

# 開発手法(流れ)

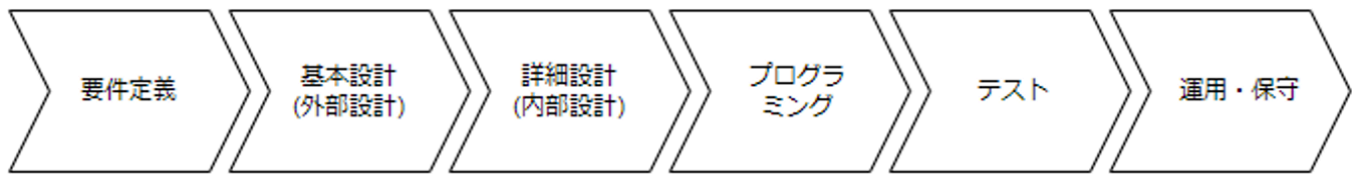
システムの開発手法にはいくつかの手法がありますが、今回はウォーターフォール型の開発手法を中心に次の2つの手法について学びます。

## ウォーターフォール型

プロジェクトをいくつかの工程に分割し、各工程での成果物(設計書などのドキュメント類やプログラムなど)を決める。各工程では決められた成果物に基づき、後工程の作業を順次行っていく開発手法。各工程を上から順番に行っていくためウォーターフォール(滝)モデルと呼ばれる。

SDLCに基づき、要求、仕様、分析、設計、プログラミング、検査、運用に分割。

ウォーターフォールの例



ソフトウェア開発ライフサイクル(SDLC:Software Development Life Cycle)の定義

高品質なソフトウェアの生産を可能にし関連するさまざまな段階を定義する。これにより開発の速度が向上し、低コスト低リスクでプロジェクト全体の改善にもつながる。

要件の収集と分析、設計、コーディング、テスト、展開(インストール)、メンテナンス

## 工程の定義例

ウォーターフォール型では、次のような内容の工程を実施する。

**要件定義**：要求分析を行い、システム化する対象範囲を決める。要件定義書を作成。

**基本設計(外部設計)**：システム化する機能を細分化し、外部仕様を決める。外部設計書、画面設計

書、テーブル定義書などを作成する。

**詳細設計(内部設計)**：システムを実現するための機能をプログラム化する。内部設計書など。

**プログラミング**：コーディングを行う。

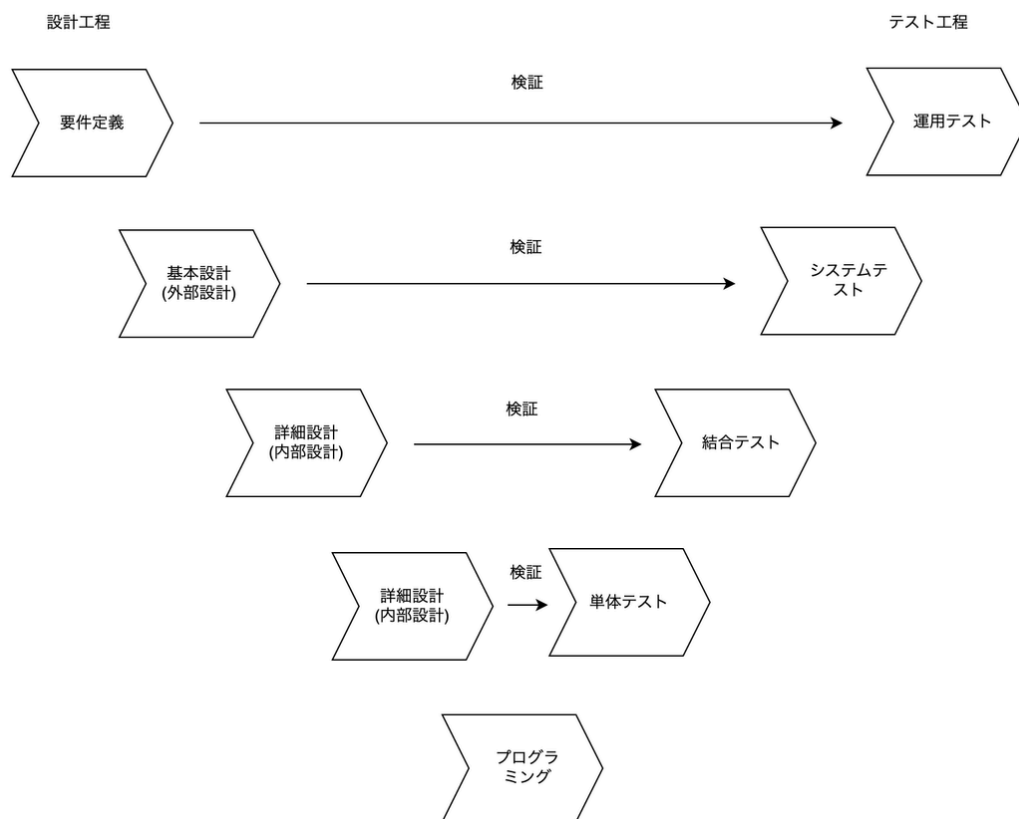
**テスト**：単体テスト、結合テスト、統合テストなどを行う。テスト仕様書を作成。

