

小島 昌一  
KOJIMA Shoichi

平山 雅之  
HIRAYAMA Masayuki

野口 国雄  
NOGUCHI Kunio

ソフトウェアエンジニアリングという技術分野が誕生して既に四半世紀がたとうとしている。この間、コンピュータの進化、普及とともに、その上で動作するソフトウェアはその利用範囲が急激に広がってきている。特に近年は計算機の普及により、ソフトウェアは極めて身近な技術になりつつある。ここでは、こうしたソフトウェアの普及と急激な変化に対応していくための技術であるソフトウェアエンジニアリングについて、その動向を述べる。また、ソフトウェアエンジニアリングは企業にとって、“より高品質なソフトウェアの効率的な開発”を実現する重要な武器と言える。当社がこの技術を実際のソフトウェア開発の中でどのように位置づけ、どのように活用しているかについてもあわせて述べる。

About a quarter of a century has passed since software engineering was born. During this period, the areas of application of software have been expanding in line with the evolution of computer technologies. Especially in recent years, software has become a familiar technology in people's lives. This paper introduces trends in software engineering as a core strategy for coping with the drastic technological changes taking place. Toshiba considers software engineering to be an important resource for the efficient development of high-quality software. Toshiba's approach to software engineering is also described here.

### ■ 身近になったソフトウェア

#### ■ ソフトウェア利用の拡大

携帯電話、冷蔵庫、自動改札機、道路監視システム、発電所制御システムなど、われわれの身の回りには様々なシステムがあふれている。そしてこれらのシステムは、すべてといっていいほどソフトウェアが搭載され利用されている。このように、計算機の普及とともにソフトウェアの需要は急速に拡大し続けている。図1は、当社の開発製品におけるソフトウェアの相対的な規模を示したものである。この図に見るように、様々なシステムの中でソフトウェアが広く使われるようになってきている。

一方、ソフトウェアが広く社会や生活の中で利用されるに伴い、これらソフトウェアには、より高い品質が求められるようになってきている。また、ソフトウェアを有効活用して、より効率的なビジネスを展開したり、あるいは快適な生活を送りたいとするユーザーの要求を反映して、よりタイムリー

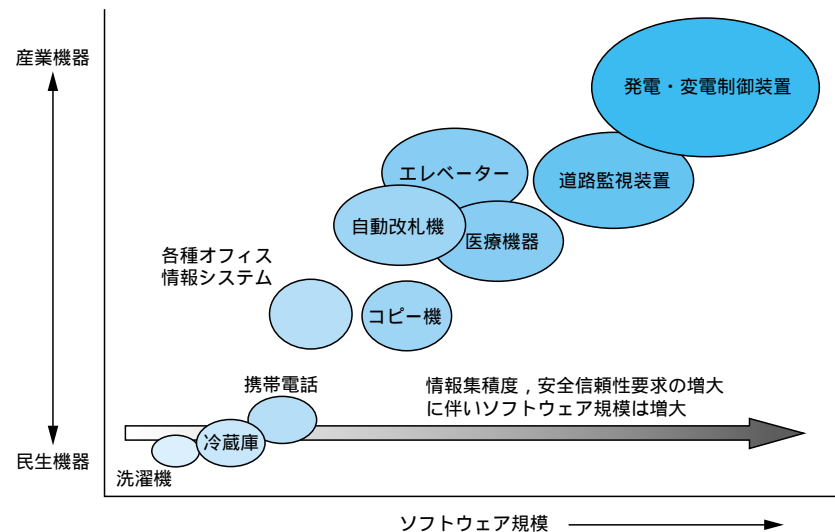


図1. ソフトウェア規模の増大 産業機器、民生機器とも、近年ソフトウェアの規模が増大している。  
Increase in scale of software

な製品リリースが求められている。

#### ■ ソフトウェアの規模増大

ソフトウェア利用の拡大は、ソフトウェア開発に大きな影響を及ぼしている。近年、情報技術の進歩により、従来以上に多くの機能をソフトウェア

で実現することが可能になっている。この結果、ソフトウェア開発規模は増加し、内部構造はより複雑になる傾向にある。一方で、タイムリーな製品リリースを実現するために開発期間はより短縮化している。このように最近のソフトウェア開発では、より多くの

機能、複雑な機能をより短期間に実現することが求められており、ソフトウェアエンジニアリングの成果を有効活用することが重要になってきている。

## ■ ソフトウェアエンジニアリングの扱う技術領域

“ソフトウェアをいかにして作るか”という課題は、ソフトウェアエンジニアリング分野で扱う課題である(図2)。ソフトウェアエンジニアリングは、もともとコンピュータ先進国の米国での研究に端を発しているが、近年では、日本国内でも盛んに研究やその実用化が進められている。

