

組込みソフト開発における課題と対応策

株式会社日立情報制御ソリューションズ

組込みシステム本部 エンベデッドソリューション部 主管技師

飯島 三朗

業務サポート本部 生産技術部 主管技師

桜庭恒一郎

1. はじめに

近年、組込みシステムは、自動車・パソコン・デジタルカメラ・携帯電話などさまざまなハイテク機器で利用されています。

このような中、組込みソフトウェアは、使いやすく、便利で安全といった機能を実現する中核的な存在となっており、組込みソフトウェアの存在なくして組込みシステムの高機能・高品質は実現できないのが実態です。

一方、組込みソフトウェアの開発現場では組込みシステム特有の課題が多く、高機能、高品質を実現するには、組込みソフトウェアの開発プロセスや開発技術の改善が必要であると考えます。

2. 組込みソフト開発における課題と対応策

当社では、永年にわたる組込みソフトウェアの大規模受託開発や自社製の組込みシステム製品開発を通して、さまざまな課題に直面してきました。

開発時に発生した課題や問題点を解決するため、設計部門や品質保証部門、生産技術部門が協力して、社内に「組込みソフト品質向上WG」を設立してプロセス改革活動を続けています。以下、それらの具体的な事例を報告します。

改善した項目は多岐にわたるため、課題と対応策を、1) 開発プロセスに関するもの、2) 開発技術に関するもの、の2つのカテゴリに分類して説明します。

2.1 組込みソフトウェア開発プロセスに関する課題と対応策

(1) ソフトウェア開発ライフサイクルの改革

当社では、永年「ウォーターフォール型のソフ

トウェア開発ライフサイクルプロセス」を適用してきました。

当初、大規模組込みソフトウェアの受託開発においても上記のライフサイクルを適用して開発を進めていました。ところが、実際の開発では、部分機能を段階的に確認しながら開発を進めていくため、開発途中段階で一部の機能が不足していたり不具合が内在するサンプル品の顧客へのリリースを何度も行わなければなりませんでした。

この問題を解決すべく、サンプルリリースを前提としたインクリメンタル開発に適した「組込みソフトウェア向け開発プロセス」(図1参照)を新たに定義し、実プロジェクトで適用を開始しました。

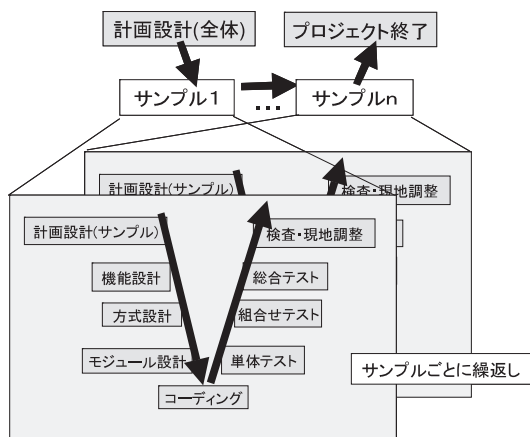


図1 組込みソフトウェア向け開発プロセス

新規に制定した「組込みソフトウェア向け開発プロセス」では、サンプルリリースごとの実装機能や制限条件などを全体計画立案時に明確化しておき、サンプル開発ごとに計画設計から検査・現地調整までのプロセスを繰り返します。

サンプルリリース時には、それぞれの出荷基準に基づいて品質保証部門が出荷可否の判定を行うこととしました。

(2) 多品種・大量生産品に対応した 製品リリース管理の強化

組込みソフトウェアは、海外の仕向け地対応や納入顧客対応などにより、派生開発を繰り返すことが多く、製品のレパートリーが増大する傾向にあります。

当社では、ソフトウェア開発の成果物である個々のドキュメントやプログラムソースコードは、それぞれの更新に際して、変更来歴やバージョン・レビジョン番号による管理を行っています。

また、ソフトウェア構成要素であるドキュメントやモジュールのバージョンの組み合わせ、すなわちベースライン管理も構成要素管理台帳などにより厳格に管理しています。これらの構成管理プロセスは規格化されており、組込みソフトウェア開発時にも適用していました。

ところが、組込みソフトウェアは、製品レパートリー、即ちバージョンの数が通常のソフトウェアに比べ格段に多くなるため、人手で帳票を作成しながらベースライン管理を実施することが限界に達していました。

ベースライン管理がうまくできないと、図2に示すように、顧客に納入した製品の不具合対策時に、機能デグレード（退行）の発生や、新たな不具合が発生するリスクが増大することになります。

