェア工学を学んで、ソフトウェアの性質や特徴を、身を以って体験した人は少ないと考えられます。ソフトウェアの品質管理は機械の品質管理と勝手が違います。^{注2} 摩耗や寿命は無いが潜在するバグを完全に潰せません。形状や色を見て劣化を判別することもできません。同じ機能で同じ性能の物を簡単に複写出来ます。また主たる目的のソフトウェア以外に、周辺の付随的なソフトウェアが沢山存在していて、それらのソフトウェアの信頼性やデファクト標準化の動向が、製品の中長期寿命に大きな影響を与えます。さらにソフトウェア製品が機械製品と一番大きく違う点は、設計から制作、テスト迄の工程の大半が感情を持ち能力差の大きい人間集団でおこなわれることです。

ソフトウェア開発の経験が浅い上位管理者にとって、このような違いを乗り越え、過去の経験則を捨て現場のソフトウェア開発責任者の意向や判断に対して的確な支援を行うことは、かなり難しい事です。気がつかないで誤った指示や判断をしている事が多いのです。理想をいえばこれからの組込みソフトウェア開発チームの上位管理者は機械の設計や製作を経験し、更にソフトウェアの開発やメンテナンス経験を有することが最善です。

2.4 3K 職場と噂され、学生に敬遠されている

学生の間には情報サービス企業やコンピュータプログラマは3K職場であると半ば信じられています。これは組込みソフトウェアの開発も業務ソフトウェアの開発も同じように区別なく信じられているようです。確かにプロジェクトでトラブルが発生したり、工程が遅延すると、労働基準法を超えた残業を強いられることもあります。^[05] また責任ある立場になり、任された仕事の大きさや重要度、そして失敗した時の損失を思うと、重圧で身震いすることもあります。しかし目標を達成した時やプロジェクトが成功した時の達成感は、何ものにも替えがたいものです。

どうして3K等と言われているのでしょうか。その理由は業界の垂直型下請け構造にあります。構造の底辺に位置する技術者はキャリアパスの機会を奪われ、労働の対価は意味のない中間マージンで目減りし、最後は便利屋的に扱われる傾向にあります。優秀な技術者であるほど将来への展望が開けない状況が、一部で存在することは認めなければなりません。このような現実をみて学校や学生が、この職種に対して3K職場とみている節があります。^{注3} 結果として優秀な学生たちが敬遠することになり、ソフトウェア開発の職場にポテンシャルの高い人材が配置されない状況が続きます。将来の日本社会や経済を担うべきソフトウェア技術者のレベルが下がることは、ゆゆしき問題です。

注2：ソフトウェアで品質確保が難しいのは、これまでの品質管理手法が全く通用しないことだ。「04」

注3：3K論の実態調査として、社会人向けに「他産業との就業満足度比較調査を実施した結果、情報

サービス産業は6業種中6位。」[06]

2.5 組込みソフトウェアセキュリティの脆弱性

組込みソフトウェアの普及率が上がるにつれ、ネットワークに接続されるシステムや製品が増えています。その場合接続されたネットワークや無線 LAN を通じて外部の意図的な攻撃にさらされる危険性が高まります。^[07] クローズした世界で成長してきた組込みソフトウェアのセキュリティ環境は脆弱なことは間違いありません。サイバ攻撃をうけると、いとも容易く侵入を許す可能性があります。製品が社会的に重要なインフラであったり、人の生命に関わるものであればあるほど、テロの格好のターゲットになります。

3. 業務ソフトウェアが辿った道(失敗の教訓を生かす)

業務ソフトウェア開発者達が幾つかの経緯をたどって組織的な活動の必要性に気がついたのはかなり早い時期でした。1970年代の半ばには、IBMを始めとする米国コンピュータメーカーの指導により幾つかのユーザ企業で大きな開発プロジェクトが動き出しています。職人的な技術者たちが設計したシステムを、職人のみで開発・運用・メンテナンスする形態から、誰でも理解しやすい平易なロジック、平易なファイル操作を目指した開発・運用・メンテナンスの試みがなされ始めました。併せて生産性の向上を実現する幾つかのツールや手法も紹介され始めました。このような試行錯誤を経て、現在に至っていません。その過程の中から、組込みソフトウェアの世界でも今後同じような問題が発生すると考えられるものを列挙いたします。

