

組込みソフトウェア開発の課題と対応

近 藤 満

ピースラッシュ株式会社

E-mail:mitsuru.kondoh@bslash.co.jp

要約：日本の強い国際競争力を持った製品のほとんどは組込みシステム機器であり、それらには組込みソフトウェアが搭載されている。半導体技術の脅威的な進展、ネットワーク性能の向上に伴い、これらの機器の複雑化・多機能化が急であるが、その実現を支えているのが組込みソフトウェアである。元来ソフトウェア開発は属人性の高いものであり、個人の経験やノウハウに品質と生産性が依存しがちである。また、見えないので管理が困難であるという特徴がある。システムのソフトウェアの規模が小さく、小人数開発が可能な時代には、個人に依存し、結果管理の開発でも大きな問題が生じなかったが、従来通りの開発方法では、急激なソフトウェアサイズの肥大化・複雑化には対応し切れなくなっているのが現状である。本稿では、このような組込みソフトウェア開発の課題とそれへの対応策について考察する。

キーワード：組込みシステム機器、組込みソフトウェア開発、V字モデル開発、モデル駆動開発、アーキテクチャ開発

1. はじめに

液晶テレビ、DVDレコーダー、デジタル複合機などに代表される、日本の強い国際競争力を持った製品のほとんどは組込みシステム機器であり、それらには組込みソフトウェアが搭載されている。半導体技術の脅威的な進展、ネットワーク性能の向上に伴い、これらの機器の複雑化・多機能化が急であるが、その実現を支えているのが組込みソフトウェアである。日本製造業の競争力の基盤となる技術として注目・期待されている。しかし、

¹ 本稿は2007年9月12日開催のコンピュータ産業研究会での報告を福澤光啓（東京大学大学院）が記録し、本稿掲載のために報告者の加筆訂正を経て、GBRC編集部が整理したものである。文責はGBRCに、著作権は報告者にある。

経済産業省が 2004 年から 4 年連続で実施した、組込みソフトウェア産業の実態調査から明らかにされたことは、日本の組込みソフトウェア開発の国際競争力への疑問である。

元来ソフトウェア開発は属人性の高いものであり、個人の経験やノウハウに品質と生産性が依存しがちである。また、「見えない」ので管理が困難であるという特徴がある。システムのソフトウェアの規模が小さく、小人数開発が可能な時代には、個人に依存し、結果管理の開発でも大きな問題が生じなかったが、急激なソフトウェアサイズの肥大化・複雑化に、従来通りの開発方法では対応し切れなくなった現状が読み取れる。

肥大化速度に個人の生産性向上は追いつかない。単に開発メンバーを増員しても開発スピードは比例しない。多くの技術者に均質な開発を期待するには、属人性を排除し、共通の規範となる開発プロセス・手法が要求される。工業製品では生産プロセスに高い品質と生産性が要求されるが、製造工程が存在しないソフトウェアでは開発工程に高い品質と生産性が要求される。本稿では、日本における組込みソフトウェア開発の課題とそれへの対応策について考察する。

2. 日本の組込みシステム産業と組込みソフトウェア産業実態

(1) 強い日本の組込みシステム産業

経済産業省（2004-2007）によれば、フォーチュン 500 社にみる世界と日本の組込みシステム関連企業の売上高割合のうちで、日本企業が 31%、海外企業は 69%を占めている。500 社中 75 社（うち日本企業は 25 社）が組込みシステム産業であり、75 社の売上高合計は 2 兆 7500 億ドルである。さらに、日本製組込み製品の市場シェアは高い（概ね 50%以上であり、80%以上のものが多い）。このように、日本の組込みシステム産業は高い国際競争力を持っているのだが、経済産業省の調査によれば、それを支えている組込みソフトウェア開発の国際競争力は必ずしも高いとは言えない。

(2) 組込みシステムの中核技術

組込みシステムの中核技術として、ソフトウェアと半導体が挙げられる。家電製品や AV 製品、OA 機器、産業機器、通信機器、自動車などの組込みシステムにおいては、ソフトウェアが機能・性能の源泉であり、それ無くしては製品機能・性能を発揮することは出来ない。半数以上の機器において日本製の基本ソフトウェアが利用されている（利用数が多