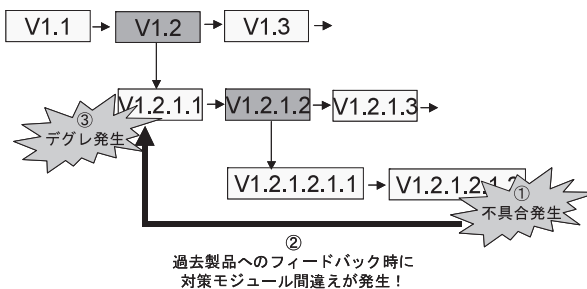


図2 派生開発におけるベースライン管理の問題

このため、自社製品の構成管理ツール SagePro®/CM*1) を開発する際に、ブランチ・マージ管理機能やラベルによるリリース管理機

能（図3参照）などの組込みソフトウェア開発向けの新規機能を追加しました。

この機能を活用することにより、派生バージョンが異なる並行開発も安全に行うことができ、過去にリリースした製品の構成要素と製品バージョンの対応付けが確実かつ容易に管理できるようになった結果、製品の品質向上、生産効率向上に繋がりました。

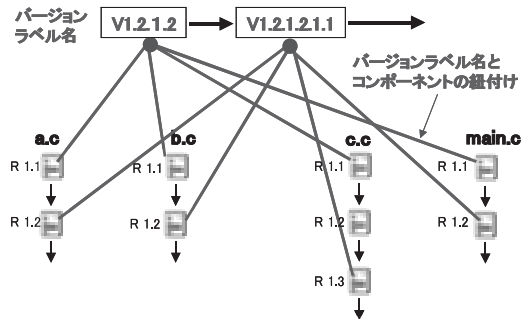


図3 ラベルによるリリース管理機能

2.2 組込みソフトウェア開発技術に関する 課題と対応策

(1) 派生開発

組込みシステムは、既存のドキュメントやソースコードを流用・改造して開発する、いわゆる「派生開発」が多く見受けられます。

実際の派生開発では、過去に開発した際のドキュメントなどが残っていないことが多く、ソースコードをよりどころにしなければなりません。このようなケースでは、ソフトウェア構造や仕様を十分理解できないまま改造を行うため、後工程で不具合が多発することになります。

この問題に対し、要求仕様や変更仕様を漏れなく確実に実装するための手法として、派生開発手法¹⁾を導入しました。

派生開発手法とは、ソース修正を実施する前に、①変更要求仕様書（What）、②トレーサビリティ・マトリックス（Where）、③変更設計書（How）を作成し、関係者とレビューを実施。その後、ソース修正を行う手法です。これにより、中途半端な「部分理解」を解消し、後戻り作業をなくしました。

*1) SagePro/CMは、株式会社日立製作所 情報制御システム事業部のソフトウェアを基に、株式会社日立情報制御ソリューションズが開発した製品です。
● SageProは株式会社日立情報制御ソリューションズの登録商標です。

(2) OSレスアーキテクチャからの脱却

組込みソフトウェアはハードウェアの付属品と思われているような風潮があり、組込み向けOSなどを採用せずに、OSレスで自前の組込みソフトを開発しているケースが多々あります。

具体的には、図4に示すように、個々の割り込み対応処理とフラグを対応付けて、イベント発生時にフラグを判定して処理を行っている事例がありました。

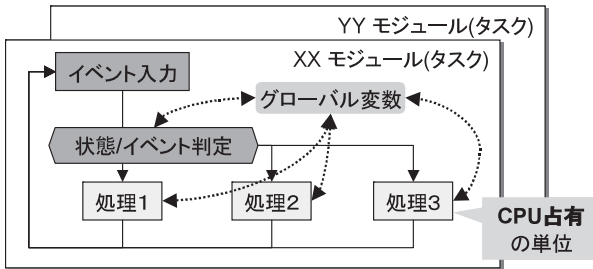


図4 フラグ判定によるイベント対応処理の例

フラグを使用したイベントドリブンなソフトウェアは、全体構造が見えにくく、新たな機能追加や変更が難しいものとなりますが、過去の実績のある財産を偏重して、ソフトウェア環境の進歩を享受しないまま改造を繰り返してきたことがうかがえます。

本来であれば、イベントで分割せずに、機能やデータ処理の単位で処理を分割することで、モジュール強度（品質）を強く、結合度を弱くすることができるはずであり、マルチタスクOSによる優先制御を利用（割り込み、高優先度タスクを優先して処理）することで、ソフトウェア構造をよりシンプルかつ保守のしやすいものにできます。

今後、組込みシステムの新規開発時には、ハード・ソフト全体のアーキテクチャを考え、共通処理と個別処理を分離して実装する、ソフトウェアプロダクトライン²⁾の考え方を基本的なシステムアーキテクチャとする抜本的な構造改革を進めていく予定であり、現在その実現可能性を調査・検討中です。

また、当社では、自動車制御システム分野で標準となっているリアルタイムOSなどの移植も手がけており、お客様のOSレスアーキテクチャからの脱却も支援しています。

(3) デバッグ効率の向上

近年、組込みソフトウェアの開発規模が劇的に増大しているにもかかわらず、従来の延長で「家内制手工業」的な開発を行っている状況から抜け出せていません。

このため、組込みソフトウェアの開発現場では、品質低下や工程遅延、開発コスト増大などの問題が常態化しつつあります。

組込みソフトウェアの開発環境に関しては、必ずしも充実しているとは言えず、限られたメモリ容量やOS機能の制限などにより、実機を使ってのデバッグ時には、メモリ内容を直接参照してトレース情報を解析したり、トレース文をソースコードに埋め込んで動作順序を解析するなど、経験と職人芸的なスキルが必要です。このため、デバッグ要員も限られ、要員ネックによる工程遅延なども頻繁に発生していました。