ソフトウェアエンジニアリングでは、実際のソフトウェア開発に関して、次の4点を特に重視している。

- (1) 何を作るか
- (2) どのように要求を形にするか
- (3) どのように品質を保証するか
- (4) どのような手順で開発するか

### ■ 要求獲得・定義技術

“何を作るか”という課題を明確にする技術が要求獲得・定義技術である。顧客がどのような機能をソフトウェアに望んでいるかを正確に把握し、分析・整理する技術である。当社では品質機能展開(QFD: Quality Function Deployment)を応用した要求獲得技術を導入し、顧客要求の具体化に活用している(囲み記事参照)。

### ■ 設計・実装技術

“どのようにして顧客要求を形にして実現するか”という課題を解決するのが設計・実装技術である。近年、この分野ではソフトウェアがかかわる物(データ、周辺ソフトウェア、ハードウェア)をオブジェクトとしてとらえ、これらを中心に設計・実装を進めるオブジェクト指向開発が主流になりつつある。また、ソフトウェアの最適な構

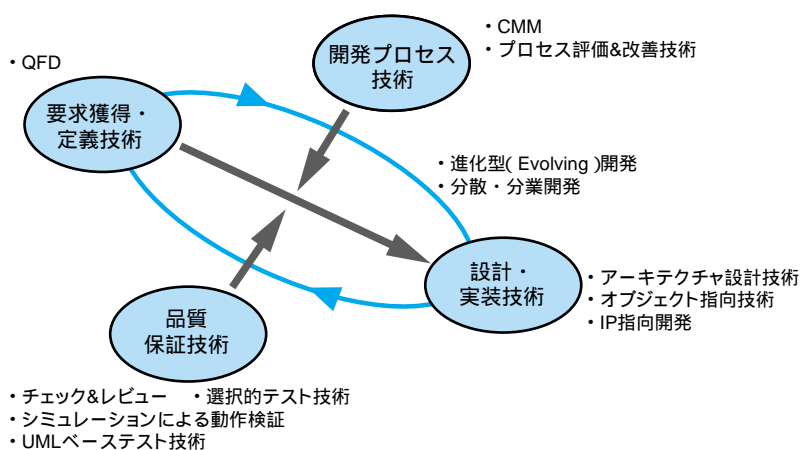


図2. ソフトウェアエンジニアリングの扱う領域 ソフトウェアエンジニアリングでは、要求獲得・定義、品質保証、設計・実装、開発プロセスの各技術領域をカバーしている。  
Technical areas of software engineering

造を決めるアーキテクチャ設計技術も重視されている。当社では、オブジェクト指向の利点を取り入れ、最適なソフトウェアアーキテクチャを実現するための設計手法 ACE(Architecture Centric Engineering)の研究を進めている。

### ■ 品質保証技術

近年、より高品質なソフトウェアが求められる傾向が顕著である。高品質ソフトウェアを実現する技術が品質保証技術である。ソフトウェアの品質は、通常、以下の二つの活動によって確かなものとすることができる。

- (1) 設計・実装過程での徹底した“チェック&レビュー”による検証
- (2) テストによる確実な動作確認

設計・実装過程でのチェック&レビューでは、設計書やソースコードを対象とした静的チェックが中心となる。静的チェックはプログラムに対する文法チェックだけではなく、内部の論理矛盾やデータ関連の整合性チェックなど、より広範囲に確認し、開発早期段階での品質作り込みを実現する。また、テストはソフトウェア品質保証の最後の砦(とりで)として特に重要である。当社では、ユーザーの利用シナリオに着目した重点的なテストを行い、発見された不具合を確実に除

去していく選択的テスト手法を開発し利用している。

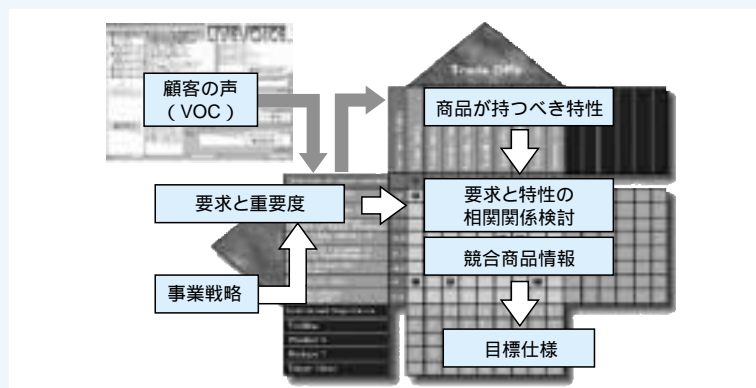
また、最近ではオブジェクト指向開発の普及とともに、設計記述法として UML(Unified Modeling Language)の活用も盛んになっている。これに対応して、UML設計記述に対する設計検証やテストケース作成手法の研究・実用化を進め、いっそうの品質向上を目指している。