いものから順に、ITRON 仕様、Linux、Windows 系（CE 以外）独自開発となっている。ファンクション（アプリケーション）・ソフトウェアも日本製であり、これらは、ほとんど市場で流通することはない。また、組込みシステムでは日本製のマイクロプロセッサが多用されており、システム LSI も日本製である。

(3)組込みソフトウェア産業の実態

経済産業省によって 2004 年から 4 年連続で実施された、組込みソフトウェア産業の実態調査から明らかにされたことをまとめると次のようになる。組込みソフトウェア開発費は年率平均 12% で成長中であり、2007 年で 3.27 兆円の規模である。組込みソフトウェア技術者への要求数は、年率平均 12.1% で増大しており、2007 年には 9.9 万人の不足状況にある。開発プロジェクトにおける最大費用はソフトウェア開発費にあり、2007 年では 62.3% にまで及んでいる。2004 年の外部委託企業比率は 82% であり、米国（47%）や欧州（35%）に比して高い。2007 年における外部委託の最大の理由は、社内技術者の不足にあり、その最大課題は品質管理の難しさである。2007 年における内部開発工数比率は 41.9% であり、外部依存率が 50% を超えた。製品出荷後の不具合件数のうち最大の原因はソフトウェア由来のものである。手戻りの最大理由は、要求仕様の不備である。組込みシステム製品のコア技術であるシステム設計を 100% 自社で行っている割合は、日本では 13% であり、米国（36%）や欧州（25%）に比べて低い。組込みソフトウェア技術者のスキル標準・スキル評価制度を持っている企業の割合は、日本では 18% であり、米国（40%）や欧州（70%）に比べて低い。組込みソフトウェア技術者チームの平均労働時間は「180 時間以上」の割合が 64% であり、米国（40%）や欧州（16%）に比べて高い。2007 年時点で、開発プロジェクトにおける最優先課題として、技術者のスキル向上・育成であるという回答がなされている。

3. 組込みソフトウェア開発の課題

(1)組込みソフトウェア開発の課題と分析

組込みソフトウェア開発における課題として、 肥大化・複雑化、² 開発サイクルタ

² 半導体技術の急激な進歩に連れて大規模化・複雑化していること、組込み製品はスタンドアローンからネットワーク化へ、PC の周辺機器へと変化していることが挙げられる。

イムの短縮、³ タイム・ツー・マーケット、⁴ 品質の向上、⁵ 生産性（開発効率）の向上、⁶ 技術者のスキル不足⁷ が挙げられる。このような問題が生じる要因を分析すると、ソフトウェアの肥大化に個人の生産性向上では追いつけないこと（個人の生産性向上には限界がある）、多くの機能モジュールが必要となるが、ボトムアップの設計では有機的な機能分担が困難であり、それにともなってシステム全体の把握が困難となるので、問題が生じた時や変更要求が生じたときの適切な対応が困難であること、すべて自前の開発・設計では肥大化や開発期間要求に対応することが不可能であること（外部委託による品質低下や技術の流出・空洞化の懸念）、モジュール数の増加によってモジュール間インタフェースやシステムの複雑さが指数関数的に増大すること（開発工数はプログラム規模の冪乗になる）、開発要員の増加に対して開発スピードは比例しないこと（プロジェクトの人数が増えるほど生産性は低下する）、テスト依存では品質向上に限界があること（膨大な数のテストケースとテスト工数の肥大化）、そもそも日本人はソフトウェア開発が不得手であることが挙げられる。このように、依然として「ブルックスの法則」は健在である（Brooks, 1995）。

(2)ソフトウェア開発の特異性

「もの」の生産と「ソフトウェア」の生産は全く異なっている。「もの」の開発では生産プロセスに高い品質と生産性が要求されるが、製造工程の無い「ソフトウェア」では開発の品質と生産性が要求される。「もの」の開発に適した組織体制がソフトウェア開発の組織体制に適しているとは必ずしも言えない。また、ソフトウェアの生産は「企画 設計 実装」というプロセスを経る。ソフトウェアの生産は常に一品生産であり、品質や生産性は開発プロセスと開発手法・技法に大きく左右される。さらに、ソフトウェアは「目に見えない」ものである。目に見えないものは管理できないし、第三者にわかり難く、組織資産となり難い。ソフトウェアの開発は属人性が高く、個人の能力差が大きく反映され、組織の財産として蓄積できない。品質保証は大変であるが、手直しは容易である。欠陥ゼロを

