ソフトウェア開発の高度化を支えるソフトウェア工学

Software Engineering Supporting Advanced Software Development

江口　和俊
■ EGUCHI Kazutoshi

社会インフラシステムやデジタル機器へのニーズの多様化と高機能化に伴い，ソフトウェアで実現する機能の多様化やソフトウェアの大規模・複雑化が急速に進んでいる。また，ソフトウェア適用領域の拡大により高信頼化の要請も強まっている。

これらに対応するために，企業のソフトウェア開発活動では，事業戦略とソフトウェア開発戦略を密接に連携させるプラットフォーム化技術，上流開発工程での品質向上技術，ソフトウェア開発プロセスの体系化をはじめとして，ソフトウェア工学（Software Engineering）の成果を積極的に適用している。

Requirements have become diversified and functions increasingly sophisticated in the fields of social infrastructure systems and digital equipment in recent years. Both the diversity of functions realized by software and the size and complexity of software have been rapidly growing as a result. At the same time, there is increasing demand for high reliability of software as the areas of application expand.

In response to these needs, Toshiba has been applying the latest accomplishments in software engineering, such as platform technology that closely aligns software development strategy with business strategy, quality improvement technologies at earlier stages of development, and organization of software development processes, to commercial software development activities.

ソフトウェア開発高度化の要請とソフトウェア工学

社会基盤を支える電力システム，交通システムなどの社会インフラシステムをはじめ，企業活動に欠かせない情報システムや，日常生活で接するテレビ，冷蔵庫，洗濯機などの家電製品，携帯電話，自動車といった多くのものにソフトウェアが搭載されている。一方，それらシステムやデジタル機器の高機能化に伴いソフトウェアは急激に大規模・複雑化している。また，ソフトウェアの適用領域の拡大や機能の高度化によりソフトウェアの高信頼化の要請も強く，更に，ユーザーニーズの多様化に伴い，システムや機器のバリエーションの増加や開発期間の短縮の傾向も顕著となっている。

高機能で大規模なソフトウェアを信頼性高く短期間で開発するためには，一部の優秀な技術者の技量や経験に頼る，いわば職人芸的な開発手法ではなく，ソフトウェアの開発技術，開発管理技術や開発プロセスを体系化したソフトウェア工学の成果を積極的に活用する必要がある。

ソフトウェア工学ということばは，1960年代後半に登場し，職人芸的なソフトウェアの開発手法から工学的な開発手法へ転換するための研究が行われるようになった。ソフトウェアの質的及び量的変化や品質要求の高まりなどを背景に，ソフトウェア工学の成果が，高い信頼性が要求される大規模な基幹システム向けソフトウェア開発をはじめとして，携帯電話や家電製品などに搭載される組込みソフトウェア開発にも適用され，高機能で大規模なソフトウェアを信頼性高く短期間で開発するための開発技術，開発管理技術，及びソフトウェア開発プロセスの高度化に貢献している。

近年，ソフトウェアの開発技術及び開発管理技術の領域では，事業戦略とソフトウェア開発戦略を密接に連携させるプラットフォーム化，ソフトウェアプロダクトライン開発技術，及び高信頼ソフトウェア開発に対応した上流開発工程での品質向上技術が注目されている。また，ソフトウェア開発プロセスに関する技術の体系化も進んでいる。

ソフトウェアの開発効率を大幅に向上させるソフトウェアプロダクトライン型開発

■ソフトウェア開発技術の変換

ソフトウェアの大規模・複雑化に並行して，ソフトウェア開発の効率化技術が研究されてきた。1960年代には，順次制御，繰返し制御や条件分岐を用い，階層的なモジュール構造とする構造化プログラミングが提唱された。1970年代には，データとそのデータを処理する手続きを一つのまとまり（オブジェクト）とし，オブジェクトの組合せでソフトウェアを開発するオブジェクト指向技術が研究され，1980年代以降，C++やJava™(注1)といったオブジェクト指向技術に基づくプログラミング言語が普及して多くのソフトウェア開発に活用されている。更に，1990年代に入ると，オブジェクト指

