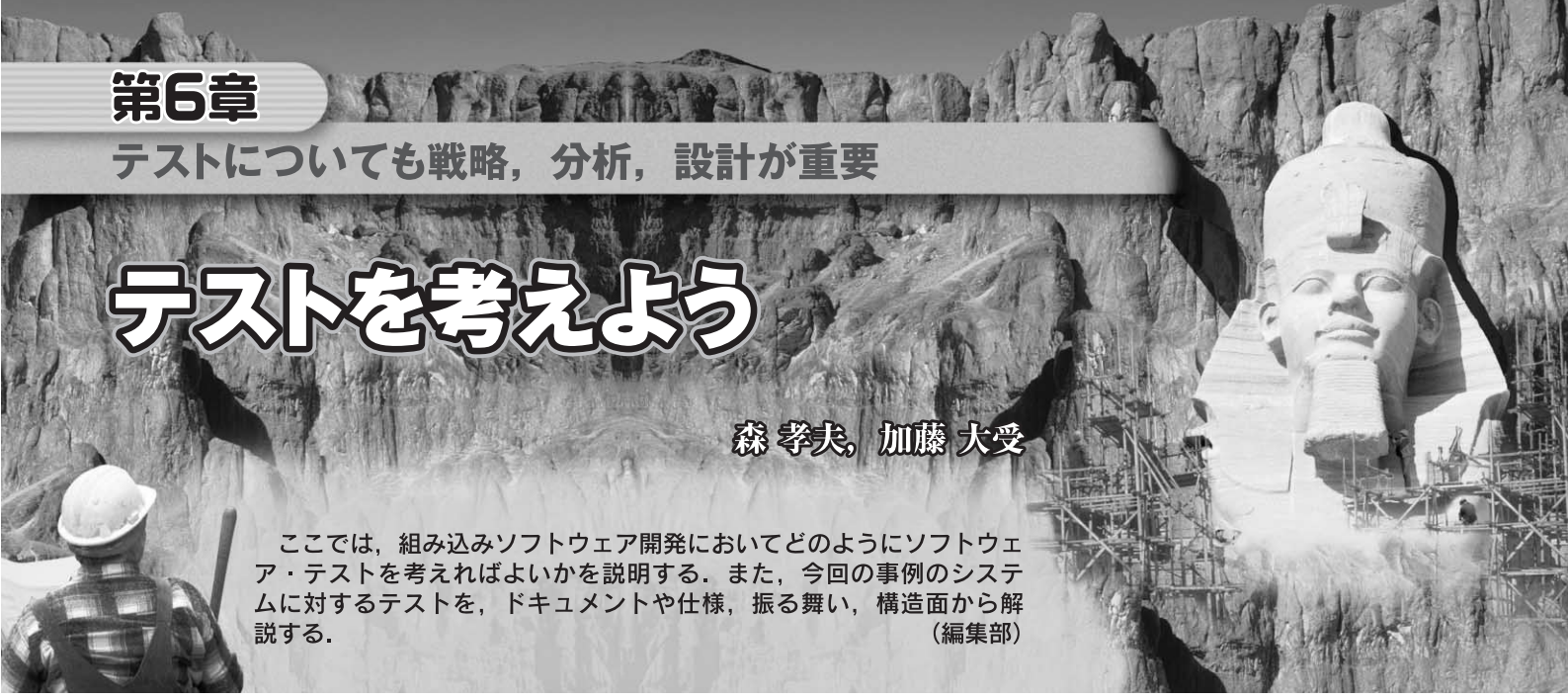

第6章

テストについても戦略，分析，設計が重要

テストを考えよう

森 孝夫，加藤 大受

ここでは，組み込みソフトウェア開発においてどのようにソフトウェア・テストを考えればよいかを説明する．また，今回の事例のシステムに対するテストを，ドキュメントや仕様，振る舞い，構造面から解説する．（編集部）

1. 組み込みソフトウェア開発におけるテストとは

「テスト」というと物が完成してから行うものだと考えてしまいがちですが，ソフトウェア開発の場合は，もっと上流の工程からテストを考え，実施していくことが求められています．ここでは「V字モデル」と呼ばれるプロセス・モデルを見ながら，話を進めることにしましょう．

