

巻頭論文

ソフトウェア開発環境の現状と展望



山下昭裕*



木槻純一**

Software Process Improvement and Development Environment

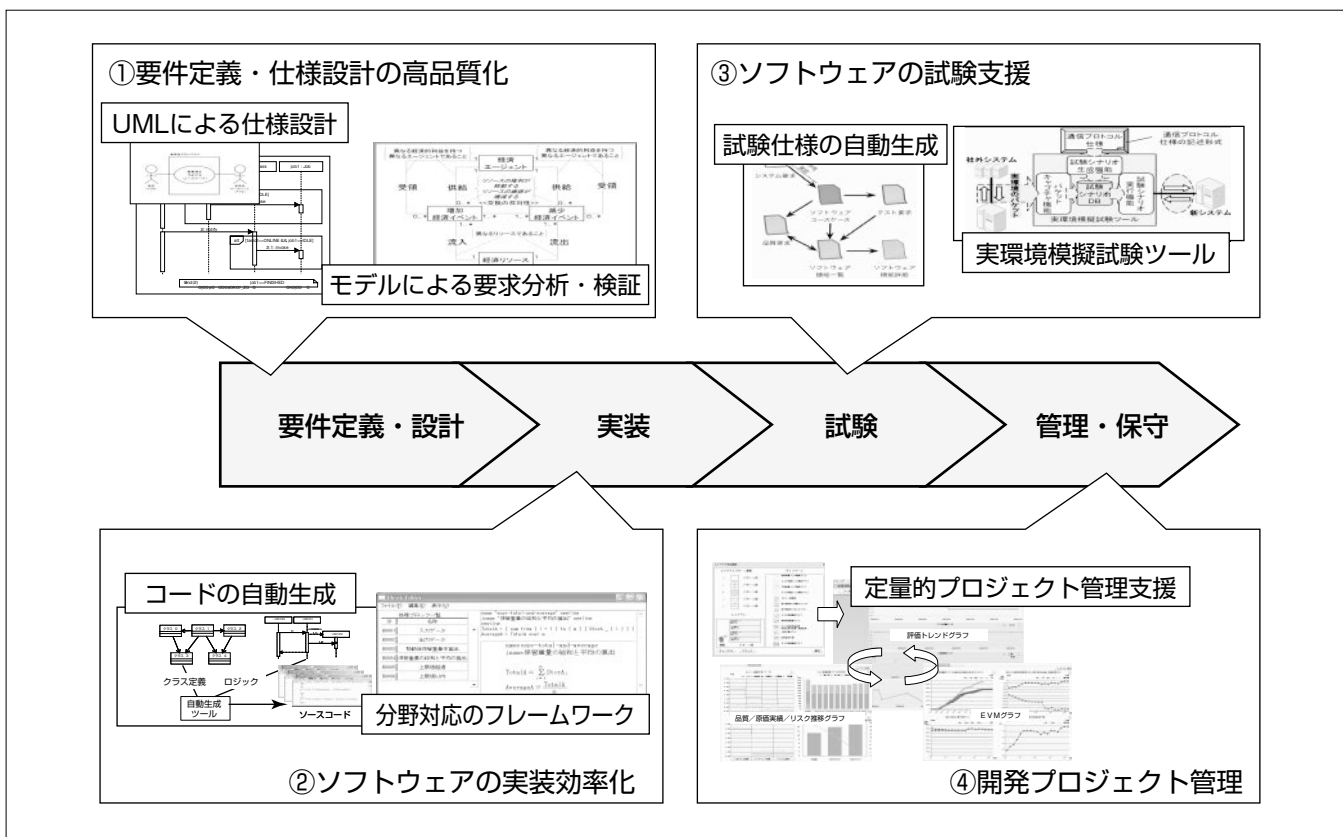
Akihiro Yamashita, Junichi Kitsuji

要旨

社会システムから家電製品に至るまで、機能の高度化が進展するとともに、製品投入サイクルも短期化しており、従来ハードウェアで実現されていた機能が幅広くソフトウェアで実現されるようになってきている。これに伴って、製品・システム開発におけるソフトウェア開発の位置付けが重みを増している。こうした観点から、三菱電機は、従来ソフトウェアの高品質化と開発の効率化を目指して、ソフトウェア開発プロセスの改善、効率的なソフトウェア開発を支援するソフトウェア開発環境の整備、ソフトウェアプロダクトラインによる開発量削減などに取り組んでいる。