3.1 部品化を目指した国家プロジェクトが失敗

1985年に通商産業省(現経済産業省)の主導で、シグマシステム計画が始まりました。当時のソフトウェア開発は人海戦術で行われており、このまま行けば2000年にはソフトウェア技術者が65万人も足りなくなる、そこで生産性の向上を目指してソフトウェアの部品化を促進し、生産工程をできるだけ自動化すべきである、と言う高邁な理想を掲げて進められました。しかし250億円の国費を費やして、失敗に終わりました。^[08] その原因はコンピュータメーカーを中心としたスキームが、本来の部品化の研究よりも開発環境のデファクト標準化競争やコンピュータメーカーの主導権争いに走った事にあります。しかしソフトウェアが本来もつ多面的な性質を考慮しないで製造業が工場部品を組み立てるがごとき感覚を適用すること自体が正しかったかどうか、その後の開発過程を見る限り判然としません。

オブジェクト指向の考え方が部品化を実現しやすいと言われますが、厳しい性能要求に対してはまだ研究途上にあります。いずれ組込みソフトウェアの世界でも、同様に部品化が叫ばれるはずですが、その時はこの失敗を教訓にし、同じ轍を踏まないでください。

3.2 大規模プロジェクトでトラブル続出

JUASの調査によるとシステム構築プロジェクトでQCD(Quality、Cost、Delivery)

の一つ又は二つ以上が、当初計画通りに進まなかった割合は 68%にもなるとの事です。

注4

特に大規模なシステムほどその割合と影響が大きくて、今に至るもトラブルは撲滅できていません。この原因は大規模、大人数のプロジェクトほどコミュニケーション・ルートが複雑になり、指数関数的に増大するからです。更に社会的・経済的な要請で短期間のシステム構築が望まれ、本来の適正規模を超えた人員構成になりがちなのがトラブルに拍車をかけています。その為プロジェクトマネジメントの強化が叫ばれ、多くの関係者が挑戦してきました。この努力が実を結んで少しはトラブルの発生が少なくなってきたように思われます。

しかし IT の世界は技術進歩が速すぎます。プロジェクトの大規模化によるプロジェクト運営の難しさに加えて、新技術採用のリスクが加わります。新技術（新ハード、新 OS、新ミドルソフト、新言語、新開発手法等）は幾つかの先行事例で成功していても、大規模システムに適用した場合はまた別の新たなトラブルが発生する確率が高いものです。更に技術習得の時間や適用のカスタマイズで、プロジェクト初期の生産性は必ず悪化します。プロジェクトの環境（スコープ、システムの種類、要求性能、期間、技術者レベル、ユーザの関与度等）は、いつも同じではありません。環境に即したプロジェクトの計画と実行がポイントで、ここはかなりの経験と行動力が必要になり、実際にプロジェクト開始後の状況を判断しながら計画の軌道修正を臨機応変に行う必要があります。この問題は組込みソフトウェアでも全く同じはずです。生半可な挑戦が痛い目にあうことは間違いありません。

3.3 ERP パッケージ以前の基幹システムはスパゲッティ化

製造業の基幹系業務システムは現在 ERP（Enterprise Resource Planning）、CRM(Customer Relationship Management)、SCM(Supply Chain Management)を実現するパッケージに変わりつつあります。それぞれの企業の特徴や戦略を反映するシステムを除いては、パッケージを利用する気運が日本でも芽生えました。この変遷の大きな理由の中に、過去にスクラッチで構築したシステムが、コスト、人材、期間等の面でメンテナンスや再構築が難しい状況になった事が挙げられます。メンテナンスを重ねてどうにか稼働しているシステムのソフトウェアは、スパゲッティ化していて保守容易性が極端に劣化している、現存するドキュメントは開発当初のままで現在のシステムと似て非なるもの、企業の経営環境変化が早くなりシステムの修正がタイムリーに即応できない、など袋小路に入りこんでいました。ある意味パッケージは救いの神だった訳です。

注4：大規模（500人月超）では予定工期より遅延（52%）、予定予算より超過（50%）、品質に不満（36%）。[09]

しかし組込みソフトウェアの世界では、製品の競争優位と差別化に直結するソフトが大半ですから、共通パッケージの流通はありえません。従ってこのスパゲッティ化をどのように防ぐかが、今後の重要課題になります。