³ 急激な技術進歩と競争激化により、新製品は発売と同時に陳腐化が始まる。また、長い開発サイクルは、サイクル内の技術進歩・競合製品の出現を許し、結果としてサイクル内での仕様変更を要求する。

⁴ 狙った時期に、狙い通りに製品化することが競争優位の原則である。

⁵ ソフトウェア製品出荷後の不具合が大きな割合を占めている。システムの大規模化・複合化に加えて、高信頼性へのさらなる要求がなされている。

⁶ 開発期間・開発経費の最小化、要因不足への対応。

⁷ スキル不足を要因とする障害が発生する。

達成するのは至難の業であり、手直しは容易であるが品質確認に大きな工数が必要となる。最も効率的なソフトウェア開発手法として、自主開発しないこと、再利用すること、出来合いを利用することが考えられてきた。

4. 組込みソフトウェア開発の課題への対応

(1) 組込みソフトウェア開発の課題への対応策

組込みソフトウェア開発の課題への対応策として、トップダウン設計の実施（将来の拡張性・発展性を見込んだアーキテクチャの開発、何を作るかを関係者に明らかにする、上流設計工程を重視し下流工程と分離すること）、抽象化設計の実施（システム全体を把握すること、設計の抽象度向上により生産性を高めること）、再利用方法の革新（コードの再利用から全ての資産の再利用へと進むこと、再利用粒度を拡大すること）、工学的手法の導入（開発プロセスや開発手法・技法の活用）、コンピュータパワーの活用（実行可能なモデル記述により V 字型品質保証から脱却すること、下流工程を自動化すること）、人材の育成が挙げられる。

(2) モデルベースのソフトウェア開発

モデルベースのソフトウェア開発を行うためには、そもそも、アーキテクチャの開発を行う必要がある。システム全体を貫くアーキテクチャを開発せずにシステムの構築を進めてしまうと、拡張性に乏しく機能追加が困難となる。それを無理やり実施しているうちにシステムは遺産化してしまう。まずは、「何を作るのか」ということを関係者にわかるように明確化することが重要である。そのためには、関係者に「わかる形」、「見える形」で何を作るのかを表現することが必要であるが、その鍵は「抽象化」と「可視化」にある。これらの実行手段としてモデル化がある。

図式化や抽象化することによって、複雑なものを記述できるようになる。エンジニアリング・モデルの主目的は、リスクを軽減することや、システム関係者に、実際に構築する前に複雑なシステムの特徴を理解してもらう手助けをすること、設計のアイデアを伝えることである。現在でも、ソフトウェアの世界でのモデル化の有効性についての議論がなされているが、1990 年代半ばのモデル記述統一言語（UML）の出現により着実に進展している。これによって、仕様書から完全なコードを自動生成するモデル駆動設計の道

が拓かれて、ソースコード中心の設計からモデル中心の設計への移行が進んでいる。

(3)ソフトウェアの再利用

ソフトウェアの再利用とは、ソフトウェアを楽に作ること、工学を上手く使うことである。究極のソフトウェア工学は、「できるだけ自分で作らないこと」である。再利用の種類として、あるものをそのまま利用する、カット・アンド・ペーストによる再利用、

既存リソースをリファクタリングして再利用、再利用資産の変動部をカスタマイズして利用するというものが挙げられる。求められる再利用のあり方は、品質低下が無く、設計を理解したうえで再利用されたものであり、資産の包括的な再利用が行われるというものである。再利用方法の革新として、コードの再利用から設計の再利用へ、さらには設計の再利用から仕様の再利用まで進展すること、結果の再利用からあらかじめ再利用を意図したソースを開発するようになること、粒度の大きな再利用を行うこと、包括的な資産の再利用を行うことが挙げられる。

(4)工学的手法の導入

ソフトウェア工学の要素と技術領域は、「再利用」、「validation/verification」、「テスト」、「開発プロセス」、「方法論」、「人材」である。開発プロセスについては、「要求分析 設計 テスト 保守」というライフサイクルを対象とした開発管理が行われている。ウォーターフォールや反復型、アジャイルという開発プロセスが代表的である。定義や計測、マネジメントのサイクルによってプロセスの成熟度を向上させている。