この特集号では、ソフトウェアの生産性向上、高品質化に向けた、ソフトウェア開発環境に関する三菱電機の取組みについて、次の4つのポイントに整理して紹介する。

- (1) 要件定義・仕様設計の高品質化
ビジネスモデルに基づいて要件定義を行うとともに、UML(Unified Modeling Language)^(注1)による仕様記述によって、漏れの防止と仕様の高品質化を図る。
 - (2) ソフトウェアの実装効率化
形式的な仕様記述と、分野別のソフトウェア基盤を活用し、実装の効率化とともに、実装誤りを防ぐ。
 - (3) ソフトウェアの試験支援
仕様記述から、試験条件を自動抽出するとともに、自動試験環境を構築して試験の効率化を進める。
 - (4) 開発プロジェクト管理
ソフトウェア開発プロジェクトを“見える化”する。
- (注1) UMLは、Object Management Group Inc.の登録商標である。



三菱電機におけるソフトウェア開発環境への取組み

家電製品から大規模な社会インフラシステムに至るまで、三菱電機の幅広い製品群で、機能の実現・製品開発におけるソフトウェアの比重が高まっている。高品質のソフトウェアを効率よく開発するために、ソフトウェア開発の上流である要件定義・設計から、ソフトウェアの実装、ソフトウェア試験、保守開発に至る各段階に対して、開発プロセスの改善とソフトウェア開発環境の改良を進めている。

1. ま え が き

社会システムから家電製品に至るまで、機能の高度化が進展するとともに、製品投入サイクルも短期化しており、きめ細かい制御を実現するため、従来ハードウェアで実現されていた機能が幅広くソフトウェアで実現されるようになってきている。また、インターネット化の進展に伴い、従来のクローズなシステムと異なるソフトウェア構造、ユーザーインタフェース(UI)を持つシステムが増加している。こうした観点から、ソフトウェア開発の効率化を目指して、ソフトウェア開発プロセスの改善、効率的なソフトウェア開発を支援するソフトウェア開発環境の整備が必要となってきた。

2. ソフトウェア開発環境の動向

事業競争力の強化のためには、製品の差別化・付加価値向上が必要であり、ソフトウェアの重要性はますます高まっている。

それに伴い、ソフトウェアが大規模化、複雑化し、また事業のスピードアップのために短期開発のニーズも強く、ソフトウェアの開発力強化が急務の課題となっている。一方、電力・交通などの社会インフラ、携帯電話・AV機器など生活必需品となったソフトウェアの品質は重要となり、ソフトウェアトラブルが社会生活に多大な影響を及ぼすことも多くなっている。

こうしたソフトウェアの課題に対し、経済産業省はIPA/SEC((独)情報処理推進機構 ソフトウェア・エンジニアリング・センター)を2004年10月に設立し、ソフトウェア開発力強化に取り組んでおり、三菱電機も設立時から参画している。

ソフトウェア開発力強化には、プロセス、技術、人のバランスよい強化が必要である。

(1) プロセス

プロセスとは、ソフトウェアを開発するための手順や工程、成果物、進め方に関する基本的な考え方を定義したものである。プロセスを改善する手法・モデルとして、CMMI (Capability Maturity Model Integration)⁽¹⁾やISO/IEC15504、共通フレーム2007(Software Life Cycle Processes : SLCP)などがあり、アセスメント手法も整備され普及している。またプロジェクト管理面ではPMBOK (Project Management Body of Knowledge)⁽²⁾によって管理手法が体系的にまとめられている。

(2) 技術

要件定義から設計、実装、テストの全工程にわたって、モデリング手法が整備され実用化されている。分析や設計段階ではUMLによるモデリング手法が実用化されており、モデル検証の研究もされつつある。一方、ソフトウェアの