3.4 ソフト・ハードの技術進化が早く技術者の世代間断絶が発生

この問題は短期的には影響ありませんのであまり騒がれていないようですが、技術の変わり目には大きな問題として私達（業務ソフトウェアの関係者）の前に立ちふさがりました。業務ソフトウェアの場合は、過去に2回大きな技術の変わり目がありました。1回目はメインフレームからオープンアーキテクチャに移行した時です。言語はもとより OS、ミドルソフト、開発ツール等が全て変わりました。それと同時に設計の手法やドキュメントの作成、開発環境等も従来の概念を変えてしまったのです。既成概念にとらわれた経験者より新人技術者のほうが理解が早く生産性が高いことが多く、プロジェクトの現場では旧世代技術のシステムは従来の技術者、新世代技術のシステムは旧世代技術管理者+新世代技術者という構成になることは必然の成り行きでした。

ところが新技術の経験がない旧世代技術管理者は新人技術者に対して技術指導ができないばかりか、重要な判断や問題の洞察もできないので、開発に関する主導権は新世代技術者に移っていきました。そして技術優先・マネジメント軽視の風潮がはびこったのです。管理者は新世代技術の分野では、リーダーシップを取れませんが、プロジェクトマネジメントや組織行動などでは先輩・上司として教育できることが沢山あります。特にプロジェクトマネジメントは OJT の中で継承すべき重要な能力です。この重要な能力の引き継ぎがないがしろにされ、途切れてしまった結果、過去に犯したミスと同じようなミスを世代が変わっても繰り返し、多大のコストと機会の損失を招きました。

2回目はクライアントサーバからウェブ（インターネット環境）に変化した時です。技術が変わっても変わってはいけない経験知や技術ノウハウが沢山あります。それを関係者で認識して、共有、継承する努力を怠ってはならないと思います。

4. 対応策の提言

組込みソフトウェアの開発・メンテナンスの問題は、業務ソフトウェアの開発やメンテナンスの問題と本質のところでは根を同じくすると考えられます。従って業務ソフトウェアで得た経験や知見から、組込みソフトウェアでも有効と考えられる対応策を、幾つか提言いたします。

4.1 政府や業界が指針を示し、それに基づいて部品化を推進する

日々の開発現場で部品化の推進を徹底するべきだと思います。この為には日常の開発チームとは別にこの目的の為に専任者が必要です。単に部品化するだけでなく、再利用しやすくする仕掛け、新しい部品を追加する仕掛けが最も大事です。また部品化に当たっては「も

の」と「こと」を意識した部品設計をすること。^[10] 自社の規模が小さい時は思いを同じくする他社との連携も一考に値すると思います。しかしその前に業界が部品化の指針を提供することが必要です。部品化のコンセプトやインタフェース等の枠組みを決めて誘導することで、製造業やソフトウェア企業の競争環境を残しながら、必要な時は部品レベルで相互利用が可能な世界を目指すことが必要だと思います。

二宮尊徳はこのように言っています。「遠きを謀るものは富み、近きを謀るものは貧す。それ遠きを謀るものは百年の為に松杉の苗を植う。・・・・・・・・ただ眼前の利に迷うて、蒔かずして取り、植えずして刈り取る事のみ眼につく。ゆえに貧窮す」。^{注5}

4.2 品質管理は工程全体を通して複合的に実施する

組込みソフトウェアの性格上、出荷した後にソフトウェア不具合が見つかるとう修復に多大な費用が発生し、信頼性は致命的なダメージを被ります。その為業務ソフトウェアと比較しても組込みソフトウェアのテスト工程は、非常に沢山の労力と時間を注いでいるようです。しかしソフトウェア品質の向上は、上位工程も含めて工程全体を通して品質管理を実践することでさらに効果が発揮されます。具体的には、①プロジェクト管理標準の導入②開発共通フレームの採用、③開発技法の確立、④CMMI (Capability Maturity Model Integration) 等組織の高度化、⑤PMO (Project Management Office) 等プロジェクト管理体制の実現等です。品質管理を組織として実施することによって QCD の底上げを図り、人による品質のバラツキを防止します。大規模開発チームでは既に実践されていると思いますが、小規模チームでも近い将来規模拡大期を迎えた時に必ず役に立つと考えます。

品質管理プロセスも見直す必要があると思います。特にメンテナンスを実施した後のテストや検査・承認プロセスにおいて、システムの品質を確保することが重要です。このプロセスは面倒でも組織的・計画的に関係者で共有することで、人為的なミスの発生を最小限にとどめることができます。またスバゲティ化したシステムには、波及分析ツールの活用が有効です。現段階では完璧なツールを求めるよりも補助的なツールを組み合わせる利用のほうが実用的です。