また、前項で述べたOSレスアーキテクチャからの脱却によりマルチタスク環境に移行すると、タスク間の連携を踏まえた上で、システム全体の動きを捉えることが必要になります。

このような問題を解決すべく、図5に示すように、実機のトレース情報を解析PC側でビジュアルに表示して、個々のタスクの動作や複数タスクの動作遷移などの解析を容易にするツールを自社で独自に開発して、デバッグ効率や性能解析などの工数を大幅に改善しました。

解析が容易になったことで、解析要員の不足も緩和することができました。（ツールは、

```

mdump #0xC0020010,0x50600 I
#*===== +0 +4 +8 +C
#*c0020010 54524396 00000001 00005000 c0055b90
#*c0020020 00030619 00000000 00000005 00000000
#*c0020030 c0020610 c0020090 c0020110 c0020190
:
    
```

トレース情報の可視化によるデバッグ効率向上



図5 SagePro/eDEBUGによるトレース情報の可視化

SagePro/eDEBUGの商品名で販売中です)

組込みソフトウェア開発の現場では、統合開発環境やCASE (Computer Aided Software Engineering) ツールなどの導入が進んできていますが、デバッグ支援ツールとの連携はまだ十分とは言えず、トレース情報とソースコードの対応付けができるようにするなど、更なるデバッグ効率改善を目指してツールベンダ各社との協業も進めています。

具体例として、キャッツ株式会社殿の開発支援用CASEツール「ZIPC®」とSagePro/eDEBUGを連携させ、状態遷移表を基にしたデバッグを可能としました。図6に画面例を示します。この連携により、今まで実機上ではなかなか確認できなかったZIPCの状態遷移表の動きが明確に分かるようになり、設計の検証、不具合の検出が容易になります。



図6 SagePro/eDEBUG-ZIPC連携画面例

さらに、株式会社ルネサステクノロジ殿の統合開発環境HEW (High-performance Embedded Workshop) との連携により、アプリケーションプログラムのシステムコール発行箇所をソース行レベルで表示するので、不具合箇所の特定が容易になります。

3. 今後の課題

(1) 複数モデル並行開発のプロジェクト管理

多品種製品の並行開発が要求される組込みソフトウェアの開発では、抽出した不具合の他の製品への展開や、情報共有のあり方など、複数モデル並行開発時のプロジェクト管理方法を確認する必要があります。

(2) 業界標準化動向への迅速な対応

自動車産業の分野では、JasPar^{※2)}などの標準化団体の活動が活発になっています。JasParのタスクフォースには当社社員も参画しています。迅速な対応ができるよう、今後の標準化動向を常に注視していく必要があります。

(3) 設計詳細化プロセスの確立

各種開発モデル手法を実装レベルに展開する際の、設計詳細化フェーズのドキュメント作成基準や手順が不十分なため、実プロジェクトでの適用を増やして、設計詳細化プロセスを確立していく必要があります。

4. おわりに

当社では、組込みソフトウェア開発方法論を確立すべく、「日本の組込みソフトウェア開発の未来を考える」をテーマに、産学連携の活動を開始しました。

また、SESSAME^{※3)}を始めとしたさまざまな団体との関係や連携もできつつあり、ビジネスの枠を超えた活動に今後も積極的に参画していきます。

当社は、ハードウェアとソフトウェアの両方の技術を有する特長があります。その特長を最大限に活かして、日本の組込みシステム分野におけるリーディングカンパニーを目指し、大きな志を持って活動していく所存です。

参考文献

- 1) 清水吉男：失敗しない派生開発 (Software People Vol.8) PP.2-69, 技術評論社発行 (2006.4)
 - 2) Klaus Pohl and Gunter Bockle and Frank van der Linden : Software Product Line Engineering, Springer-Verlag 発行 (2005.10)
 - ※2) JasPar (Japan Automotive Software Platform and Architecture) 自動車の電子制御ユニット (ECU) のソフトウェア基盤や車内LANインタフェース規格の標準化を推進することを目的に、国内の自動車メーカーや自動車エレクトロニクス関連メーカーが集まり、2004年9月に発足した団体。
 - ※3) SESSAME (Society of Embedded Software Skill Acquisition for Managers and Engineers) 2000年11月に設立された、NPO法人 組込みソフトウェア管理者・技術者育成研究会。
- ZIPCは、キャッツ株式会社の登録商標です。

株式会社日立情報制御ソリューションズ 営業本部 ソリューション第三営業部

部長代理：安倍 史英 (あべ ふみひで)
E-mail : fumihide.abe.sb@hitachi-ics.co.jp
主 任：左座 敦志 (さざ あつし)
E-mail : atsushi.saza.rj@hitachi-ics.co.jp