（注1）　Javaは，米国Sun Microsystems, Inc.の米国及びその他の国における商標又は登録商標。

向を発展させ，ソフトウェアの構成単位であるコンポーネントを要求分析から実装まで一貫して適用する，コンポーネントベース開発方法論も提案され研究が行われている。

一方，ある特定の機器やシステムの機能拡張計画及びラインアップ計画などの製品戦略を前提として，再利用可能なコア ソフトウェア資産をベースとした組織的開発方法論も研究されており，1990年代後半に，米国のカーネギーメロン大学ソフトウェア工学研究所（SEI：Software Engineering Institute）が“ソフトウェアプロダクトライン”として体系化している。

■ソフトウェアプロダクトライン型開発

ソフトウェアプロダクトライン型開発方法論では，その製品ファミリーで共通資産となるソフトウェアアーキテクチャ資産，プログラム資産やドキュメント資産などの“コアアセット開発”活動，個々の製品仕様に対応してコアアセットをベースに製品又はシステムを開発する“プロダクト開発”活動，及びコアアセット開発やプロダクト開発を支える技術面や組織面の“管理”活動の三つを基本としている。東芝も，携帯電話やデジタルテレビなど多くの製品開発にこのソフトウェアプロダクトラインの概念を適用している。

また，ソフトウェアプロダクトラインが体系化されるはるか以前の1960年代後半には，同様な概念である“ソフトウェアファクトリ”が提唱され，特に日本企業が積極的に展開した。当社も，この組織的開発方法論を1970年代から大規模制御システムに適用し，多くのシステムを出荷している。

それらの実績から，2008年にアイルランドで開催されたSEI主催の第12回ソフトウェアプロダクトライン国際会議において，当社の発電所向け監視制御システムが日本企業ではじめて“ソフトウェアプロダクトラインの殿堂（Software Product Line Hall of Fame）”に登録されている。

■ソフトウェアプラットフォームの開発

ソフトウェアプロダクトライン型開発アプローチでは，プロダクトラインの生産性や品質に大きな影響を与えるコアアセットいわゆるソフトウェアプラットフォームの開発が特に重要である。ソフトウェアプラットフォーム開発の基本的な流れは，次のとおりである。

（1） プロダクトラインとして扱う製品計画を策定する。

（2） その製品計画を重要なインプットとして，ソフトウェアプラットフォームに対する要求分析を行う。

（3） ソフトウェアプラットフォームを構成するソフトウェアプラットフォーム アーキテクチャや，ソフトウェアコンポーネント，テスト戦略及びテストケース，関連ドキュメントなどを開発する。

要求分析フェーズでは，特に製品計画を前提とした共通部分と個別部分を明確にした要求抽出や，将来を見据えた保守性などの非機能要求の抽出が重要となる。ソフトウェアプロダクトライン型アプローチを取らずに開発された既存製品をベースに，ソフトウェアを再構築しプラットフォーム化を行う場合も多い。そのような場合には，既存のソフトウェア資産をベースに要求分析や構造解析を行ったうえで，今後の製品計画をベースとした要求仕様定義，ソフトウェア構造再構築，及びソフトウェアプラットフォーム開発を行う。この際重要なのは，既存ソフトウェア開発時の意図を理解するとともに，既存のソフトウェア構造に惑わされない要求定義と構造再設計を行うことである。

一方で，プラットフォームの優劣を判断する定量的な尺度が現在明確ではなく，プラットフォーム開発時のプラットフォーム品質評価技術と，プラットフォームをベースに製品開発を重ねた際のプラットフォーム劣化度評価技術の確立が今後の課題と考えている。

ソフトウェア品質向上のための開発技術

■ソフトウェア品質の作り込み

ソフトウェアの適用範囲拡大や役割変化などからくる品質要求の高まりに対応して，ソフトウェア品質を作り込むための技術研究が進んでいる。

ソフトウェア品質を確保する技術を大別すると，ソフトウェア開発の上流工程で行うレビューや仕様検証，静的解析など，ソフトウェアの誤り混入を予防するための技術と，ソフトウェアを実装した後で製品に内在したバグを検出し取り除くための動的テスト技術がある。