再利用技術としては、SPL(Software Product Line)型開発によって製品ロードマップに沿った再利用型開発手法が実用化されつつある。

(3) 人

人材育成の観点から、ETSS(Embedded Technology Skill Standards)などのスキル標準、キャリアアップ体系が整備され、企業への導入が加速されつつある。

3. 三菱電機の取組み

三菱電機でも、1980年代からエンタプライズ系のソフトウェア開発力強化活動を推進してきたが、組込み系ソフトウェアにおいて2章で述べた課題が大きくなってきたことから、両分野を合わせたソフトウェア開発力強化活動を2004年から全社的に展開している。この活動には、ソフトウェアを開発する事業所だけでなく、ソフトウェア開発を担当するグループ会社、ソフトウェア開発の先進技術を研究する研究所・技術センターも参画している。

この活動では、開発力強化のねらい(施策)として“フロントローディング化”に注力している。フロントローディング化とは、開発工程の上流設計段階で品質を作り込む手法である。フロントローディング化によって、上流での課題解決による手戻り削減と開発期間短縮が図れる。

三菱電機では、フロントローディング化を次の2段階で進めている(図1)。

(1) 第一段階

ソフトウェアの開発プロセスに着目し、従来、テストなどの後工程で発見していた不具合を要件定義や設計など上流工程で検出する活動である。手法としては、CMMIやISO15504プロセスモデルに従ったプロセスの定義と、定量データの収集・評価(見える化)を行う。設計不具合は設計レビューで検出し、実装の不具合は静的解析技術を使って検出する。

(2) 第二段階

ソフトウェア設計技術に重点を置く。不具合を検出するのではなく、不具合の混入を防止する技術である。そのために、モデルベースのソフトウェア設計・開発や、製品分野に合わせたソフトウェア共通プラットフォームの開発・利用やSPL手法を活用している。

第一段階、第二段階ともに、開発に携わる人の育成が重

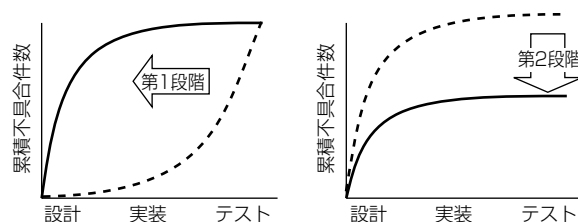


図1. フロントローディング化

要であり、スキル教育だけでなく、特にソフトウェアプロジェクトリーダーの育成に力を入れている。

4章以降に、ソフトウェアの開発環境に関する三菱電機の取組みを次の4つのポイントに整理して述べる。

(1) 要件定義・仕様設計の高品質化

ビジネスモデルに基づいて要件定義を行うとともに、UMLによる仕様記述によって、漏れの防止と仕様の高品質化を図る。

(2) ソフトウェアの実装効率化

形式的な仕様記述と、分野別のソフトウェア基盤を活用し、実装の効率化とともに実装誤りを防ぐ。

(3) ソフトウェアの試験支援

仕様記述から、試験条件を自動抽出するとともに、自動試験環境を構築して試験の効率化を進める。

(4) 開発プロジェクト管理

ソフトウェア開発プロジェクトを“見える化”する。

4. 設計の効率化・支援

ソフトウェアの開発で、システムが果たすべき機能・仕様を早い段階で確定・検証することが、ソフトウェア開発全体の効率化に寄与する。近年、オブジェクト指向によるソフトウェア要求分析・設計が広く採用されており、そのための仕様記述方法として図形言語であるUMLが使われる。特に、要求分析段階ではユースケース図が、設計段階ではクラス図、シーケンス図、状態マシン図などが使われている。さらに、対象分野に適した要求仕様モデルを使うことによって、要件定義の容易化とともに、仕様の完全性を検証する取組みや、仕様段階でソフトウェアの動作を模擬して検証するモデルベース開発などの取組みも広がってきている。

4.1 要件定義の高品質化・効率化

ソフトウェアの要件定義では、UMLのユースケース図を使う。しかし、ユースケース図は記述の自由度が高いため、記述レベルが書き手によって異なったり、要件定義に漏れが生じたりする可能性がある。そこで、記述規則やそれに対応したツールを用いて仕様定義を行ったり、機能モデルに基づいて仕様記述を行うことで、要件定義の高品質化を図る。