多くの技術者が均質な設計品質を達成するためには、共通規範となる設計方法論が必要である。それによって属人性の排除を行うことができる。設計対象を表現する記法として、プログラム言語やモデル表記法、フォーマルメソッドが挙げられる。設計手順を定める運用として、トップダウンアプローチ/データ志向アプローチや、抽象データ型/構造分析・設計/オブジェクト指向分析・設計、ドメイン・エンジニアリング/プロダクトライン・エンジニアリングがある。

(5)要求工学

システム開発の最大課題は要求仕様にあるといえる。要求仕様が不完全あるいは曖昧であったり、要求仕様が顧客の目標に合致していなかったり、顧客の求める要求仕様変更

追従できないということが問題となる。組込みソフトウェアの開発における手戻りの最大の原因は、要求仕様の不備・仕様書の不備にある。

要求工学の狙いは、仕様と開発目的のすり合わせ、あるいは曖昧な目的を明瞭なものに洗練化し、それを最大限忠実に表現する仕様を作成する（プログラムが完成する前に仕様のバグを発見する）ことにある。要求工学のプロセスは、「要求の抽出 要求分析 要求の妥当性確認 要求管理 ステークホルダー分析 ユースケース分析 変動性と機能の分析」となっている。

(6)モデル駆動開発

モデル駆動開発とは、ドキュメントとソースコードではなく、モデルを用いてソフトウェアを開発する方法論である。特定のプログラミング環境に依存せず、機能をモデル化し、そのモデル情報を基にコードが自動生成される。モデル駆動開発のメリットとして、ドキュメントに比べて曖昧さが減少すること、プログラミング言語に比べて高い抽象度で記述できること、モデルが実行可能であること（仕様変更発生時にすべてのプログラムを見直す必要が無くなる）、品質・生産性の向上、変化への応答速度向上、資産の蓄積がなされることが挙げられる。理想的なモデル駆動開発とは、仕様をモデルで記述すれば、その下流工程（詳細設計やコーディング）が大幅に自動化されるというものである。モデル駆動開発への評価としては、アセンブラから高レベル言語登場以来の大きな転換であるとする見方や、1980 年代の CASE ツールの登場と同じとする見方などがある。高水準言語によるプログラミングも、かつては CPU 上で実行可能な計算モデルに自動変換する自動プログラミングとして見なされていた。

V 字モデルによる品質保証では、「問題領域からの要求獲得（自然言語） 要求分析（自然言語） 設計（自然言語） プログラミング（プログラミング言語） コンパイラによる変換 実行可能な計算モデル 単体テスト 統合テスト システムテスト」という一連の開発プロセスにおいて、レビューによる合致確認や実行に基づく合致確認、実行に基づく妥当性確認が行われている（図 1）。これに対して、モデル駆動型開発による品質保証では、開発と検証が同時進行で行われている。すなわち、V 字モデルにおける問題領域からの要求獲得から実行可能な計算モデルの生成までの間に、検証も同時に行うことによって、残りのプロセスを省略することができるのである（図 2）。これを報告者は「Backslash Model」と称している。