ソフトウェア品質向上のための基本的な考え方は，現在ではW字モデル（**囲み記事参照**）を基本として，開発の上流段階からテスト戦略立案，テスト設計，及びテストケース作成を行うテストプロセスが広がりを見せている。また，下流工程のテストの効率化やテスト網羅度の向上のために，直交表(注2)などを活用し組合せテストケース数の過大な増加を抑えながら組合せ網羅度を高める技術研究も進んでいる。テストを行う前の品質確保技術としては，プログラミング後のソースコードを動作させずに問題点を抽出する静的解析技術(注3)が，ツールの高度化に伴い既に普及段階にきている。

一方，仕様記述言語による仕様記述やモデル検証などにより開発の初期段階で誤りを発見し，品質の確保や開発コストの減少を狙ったフォーマルメソッドは，1970年代から盛んに研究されて

（注2） **直交表**
任意の2因子について，その水準のすべての組合せが同数回ずつ現れるという性質を持つ割付け表。

（注3） **静的解析技術**
実際にプログラムを動かすことなく，ソフトウェアの設計図にあたるソースコードを静的に解析して，品質上の問題点となる箇所を指摘する技術。

ソフトウェア開発の流れとV字モデル及びW字モデル

ソフトウェア開発の基本的な流れは，次のとおりである。

(1) ソフトウェアに対する要求を分析し，要求仕様としてまとめる。

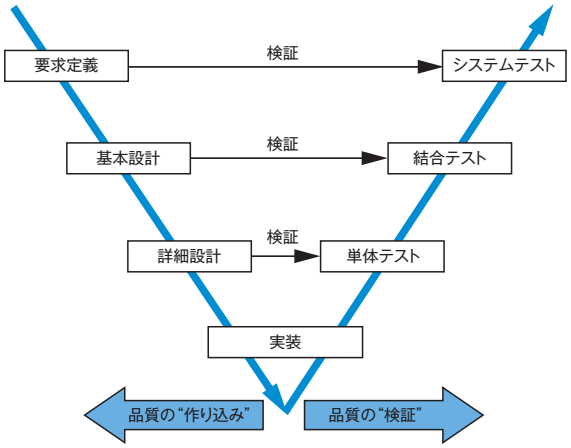
(2) 要求仕様を実現する最適なソフトウェアの構造，処理方法を設計する。

(3) 最適なプログラミング言語でプログラムを作成する。

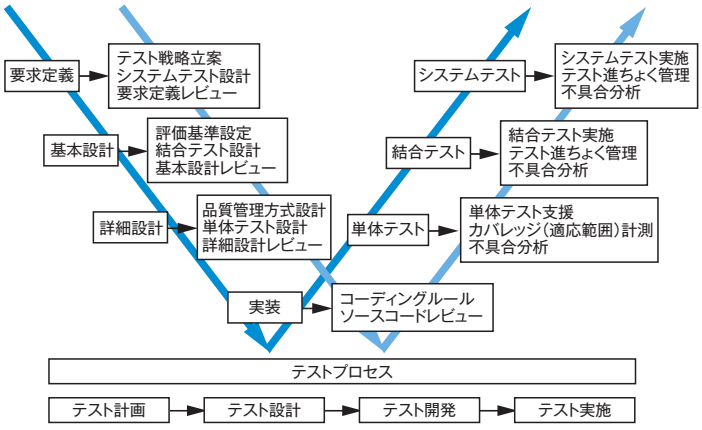
(4) 正しくプログラミングされていることをテストする。

この流れを，プログラムを作成するコーディングフェーズを境として，上流工程と下流工程を左右に対称的に配したものをV字モデルと呼ぶ(**図A**)。V字モデルでは，上流の要求定義，基本設計，詳細設計と，そこで意図したことが正しく実現していることをそれぞれ確認する下流のシステムテスト，結合テスト，単体テストとが対比されている。また，V字モデルを発展させ，テスト戦略立案やテスト設計をV字の左側，つまり，ソフトウェア開発の上流工程で並行して行うことにより，ソフトウェア品質をより高める考え方が提案され，V字を二つずらして並べた形態からW字モデルと呼ばれている(**図B**)。