この特集号では、要件定義の高品質化への三菱電機の取組みとして、システム化の対象をREA (Resource Event Agent) モデルを活用してモデル化し、システム機能仕様を明確にする手法を紹介している⁽³⁾。

また、設計品質の向上を目的として、UMLで記述された設計情報を網羅的に検査するツールを開発した。このツールは、“形式手法”の一つであるモデル検査技術を採用しており、オートマトン／時相論理／グラフ理論を用いて、ある範囲のシステムの動作を網羅的に検査する。このツ

ルによって、サーバ間のメッセージシーケンス、Webサービスの呼出しシーケンスや通信プロトコルなどの仕様を高品質化することができた⁽⁴⁾。

4.2 要件定義・仕様設計における今後の課題

性能・可用性・信頼性といった非機能要求を形式的に記述することは、従来の手法では難しい。

この課題に対して、三菱電機インフォメーションシステムズ株式会社(MDIS)も参加した非機能要求グレード検討会が、非機能要求を明確化するためのガイドラインと要求項目一覧表⁽⁵⁾などを公表している。これは非機能要求を、可用性、性能・拡張性などの6種に分類し、各要求の水準を段階的に数値化することで、システムの非機能要求を“見える化”する。

ただ、こうした非機能要求を、詳細仕様設計や実装に反映する方法とソフトウェア開発環境については今後の課題となっている。

5. ソフトウェアの実装効率化

5.1 ソフトウェアの再利用

ソフトウェアの再利用による開発効率化は、ソフトウェア開発の初期から様々に試みられてきた。

(1) 流用

最も簡単であるが、単純な流用では、派生コードが複数できてしまうなど、版の管理や保守に問題が生じやすい。また、流用を繰り返すことでコードが複雑化し、品質や性能が低下するとともに機能拡張の開発効率が低下することもある。

(2) 共有ライブラリ

複数のプログラムから使われる機能を、ライブラリとしてまとめて提供する。文書処理など共通性の高い分野では有効である。一方、システムごとに要求仕様が異なるビジネスロジックなどではライブラリ化は進んでいない。

(3) フレームワーク

ライブラリでは、全体の流れをソフトウェア開発者が決める。これに対して、全体の流れをフレームワークとして共通化し、システム固有の処理部分だけを開発する手法が出現した。例えば、Webサーバの処理は、クライアント側からのページ要求に対して、HTML (Hyper Text Markup Language) ページを作成して返送することである。そこで、要求の受付と返送の処理をフレームワーク化し、ページ作成部分だけを開発すればよいようにしたのが、Apache Strutsなどである。

(4) ソフトウェアプロダクトライン (SPL)

家電などの組込みソフトウェアの場合、仕様の異なる多くの機種に向けて短期間で開発する必要がある。ここで使われるのが、SPLである。対象分野に合わせてソフトウェアを機能領域に分割し、各領域で異なった仕様を実現する

コンポーネントをあらかじめ用意する。実際の機種開発では、用意されたコンポーネントを組み合わせるか、一部だけを新規に開発してソフトウェアを開発する。

この特集号では、空調機器制御ソフトウェアの事例⁽⁶⁾、などを紹介している。

5.2 ソフトウェアの自動生成

モデルを活用して形式的に仕様を記述することで、ソフトウェア仕様の品質を向上させるだけでなく、コード自体を仕様から自動的に生成し開発を効率化することができる。UMLクラス図からクラスのスケルトンコードを生成するだけでなく、UIのような定型なソフトウェアについては、専用ツールや定義ファイルを用いることで、ほぼプログラミングなしにソフトウェアを開発することができる。この特集号でも、クラス図とシーケンス図を用いて、Webアプリケーションのビジネスロジックを自動的に生成する仕組み⁽⁷⁾を紹介している。

また、分野対応にソフトウェアを自動的に生成する手法も検討しており、複雑な科学技術計算ソフトウェアを、数式エディターを使って開発する事例⁽⁸⁾も紹介する。

将来に向け、Webブラウザの機能拡張と同様に、分野対応のフレームワークとアドオンモジュールとを組み合わせるソフトウェアを構築する手法についても研究を進めている(図2)。この手法では、アドオンモジュールを機器定義データから自動的に作成し、最小限の操作手順だけを簡易言語で指定することで、開発量を大幅に削減できる可能性がある。