### ■ 開発プロセス技術

ソフトウェア開発では要求獲得・定義、設計・実装、品質保証といった技術とともに、開発にかかわる“人”と“組織”を中心とした開発プロセスの視点からの管理も重要である。開発プロセスの管理は、現状の開発のスタイル(手順や作業)がどのようになっているかを把握し、問題箇所を改善するといった、“プロセス評価&改善技術”が中心となる。これについては、米国カーネギーメロン大学ソフトウェア工学研究所(CMU/SEI)で開発された CMM(Capability Maturity Model)の応用による開発プロセスの改善が有効と考えられる。また、開発プロセス管理技術は、個々の開発担当者のモチベーションや個々の組織の組織マインドをも改善し、より前向きな姿勢を引き出すことも目的の一つ

## 要求獲得・定義技術

“何を作るか”を明確にするためには、まず企画活動の原点である顧客の生の声(VOC: Voice Of Customer)を収集し、企業コンセプトを盛り込んで顧客に提示して、要求の重要度評価を行ってもらう。そして重要な要求を実現するために、過不足のない機能特性の目標仕様を設定するというプロセスを進めている。このプロセスでは、まず膨大なVOCから重要かつ潜在的な要求を分類・抽出して、顧客からの要求に対する理解のレベルを高める。次に、品質機能展開(QFD)を活用し、抽出した顧客要求を製品の機能特性に変換し、要求の重要度や顧客満足度、トレードオフ、ベンチマーク分析などを行って、機能特性の目標仕様を明確に



QFD(品質機能展開) 品質機能展開を利用した、顧客要求の仕様への反映

する。この手法は、設定した仕様が顧客要求に与える影響を定量的に評価できるので、VOCからQFD表まで一貫し

た製品企画データベースとして活用できる。

である。

### ソフトウェア開発技術の最前線

近年のソフトウェア開発では、更に様々な方向に技術進化が進んでいる。

#### IP指向開発

ソフトウェアを効率的に作るには、既存のソフトウェア資産を有効に活用するのがもっとも効果的と言われている。IP(Intellectual Property)指向のソフトウェア開発は、ソフトウェアのソースコードだけでなく、様々なノウハウも含めて設計情報をパッケージ化して再利用を加速する考え方である。既に世の中でもJavaBeans<sup>(注1)</sup>やActiveX<sup>(注2)</sup>など、様々な商用コンポーネント(COTs: Commercial Off The shelf)が利用されている。当社では、こうした従来のコンポーネント指向設計をより積極的、広範囲に拡大したIP指向ソフトウェア開発を実現することで、より効率的な開発を目

指している。当社におけるIP指向ソフトウェア開発では、既存ソースコードからIPとして再利用が有望な箇所を検索・抽出する技術や、第三者が提供するIPやCOTsの特性や品質を評価する技術を研究、実用化している。

#### 進化型開発

近年は、まったく新規にソフトウェアを開発することはまれになってきている。多くは既存のソフトウェアに対する機能追加や改良などによって、ソフトウェア開発効率向上を図っている。このようなソフトウェア開発のスタイルは、一般に進化型(Evolving)開発と呼ばれている。また、近年の大規模ソフトウェア開発は、一社だけで開発せずに、システムの部分ごとに別の会社が分散・分業開発するというスタイルも珍しくない。このような進化型開発あるいは分散開発では、過去のソフトウェアや第三者のソ

フトウェアを正しく理解する必要がある。当社ではプログラム内の特定変数に着目し、その使われ方を把握するスライシングという手法を応用して、プログラム理解を円滑に進める技術も開発し利用している。

また、進化型開発や分散開発によるソフトウェアは、テストなどの動作確認の際に、すべての関連ソフトウェアがそろわない、あるいは周辺ハードウェアが一社だけでは用意できないといった課題も生ずる。当社ではシミュレーションを活用して、円滑に動作検証を行う方式を開発し、家電製品や情報機器をテストしている。

更に、進化型開発や分散開発では、開発時の進捗(しんちよく)の把握やバージョン間の整合性チェック、不具合情報の共有化なども大きな課題になると言われている。当社ではWebを活用したバージョン管理、不具合管理システムを活用することで、この課題の克服を図っている。