ソフトウェア開発の流れは，ソフトウェア開発プロセスと呼ばれ，実際には，これらのモデルどおり上流から下流まで順次開発を進めていくウォータフォール型や，主要な部分から全体に徐々に開発範囲を広げていくインクリメンタル型など，開発対象や開発組織に適したソフトウェア開発プロセスが使われている。



図A　V字モデル



図B　W字モデル

きたが，現在では，そのなかでも特にモデル検査技術への関心が高まっている。

■モデル検査による品質向上

当社は，ソフトウェア開発の仕様設計段階で誤り混入を防止し品質を向上させる技術として，モデル検査技術に着目している。モデル検査ツールは，既にいくつか公開されているが，当社はSPIN(注4)を実際のソフトウェア開発に適用している。モデル検査の一般的なプロセスの基本形は次のとおりで，問題点がなくなるまで繰り返す。

(1) 仕様設計段階で作成する状態遷移モデルをベースに，専用の言語で検査用のモデルを記述する。

(2) 検査条件を検査式として作成する。

(3) 検査モデルに対し網羅的な検査を行う。

(4) モデル検査ツールで検出した問題点を解析する。

(5) 状態遷移モデルの誤りを修正し検査を行う。

並列動作を行う産業用機器にSPINを適用した事例では，通常のテストではテスト条件が作りにくいケースや，技術者が気づきにくく，また，まれにしか発生しない問題点なども検出できており，開発に大きな効果があると判断している。今後の課題としては，状態数が大きくなりすぎて現実的に検査できなくなる状態爆発を回避する技術の高度化や，リアルタイム性を必要とするシステム検証への対応方法が挙げられる。

一方，システムの時間制約をモデル検査ツールのUPPAAL(注5)で検証する試行も行っている。また，仕様設計段階ではなく，プログラミング後のソースコードを対象とした“モダンモデル検証”技術も，活用の可能性がある技術として注目している。

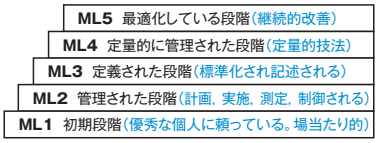
(注4) SPIN
米国AT&T Bell研究所で開発されたモデル検査用ツール。

(注5) UPPAAL
スウェーデンのUppsala大学とデンマークのAalbog大学で開発された形式手法によるモデリングと，そのモデル検査を行うツール。

ソフトウェア開発プロセスの構築による組織力強化

■ソフトウェア開発の組織力強化手法

ソフトウェア開発プロセスに関する技術の体系化については，SEIが1993年に提供を開始したSW-CMM®（Capa-



ML：Maturity Level(成熟度レベル)

図1．CMMIの成熟度レベル ― ソフトウェア開発組織が，場当たり的な開発プロセスから，成熟した規律ある開発プロセスに進化するためのロードマップを示したモデルである。

Maturity levels in Capability Maturity Model Integration (CMMI)

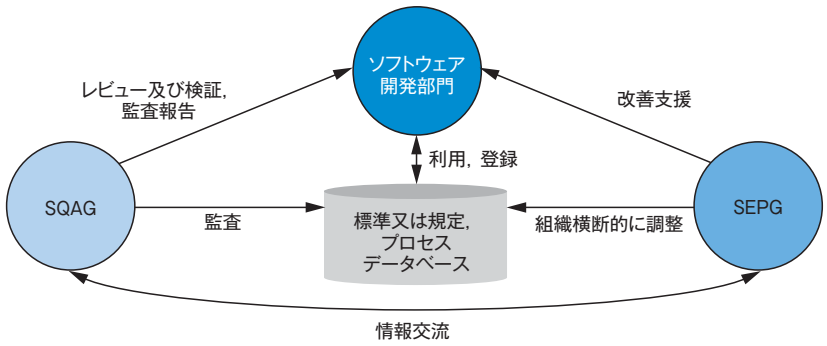


図2．SEPGとSQAGの連携 ― ソフトウェア開発部門，SQAG，SEPGがそれぞれの役割を果たすとともに，それらの連携と調整が重要である。

Relationships between Software Engineering Process Group (SEPG) and Software Quality Assurance Group (SQAG)