6. ソフトウェアの試験支援

ソフトウェア開発で最も時間とコストを要するのは、試験である。図3に、経済産業省が公開した組込み機器開発における工数比率⁽⁹⁾を示す。全工数の半分がシステム設計とソフトウェア開発であり、ソフトウェア開発の3分の2を試験工数が占めていることが分かる。

実装については、既存コードの流用、コードの部品化や仕様からの自動生成などによって、開発量を削減すること

が可能となってきた。一方試験については、一部のコードの変更が全体の動作に影響を与える可能性があるため、機能追加や変更のたびに、すべての試験を再実行する必要がある(回帰試験, Regression Testing)。このため、コードの生成を自動化しただけは、ソフトウェア開発コストを大幅に低減することはできない。

そこで三菱電機では、次のような試験の自動化・効率化にも取り組んでいる。

- (1) システム仕様から試験仕様、試験条件、試験プログラムなどを自動生成する仕組みの構築
- (2) 試験をシナリオに従って実行するとともに、試験結果を自動的に生成する仕組みの構築
- (3) プラントなどの監視制御システムの試験を、実機なしに実行するためのプラント・シミュレータの構築

この特集号では、Webアプリケーションのセキュリティをアプリケーションソース(WebページデータやWebスクリプト)から静的に解析して問題の有無を診断するとともに、典型的な異常リクエストに対する応答を動的に診断する機能を実現した例⁽¹⁰⁾と、システムの再構築にあたって、実システムを流れるデータを試験対象システムに投入して結果データを照合することでシステムの試験を行う仕組み⁽¹¹⁾を紹介している。

7. 開発プロジェクト管理

ソフトウェアの機能拡大と製品投入の早期化によって、ソフトウェア開発プロジェクトは、ますます大規模化・短期化・複雑化している。こうした状況で、高品質なソフトウェアを効率よく開発するためには、開発プロジェクトの状況をタイムリーかつ定量的に把握し、問題に迅速に対処することが大切である。これを実現するためには、ソフトウェア開発プロジェクトの全体を通して、定量的なデータに基づくプロジェクト管理を実施し、PDCA(Plan-Do-Check-Act)のサイクルを回す必要がある(図4)。