組み込みシステム開発におけるプロセス・モデルとしては，「SLPC-JCF98」[注1]にも定義されているV字モデルが一般的です．このプロセス・モデルのうち，ソフトウェアをシステムにおける一つのパーツとして捉えることで，二つのV字モデルに分解することができます（**図1**）．つまり，下位V字モデルでソフトウェアを開発しておき，システムに統合して上位V字モデルで全体をテストする，という考え方ができるのです．

上位V字モデルにおけるテスト（システム結合，システム適格性確認テスト）では，試作ハードウェアや実機を利用して，システム全体の動作やシステムそのものが要求に合っているかなどをテストします．これに対して下位V字モデルにおけるテスト（ソフトウェア・テスト）には，パソコン上の開発環境や静的解析ツールなどを利用して行うテスト（単体テストなど）や，シミュレータや試作ハードウェアを使うテストなどがあります．こうして，さまざまな角度からテストを実施し，開発したソフトウェアの動作に問題がないか，要求に合っているかなどを確認します．

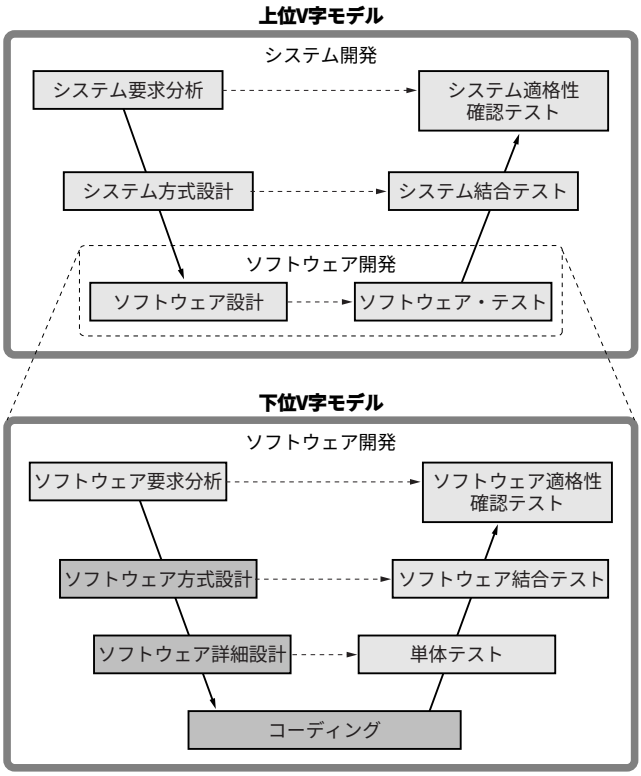

図1　上位下位V字モデル
一般的なシステム開発のV字モデルからソフトウェア開発の部分を取り出し，別のV字モデル（下位V字モデル）としてとらえたもの．このモデルによって，評価や検証における作業事項について，実機以前に行うべき事項と実機で行うべき事項を切り分けてテストを計画・設計するべきであることを明確に認識できる．なお「適格性テスト」とは，全体を統合して，要求に対して適格性のある物ができているかをテストすること（いわゆる「統合テスト」のこと）である．

注1：SLPC-JCF（Software Life Cycle Process - Japan Common Frame）98とは，情報処理振興事業協会が定めた「ソフトウェアを中心としたシステム開発および取引のための共通フレーム 1998年版」のことを指す．「共通フレーム98」とも呼ばれる．

KEYWORD ── V字モデル，上位下位V字モデル，Boehmの法則，積み上げ型の検証，テスト計画，テスト戦略，テスト・アプローチ，テスト分析，テストの狙いどころ，テスト単位，テスト観点，テスト設計，テスト・ケース

● 不具合は前倒しで発見した方が得

ソフトウェア工学の世界では，「不具合発見が後になればなるほど修正に掛かるコストが増大していく」という法則があります（Boehm の法則）[1]．つまり，下位 V 字モデルでソフトウェアをしっかりテストしておき，システム結合以降のテストに入るまでにソフトウェアの不具合を除去しておいた方が，全体のコストが少なくて済むのです．特に組み込みシステムの場合，システム結合後に不具合が出ると原因究明に苦労し，修正コストの増大とスケジュールの遅延を引き起こす可能性が高くなります．そういった意味で，できる限りソフトウェアそのものの品質を上げた上で，システム結合以降の工程に入った方がよいでしょう．