bility Maturity Model for Software：ソフトウェア能力成熟度モデル)(注6)が代表的である。これは，ソフトウェアの品質確保や開発効率向上にはソフトウェア開発組織の組織力強化が欠かせないことから，ソフトウェア開発の進め方や管理の仕方をプロセスとして明確に定義し，その実行度合いなどから組織を“成熟度”の視点でとらえ，組織の継続的な改善と強化を狙ったものである。

2002年には，ソフトウェア開発やシステム開発，製品開発などのプロセス能力成熟度モデルを統合したCMMI®（CMM® Integration）(注7)が提案され，2006年には，SW-CMM®のサポートを終了しCMMI®に移行している（**図1**）。

近年，この技術領域では，組織レベルのプロセスモデルからチームや個人のプロセスモデルへの拡張がなされ，また，企業活動の異なった視点で使用している国際標準化機構（ISO)が制定している標準，シックスシグマやCMMI®といった複数モデルの融合も検討されている。

■ソフトウェア開発プロセスの高度化と定量的管理

東芝グループは，1994年から，CMM®を参照モデルとして採用したソフトウェア開発プロセス改善の試行を開始し，2000年から本格的にSPI（Software Process Improvement：ソフトウェアプロセス改善）活動の展開を開始した。コーポレート，カンパニー，及び部門の3階層構造の推進体制を基本とし，ソフトウェア開発グループといっしょに組織力強化の視点で活動を推進するSEPG（Software Engineering Process Group），及び開発プロセス品質の視点で活動を推進するSQAG（Software Quality Assurance Group)が連携してSPI活動を行っている（**図2**）。

現在では，より高成熟度の部門を拡大するため，特にソフトウェア開発に関連した定量的管理データの収集及び分析の体系化と高度化に取り組んでいる。また，改善の方向性を示し，組織力強化を効果的に推進するため，組織プロセスの成熟度を診断するSEI公認アプレイザーの養成も行っている。

ソフトウェア開発技術の高度化と産学連携

ソフトウェア開発現場に種々のソフトウェア工学の成果が取り入れられてきている。ソフトウェア工学の成果は，例えばオブジェクト指向やソフトウェアプロダクトラインといった開発方法論の形で具現化し，また，数理論理学を基盤とした形式検証手法とそれを実問題に適用するためのツールという形で提供され，企業活動に生かされている。われわれの社会活動の非常に広い範囲でソフトウェアの重要性が更に増し続けていき，それに対応して，ソフトウェアはますます高機能・大規模化し，ソフトウェアに対する品質要求もいっそう高まっていくことは論をまたない。そのような状況で，ソフトウェア開発の課題解決には，ソフトウェア工学的なアプローチをより積極的に取り入れていく必要がある。

一方，ソフトウェア工学の成果としての開発方法論や最先端ツールを十分に活用するためには，ソフトウェア科学に関する知識も重要である。当社は，ソフトウェア開発にソフトウェア工学的アプローチを積極的に取り入れていくとともに，ソフトウェア工学の高度化のため，また，ソフトウェア科学の知識を持ちソフトウェア工学の成果を企業活動に十分活用できる人材を育成するため，ソフトウェア関連の研究と教育の面で，産学連携を強めていきたいと考えている。

(注6)，(注7)　CMM，CMMIは，米国カーネギーメロン大学の登録商標。

江口　和俊
EGUCHI Kazutoshi
ソフトウェア技術センター所長。コンピュータシステム，及び組込みソフトウェアの研究・開発に従事。IEEE，情報処理学会会員。
Corporate Software Engineering Center