4.3 大規模プロジェクトのマネジメントと人材確保

既に大規模なチームになっている場合は、プロジェクトマネジメントの御苦労が多いと思われる。そしてマネジメント人材の不足が顕著ではないでしょうか。この解決策は第一にマネジメントの標準化です。大規模開発プロジェクトでは PMBOK (Project Management Body of Knowledge) などの世界標準のマネジメント方法論が存在していますから、早急に取り入れて無用のトラブルを減らす必要があります。

注 5：二宮尊徳の報恩思想を表す名言の一つ

第二は有能なプロジェクトマネージャの育成です。業務ソフトウェアの世界ではプロジェクトマネージャ育成の重要性がようやく認識され、あちこちで講演・模擬体験研修などの教育投資が行われています。しかし一番効果的な育成は、実際のプロジェクトでPMOの一員として経験させる事です。それに加えて大規模なプロジェクトのマネージャはリーダーとしての資質が問われることとなります。技術的な能力だけでなく、時には胆力や政治力も求められます。プロジェクトの規模や環境（ユーザ、投資規模、競合状況、人材動員力等）によって、プロジェクトマネージャの適否が変わります。何よりも人選をする事業責任者クラスの人達が、この点を認識しなければいけません。

業務ソフトウェアの世界で育った人との人材交流も、これから必要な事です。組込みソフトウェアが業務ソフトウェアと接続されたり、サブシステム化する場合、経験者同士が交流を重ねてお互いの領域に踏み込まなくては良いシステムはできません。特にシステムライフサイクルに対する考え方は、相当程度違うと思われます。このような交流を実施・促進出来るのは企業の経営者です。同じ釜の飯を食って同質化している組織に、一石を投ずることの重要性は今さら言うまでもありません。自社だけの限られた環境では、大規模プロジェクトマネージャを育成するには時間がかかります。実績のある外部人材の活用なども有効な手段です。

4.4 ソフトウェアの知財化を急ぐ

ソフトウェアの知財化も重要なテーマです。特に組込みソフトウェアは製品性能に直結して競争力の源泉ですが、意外と確固たる理念と信念のもとで遂行されているとはいえないようです。知財化はソフトウェアのモジュール化や部品化とも関連していますので、相乗効果が得られます。しかしこれはソフトウェア開発の現場任せではできません。会社全体のミッションとして明確に掲げて推進するべきと考えます。

更に注意すべき点は海外も含めた知財の外部流失です。今や外部リソースの利用が当たり前になっているので、直接の発注先へは注意も行き届いているようです。しかしグローバル化されつつある今日、発注先（外部委託先）のその先は中国やインドであったりします。自動車産業では発注先を複数にしてリスク分散をしたはずが、それぞれの再委託先が同じだったためリスクヘッジにならなかったということが、今年の3.11やタイの洪水で明らかになりました。外部発注した時その先の実態は、意外と見えないものです。戦略的な知財が垂れ流しになる危険性を防止し、有効な成長戦略を描く為に、組込みソフトウェアを知財化し、管理し、活用する環境を整え、その上で知財の流失防止策を考えることが急務です。

4.5 コンテキスト・アウェアとデータ処理に留意する

コンテキスト・アウェアとは、対象の状況を理解した上で判断する概念とされています。これまでの組込みソフトウェアは対象となる相手は機械が主でしたが、適用領域が拡大し

システムの相手は人間であることが増えてきています。例えば自動車のアクセルやブレーキ制御、携帯電話の操作等です。あらかじめ決まった反応や動作をしてくれる機械と違い、人間の場合は想定外の動きになることがしばしばあり、また感性の領域にまで踏み込んでシステムが対応しなければならないのです。このような状況は業務ソフトウェアでは多く経験済みです。

さらに組込みソフトウェアもシステムによってはデータ処理の概念が入ってきます。これまで大量のデータを扱うことが少なかった組込みソフトウェアはソフトウェアと独立したデータの概念が明確にされていないと思われます。データ処理は業務ソフトウェアではシステムの基本であり、データ中心のシステムはプログラムの機能よりもデータ構造や定義の設計に重きが置かれています。この意味を早期に理解する為には、業務ソフトウェアの経験者との交流が欠かせません。