開発プロジェクト管理の目的は、ソフトウェアの品質、コスト、納期を計画目標の範囲に収めるとともに、発生す

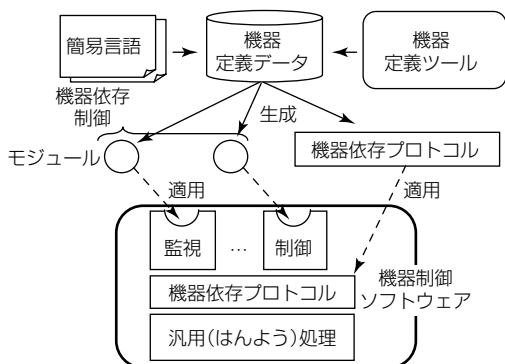


図2. モジュールによる機器への対応, 機能拡張

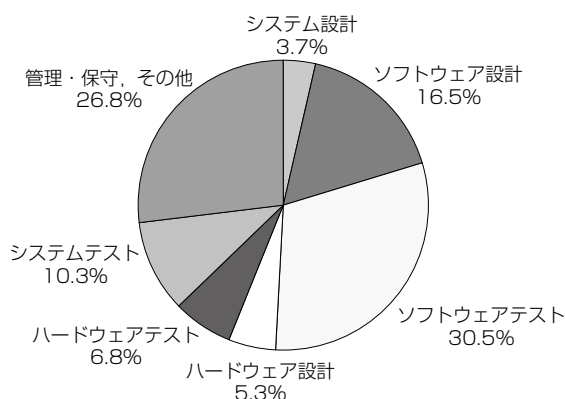


図3. 組込み機器開発における工数比率⁽⁹⁾

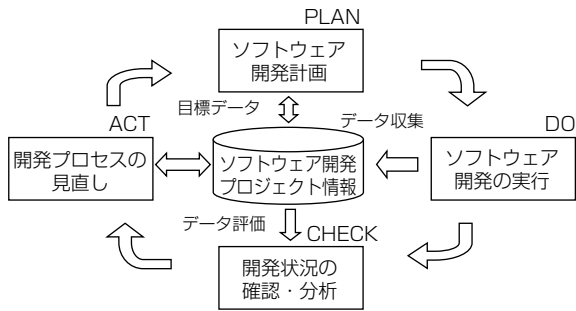


図4. 開発プロジェクト管理のPDCAサイクル

るリスクを最小化することである。このため、プロジェクトの各段階で、仕様書やコードの開発量、レビューの実施状況・指摘件数、試験の密度・網羅率やバグ発生状況などを定量的に収集し、これらのデータを評価・分析することによって、工程の遅延、コストオーバーラン、品質の劣化などをプロジェクトの途中段階で“見える化”することができる。

この特集号では、MDISが開発した、CMMIレベル4相当の定量的プロジェクト管理システム“P-Support”⁽¹²⁾について紹介している。このシステムは、プロジェクト・ポータルから計画・実績データを自動的に取り込み、プロジェクトの品質／コスト／工程／リスクの評価を行う。また評価結果やそのトレンドを、プロジェクトダッシュボード上に“見える化”しており、これによって、管理負荷を増やすことなく、経営者から開発者までプロジェクトの状況を容易に把握し、問題への迅速な対処が可能となっている。

8. む す び

ソフトウェアの高品質化と低コスト化に向けたソフトウェア開発の動向と、三菱電機における取組みについて述べた。組込みソフトウェアから大規模システムに至るまで、ソフトウェア機能の増大と複雑化による開発規模の増加は、今後も継続していくと考えられる。また、これに合わせて、ソフトウェア開発の上流から下流まで新たな手法が導入されていくと考えられる。三菱電機におけるソフトウェア開発量も毎年増加しており、今後もソフトウェア開発の改善に向けて、開発プロセス・手法の改良と開発環境の改善を継続していく。

参 考 文 献

- (1) Carnegie Mellon University Software Engineering Institute：成功するソフトウェア開発(CMMによるガイドライン)，(株)オーム社 (1998)
- (2) Project Management Institute：A Guide to the Project Management Body of Knowledge-Fourth Edition, Project Management Institute (2008)
- (3) 篠崎 衛，ほか：システム構築上流設計手法の改善，三菱電機技報，**84**，No.5，279～282 (2010)
- (4) 上野浩一郎，ほか：高信頼性を実現するシステム仕様検査技術，三菱電機技報，**84**，No.5，295～298 (2010)
- (5) 非機能要求グレード検討会：システム基盤の非機能要求に関する項目一覧 (2010)
<http://www.nttdata.co.jp/nfr-grade/result.html>
- (6) 大河原 繁，ほか：空調機器制御ソフトウェアの再利用開発，三菱電機技報，**84**，No.5，311～314 (2010)
- (7) 原田雅史，ほか：Webシステム開発フレームワーク，三菱電機技報，**84**，No.5，271～274 (2010)
- (8) 井上勝行，ほか：数値計算プログラムのビジュアルな構築環境，三菱電機技報，**84**，No.5，299～302 (2010)
- (9) 経済産業省：2008年版組込みソフトウェア産業実態調査報告書—プロジェクト責任者向け調査—
http://www.meti.go.jp/policy/mono_info_service/joho/downloadfiles/2008software_research/project_hokukokusho.pdf
- (10) 河内清人，ほか：ハイブリッドセキュリティ診断技術，三菱電機技報，**84**，No.5，291～294 (2010)
- (11) 川崎将人，ほか：システム試験の効率化／高精度化技術，三菱電機技報，**84**，No.5，287～290 (2010)
- (12) 坂田賢志，ほか：定量的プロジェクト管理支援システム“P-Support”の開発と試行—定量データに基づくプロジェクト状況の把握への取組み—，三菱電機技報，**84**，No.5，283～286 (2010)