図1 V字モデルによる品質保証

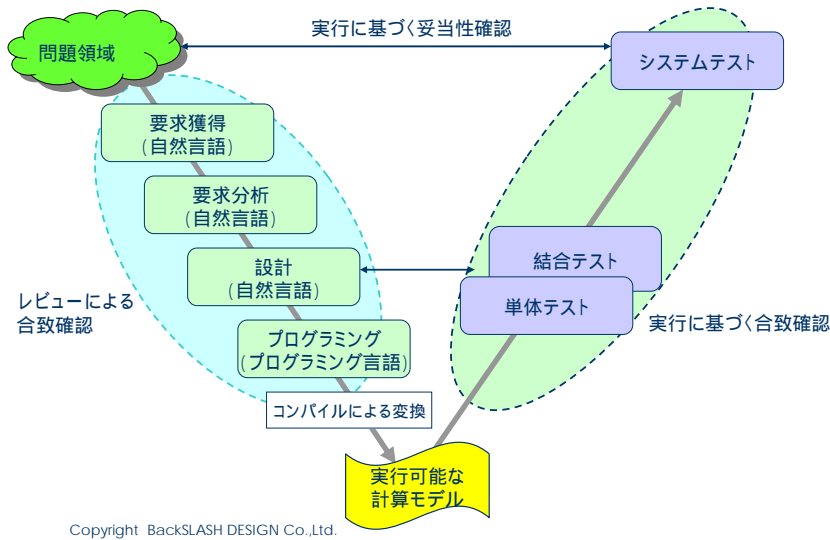
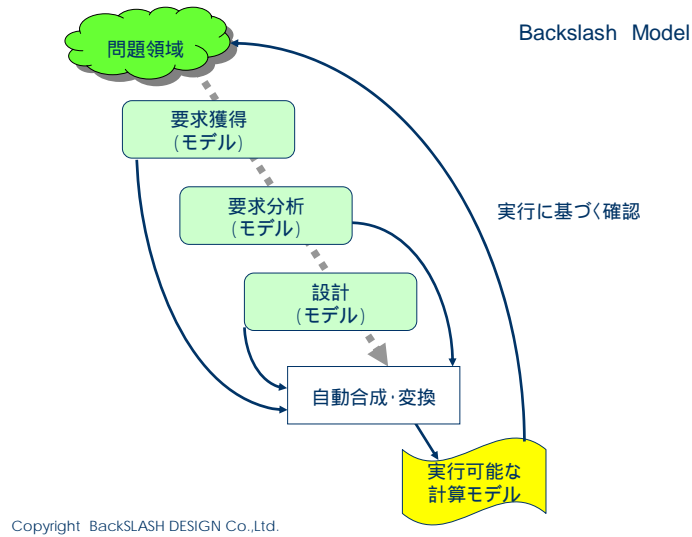


図2 モデル駆動開発における品質保証



(7) プロセス改革の達成とマネジメント

プロセス改革を達成するためには、専任組織によるプロセス改革活動が必要となる。しかし、乗り越えるべき壁もある。商品設計部隊による再利用設計は、特定の製品向けに

開発するより多くの工数を要し、リソースの継続的な維持・管理の工数を要し、再利用設計者への理解・評価が無く、技術のハードルが高いことから現実的ではない。さらに、設計部隊も自らプロセス改革を希望したり、自組織のコア技術に対して強く NIH が働いたり、自組織内で再利用が上手く出来ているという自負があり、他人の作ったソフトウェアを利用しない理由は豊富であり妥当性もあるため、専任部隊が開発するソフトウェア部品への抵抗は根強い。このような、プロセス改革に対する壁を打破するためには、トップダウン活動とボトムアップ活動を融合し、個々の設計者によるプロセス改革行動⁸が行われて、専任組織部隊と商品開発部隊との連携⁹が必要であると考えられる。

参考文献

- Brooks, F. P. (1995). *The mythical man-month: Essays on software engineering* (Anniversary ed.). Reading, MA: Addison-Wesley.
- 経済産業省商務情報政策局編 (2004-2007) 『組込みソフトウェア産業実態調査報告書』2004 年度版, 2005 年度版, 2006 年度版, 2007 年版.
- Lavagno, L., Martin, G., & Selic, B. (Eds.). (2003). *UML for real: Design of embedded real-time systems*. Boston: Kluwer Academic.

⁸ そのためには、技術者自身がソフトウェア工学を学習したり、再利用による品質と生産性を向上することが必要である。

⁹ このような連携が必要となる分野として、コア資産開発や商品開発、技術マネジメント、組織マネジメントが挙げられる。

赤門マネジメント・レビュー編集委員会

編集長 新宅 純二郎

編集委員 阿部 誠 粕谷 誠 高橋 伸夫 藤本 隆宏

編集担当 西田 麻希

赤門マネジメント・レビュー 6巻10号 2007年10月25日発行

編集 東京大学大学院経済学研究科 ABAS/AMR 編集委員会

発行 特定非営利活動法人グローバルビジネスリサーチセンター

理事長 高橋 伸夫

東京都千代田区丸の内

<http://www.gbrc.jp>