このような新しい流れの中での的確なソフトウェアを制作する為には、業務ソフトウェア経験者の知見を利用する事をお勧めします。相互交流をすることでお互いの違いを認識し、課題に対応することが望ましいと考えます。

4.6 セキュリティ対策

残念ながらこの問題に絶対の処方箋はまだありません。先ず関係者がその重要性和重大性を認識すること、現在の水準でできる限りの対応策を講ずること、問題が生じた時は製品の利用者も含めた対処方法を身につけることが最低限望まれます。^[07]

組込みソフトウェアにとってセキュリティ問題は二重の意味で重要課題です。ネットワークに接続されるようになったことから、外部からのサイバ攻撃やウイルス感染にどう対処するかという観点、もう一つは製造する製品の機能や性能を決定つけるノウハウの集積である組込みソフトウェアの機密を守る為、物理的な防衛策と人間系を含めた流失防止システムの構築という観点です。

どちらもすこぶる難題であり、一朝一夕に解決できることではありません。しかし完全を目指せなくても智恵を絞ってできることだけでも早急に実現すれば、ある程度の防御は可能です。新興国に追い上げられていると言われている日本の製造業ですが、比較優位を保ち、更に研究開発の成果をおり込んで再スタートする為には、この課題の克服は経営者、事業責任者が自ら取り組む必須条件になります。

5. 組込みソフトは日本の社会や製造業企業の浮揚の鍵を握る

組込みソフトウェアは製品競争力の源泉であります。日本の製造業は幾つかの理由で新興国の追い上げにあい、製造拠点を海外に移しています。しかし組込みソフトウェアは日本国内に残して、今以上に付加価値を高めるべきです。事前調査先で「組込みソフトウェアは“擦り合わせ”が重要なのでオフショアには向かない」という話がありました。日本人の国民性、中でも協調性はチーム活動の必要なソフトウェア開発やメンテナンスに向い

ています。かつて半導体が「産業の米」と言われたように、組み込みソフトウェアは日本にとどまって「産業の知恵」となるべきです。日本社会の発展の為にもそうすべきです。是非ともこれから遭遇する高いハードルを越えて世界に先駆け大きく成長し、再び日本の製造業を大躍進させてほしいものです。日本製造業浮沈のカギを握っているといっても過言ではありません。

終わりに

本提言の作成に当たり、2010年10月～2012年4月にかけて大学の研究現場と企業(3社)のソフトウェア開発現場を直接訪問しインタビューを実施しました。それぞれ2時間程度の中で貴重なご意見や知見をご教授頂きました。ご担当の皆様には、我々の素朴な質問にも、丁寧に対応していただきました。この場をお借りしてお礼を申し上げます。

以上

参考文献とリンク先（カッコ内の日付は確認日を示します）

- [01] Wikipedia の「組込みソフト」より。(2012年5月31日)
- [02] 株式会社東日本技術研究所、「事業内容」より。
この内容は、以下のURLからダウンロードできる。(2012年5月31日)
(<http://www.tounichi-g.co.jp/business/microcomputer.html>)
「現在の車には数多くのマイコンが搭載され、組込まれている制御プログラムの
ステップ数は、地方銀行の勘定系システムに匹敵するといわれています。」
- [03] 芳賀正憲、「連載・情報システムの本質に迫る・第11回」、情報システム学会
メールマガジン、P4 2008年4月25日.
- [04] 日経 BP 社トヨタリコール問題取材班、「不具合連鎖—「プリウス」リコールからの
警鐘」P73（これまでの品質管理手法が通用しない）日経 BP 社、2010年3月23日
- [05] 経済産業省、「組込みソフトウェア産業実態調査」報告書 2010年版、「技術者個人向
け調査 Q3-5 月平均実労働時間」、2010年6月.
- [06] IPA(独立行政法人情報推進機構)、「IT 人材市場動向調査 調査報告書概要版 NO3」
第6章、2009年3月27日.
- [07] 萱島信、IPAセキュリティセンタ、「組み込みセキュリティセミナー「組み込みシス
テムセキュリティの取り組み」講演資料」、2012年03月27日.
- [08] Wikipedia の「プロジェクト」から。(2012年05月31日)
- [09] 日本情報システム・ユーザー協会、「企業 IT 動向調査 2009 報告書」、日本情報ユー
ザ協会、2009年5月.
- [10] 中村善太郎、「シンプルな仕事の発想法」(第二部)、日刊工業新聞社、2003年11月
20日.