この話は，下位 V 字モデルの工程についても当てはまります．単体テストでつぶせる不具合はつぶしておき，ソフトウェア結合テスト，ソフトウェア適格性確認テストに進んだ方がコストが少なくて済みます．

一般に，組み込みシステムや組み込みソフトウェアには，高い信頼性が求められます（ハードウェアの各部品に高い信頼性が求められるのと同じように…）．この要求を満たすためには，結合してから全体のテストを行うのではなく，部品である一つ一つのコンポーネントの信頼性を高め，これらの部品を積み上げてからテストを進める「積み上げ型の検証」を行うのが望ましいでしょう．

「積み上げ型の検証」が効果を上げるか否かは，システムの構成要素である各部品に対して，きちんと要求を定義できているかにかかってきます．例えば，下位 V 字モデルのテストの効果を上げるためには，システム方式設計の段階で，ソフトウェアに対する要求を確立することが重要です．そこがぶれてしまっては，テストの効果が下がってしまうからです．同じように，単体テストの効果を高めるためには，ソフトウェア設計時に，ソフトウェア部品に対する要求を明確に確立していく必要があります．

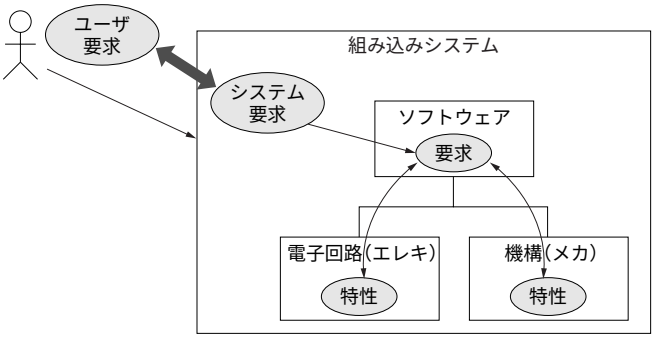

図2　組み込みシステムとソフトウェア特性
組み込みシステムにおいて，機構（メカ）と電子回路（エレキ）とソフトウェアは密接に関連している．そのため，ユーザの要求（システム要求）が変わらなくても，機構や電子回路に変更があるとソフトウェアも変更せざるを得ない．

● なぜ組み込みソフトウェアのテストは難しいのか

組み込みソフトウェアに対する要求は，確立しやすい場合としにくい場合があります．その要因は，顧客や企画レベルの要求が変わりやすいから，という理由だけではありません．

組み込みシステムの場合，組み込みシステム全体に対する要求は同一であったとしても，ハードウェアの構成や特性が変われば，ソフトウェアに対して求められる要求が変わってしまいます．なぜなら，ハードウェアの構成や特性が異なると，必然的にハードウェアの動きが変わってしまうため，それに合わせてソフトウェアも動きを変えざるを得ないからです（**図2**）．

ハードウェアの設計が本質的に難しく，何度も試行錯誤を繰り返さないと完了しないようなシステムや，ハードウェアの特性にソフトウェアが強く依存するようなシステムの場合には，ソフトウェアの要求を確立するのが困難になります（なお，この問題に対して取りうる一つの答えを，右掲の**コラム1**に示す）．

2. ソフトウェア・テストの進め方

ここまで，テストが重要なこと，そして不具合を前倒しで防いでいくことが全体の効率を向上させることを説明しました．では，ソフトウェアのテストは具体的にどのような手順で進めていけばよいのでしょうか．

● まずはテスト計画書を作ろう

テストはソフトウェア開発と同じように，計画を立てて実施していきます．具体的には，**図3**のようなことについて計画を立てます．

テストの計画をまとめたものを「テスト計画書」といいます．テスト計画やテスト設計仕様などといった，ソフトウェア・テストに関するドキュメントについて規定した標準（テンプレート）として，IEEE 829-1998 があります．この標準は，以下のような項目について規定しています．

1) テスト計画書識別番号