(注1) Java及びその他のJavaを含む商標は、米国SunMicrosystems社の商標。

(注2) ActiveXは、米国Microsoft Corporationの米国及びその他の国における商標。

## ソフトウェアエンジニアリングへの組織的な取り組み

実際のソフトウェア開発では、設計・実装、品質保証などの技術的側面だけでなく、人的・組織的側面も、品質や生産性に大きな影響を及ぼす。いかに開発者のやる気を引き出し、どのように組織的な開発を実現するかが、ソフトウェアエンジニアリングが抱えるもう一つの課題である。これら二つの側面は、それぞれプロダクトの側面、プロセスの側面ととらえることもできる。

プロセスの側面については、近年、米国ではソフトウェア開発における個人やグループを重視するPSP(Personal Software Process)やTSP(Team Software Process)といった手法の研究が進められている。これらの研究は、ソフトウェア開発の本質が個々の開発者のスキルや組織マインドであることをとらえたものと言える。

当社でも、ソフトウェア開発における人と組織を重視しつつ、ソフトウェア開発をプロダクトとプロセスの二つの側面からとらえ、組織的な改善の推進を図っている。当社では、小さいものでは家電製品に組み込まれるソフトウェアから、大きなものでは発電所制御システムまで、様々なソフトウェア開発を行っている。こうしたソフトウェアは分野ごとに様々な特性を

持ち合わせているが、ソフトウェア開発を効率化する手法・ツールの導入、組織力の向上やソフトウェア開発の可視化などは、全社共通課題として位置づけ、施策を推進している(図3)。

### ・開発手法・ツール導入

ソフトウェア開発に関しては、これまでのソフトウェア工学の研究によって様々な手法やツールが提案されている。当社では、これらの手法やツールに当社のオリジナル技術を融合し最適化して、最大の効果が得られるように工夫している。また、これらの技術導入にあたっては、当社全体のソフトウェア設計力向上を目標に、全社施策として推進している。

### ・組織力の向上

組織力の向上については、前述のCMMをコア技術として採用し、ソフトウェア開発組織の診断と改善を進めている。プロセス評価で明らかになった個別の課題は、ソフトウェアエンジニアリングで提供される要求獲得・定義、設計・実装、品質保証の各技術を活用し、その解決を図っている。

### ・ソフトウェア開発の可視化

ソフトウェア開発では、通常、設計過程の様々なドキュメントやソースコードが作業のアウトプットとなる。しかし、これらは必ずしもわかりやすいものではなく、ソフトウェア開発を見通しの悪いものになっている。当社で

はデータドリブンの考え方に基づき、統計手法を駆使した経営変革運動を全社的に推進している。ソフトウェア開発においても、様々な定量化指標を用いて、ソフトウェアの開発状況の可視化と把握を進めている。

## より高度なソフトウェア開発を目指して

ソフトウェアエンジニアリングの研究は日進月歩で進んでいる。当社では、これらの技術を活用して、より高度なソフトウェア開発を実現し、顧客のニーズにこたえていきたい。

なお、この特集の以下3編の論文では、ソフトウェアの設計・実装、品質向上、開発プロセスに関する当社の具体的な取り組みを紹介している。



小島 昌一  
KOJIMA Shoichi

研究開発センター システム技術ラボラトリー室長。  
システム・ソフトウェア技術の研究・開発に従事。IEEE、情報処理学会会員。  
System Engineering Lab.



平山 雅之  
HIRAYAMA Masayuki

研究開発センター システム技術ラボラトリー主任研究員。  
ソフトウェアの信頼性保証、テスト・検証技術の研究に従事。情報処理学会会員。  
System Engineering Lab.



野口 国雄  
NOGUCHI Kunio

研究開発センター システム技術ラボラトリー主任研究員。  
製品上流設計技術の研究に従事。情報処理学会会員。  
System Engineering Lab.

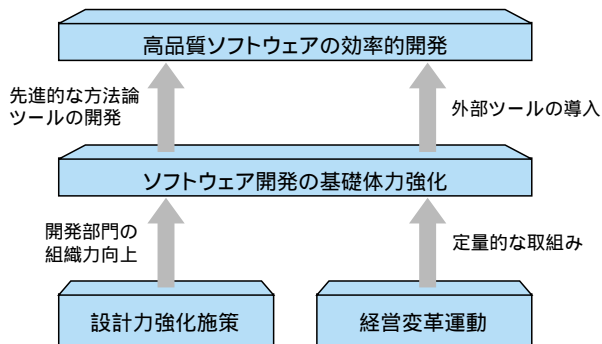


図3 ソフトウェアエンジニアリングへの組織的取り組み 設計力強化と経営変革運動により、当社全体のソフトウェア開発の基礎体力強化を図っている。

Toshiba's approach to software engineering