

2 ソフトウェア開発技術

ソフトウェア実装プロセスは、取得者と供給者にソフトウェア開発の“共通のものさし”として提供する作業項目を定義したものである。この作業を実際に行うために使用されるのが、ソフトウェア開発技術である。

2-1 ソフトウェア開発手法

2-1-1 ソフトウェア開発モデル

ソフトウェア開発モデルとは、ソフトウェアを開発するための定型的な手順を表現する標準モデルである（システムを開発するためのシステム開発モデルとも呼ばれる）。

【ソフトウェア開発モデルのメリット】

- ・ 開発作業が定型的なもので、開発経験が浅くても作業に参加できる。
- ・ 成果物が定型化されやすいので、保守作業を軽減できる。

(1) ウォータフォールモデル

ウォータフォールモデルは、大規模で複雑な開発工程を複数の段階（フェーズ）に分割し、段階ごとに開発を進めていく手法である。ウォータフォール（滝）という名前が示すとおり、上流工程から下流工程へと、後戻りせずに開発を進めていく手法である。

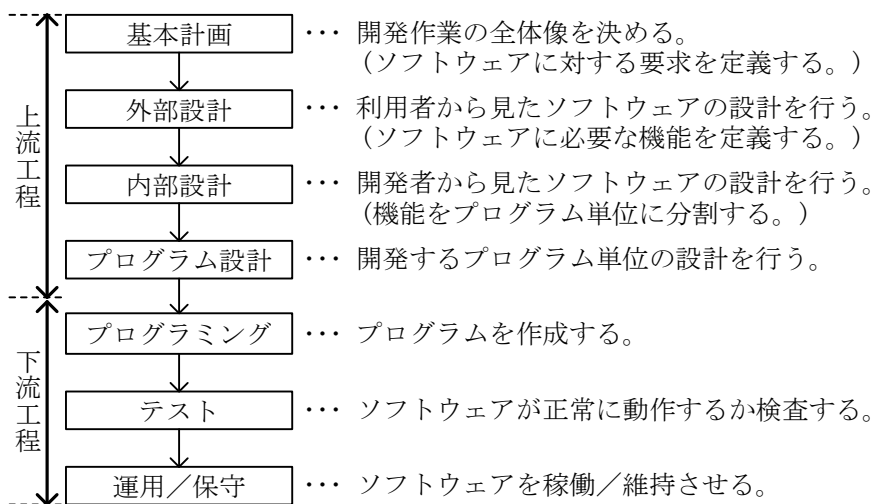


図 4-5 ウォータフォールモデル

ウォーターフォールモデルの開発工程で実施する主な作業内容，及びSLCPとの大まかな対応は，次のようになる。

開発工程名	主な作業内容	SLCP 対応
基本計画	システム化計画，要求定義	ソフトウェア要件定義プロセス
外部設計	サブシステムの定義，入出力概要設計，コード設計，論理データ設計	ソフトウェア方式設計プロセス
内部設計	機能分割／詳細化，入出力詳細設計，物理データ設計	
プログラム設計	モジュール分割	ソフトウェア詳細設計プロセス
プログラミング	コーディング	ソフトウェア構築プロセス

ウォーターフォールモデルによる開発では，ソフトウェアの全体像から徐々に詳細な設計へと進めていく**トップダウンアプローチ**が利用される。一方，開発したソフトウェアのテスト工程では，詳細レベルからソフトウェア全体へとテストを進めていく**ボトムアップアプローチ**が利用される。したがって，設計段階と各テストの関係を図示すると，次のようなV字型になる（これを**V字構造モデル**という）。

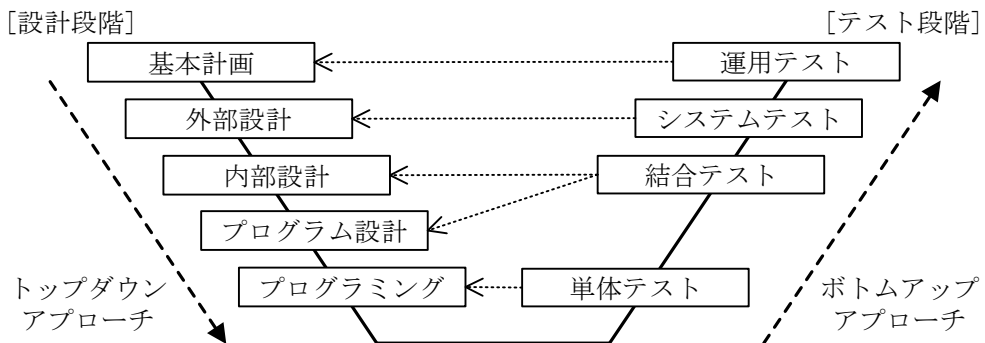


図 4-6 ウォーターフォールモデルの全体像（V字構造モデル）

テスト名	テスト内容	SLCP 対応
単体テスト	一つひとつのモジュールが正しく動作することを確認する。	ソフトウェアユニットテスト
結合テスト	モジュールを組み合わせるとして，正しく動作することを確認する。	ソフトウェア結合テスト
システムテスト	ソフトウェア全体に対して，要求された機能を満たしているか，操作性や性能に問題はないかなどを確認する。	ソフトウェア適格性確認テスト
運用テスト (受入れテスト，承認テスト)	利用者であるユーザ部門が，実際の運用と同じ条件でソフトウェアを動かす，要求を満たしているかどうかを確認する。	受入れテスト 又は運用テスト

(2) プロトタイプモデル（プロトタイピング）

プロトタイプモデルは、短期間で暫定的に試作品（プロトタイプ）を作成し、利用者に試用／評価してもらい、修正を繰り返しながら仕様を確定する開発手法である。

開発工程の早い時期に利用者の要求に合致しているのかを確認できるので、手戻りを少なくすることができる。しかし、スケジュールの調整が難しく、完全とはいえなくても試作品を作成しなければならないことから、小規模システムの開発に向いている。

最終的には、仕様確認後に試作品を廃棄して新規にシステムを作成するか、作成したプロトタイプを改良（機能追加）して本システムとするか、の2通りの方法がある。

(3) スパイラルモデル

スパイラルモデルは、大規模なシステムを独立した部分単位に分割して、部分ごとに設計／開発／テストの工程を反復しながら、完成度を高めていく開発手法である。ウォーターフォールモデルとプロトタイピングを合わせた手法であり、システム開発要員が限定されているような大規模システムの開発に向いている。

(4) RAD (Rapid Application Development)

RADは、少人数で、開発支援ツールを利用して行う高速アプリケーション開発である。早い段階で開発作業にユーザを参画させることができる。もともとは、少人数による開発を実現し、コミュニケーションの円滑化を図ることを目的とした手法である。

(5) ソフトウェアプロダクトライン

ソフトウェアプロダクトラインは、開発するソフトウェア群を分析し、共通利用できる部分とそうでない部分に細分化して開発する手法である。共通利用できる部分をコア資産として開発することで、効率の良いソフトウェア開発を行うことができる。

(6) 繰返し型モデル

繰返し型モデルは、ソフトウェアの開発工程（設計／開発／テスト）を何回も繰り返していく手法である（スパイラルモデルも繰返し型モデルの一種である）。

- ・ **インクリメンタルモデル（段階的モデル）**

ソフトウェアを独立した複数の機能に分割して、機能単位で段階的に開発／リリースしていく手法である。

- ・ **エボリューショナルモデル（進展的モデル又は成長モデル）**

小さな機能範囲のソフトウェアを開発し、改良を加えていく手法である。