## ウォーターフォール型の検証

各工程で設計書を作成し、その設計書に基づいてプログラミングを行う。成果物は各工程ごとにレビュー(検証)を行い、承認されたものだけ次の工程へ進むことができる。

承認されなかったものは問題点を修復する作業が必要となる。

### ウォーターフォール型での検証

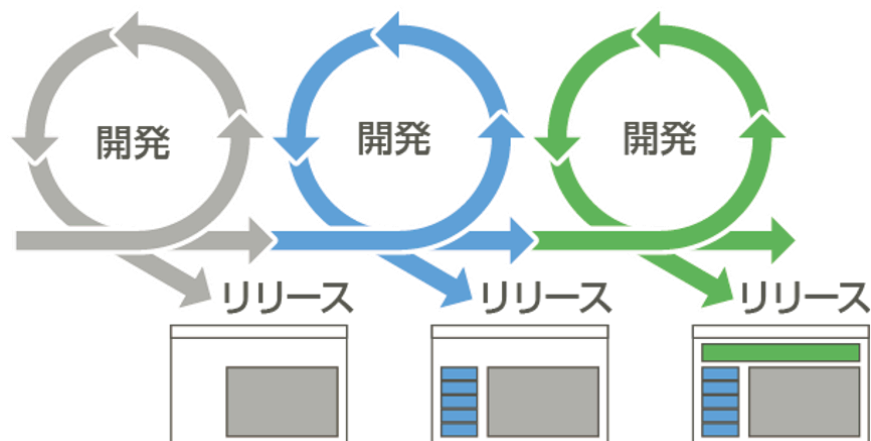


## アジャイル型

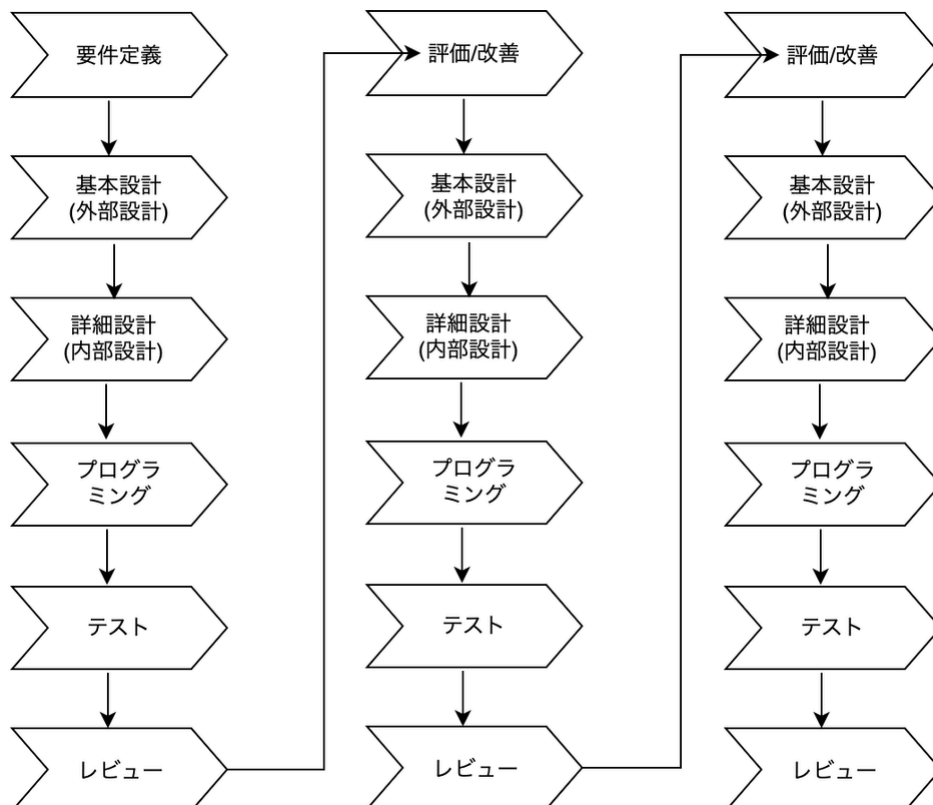
スプリットと呼ばれる小さな開発単で「計画」→「設計」→「実装」→「テスト」の工程を繰り返し開発を進めて行く手法。発注者の要求を少しずつ開発しリリースすることを繰り返します。小さな単位で開発を行うため、開発途中での修正などは容易であるというメリットがある。システムの完成時期の見通しが難しいため、納期ある場合の提供には向いていない。

アジャイル・・・素早いという意味

## アジャイル型のイメージ図



## アジャイル型の開発工程例



## その他の開発手法

その他の開発手法としてプロトタイプモデル、スパイラルモデルなどがある。

### 別ページへのリンク（今回は使用しません）

 プロジェクト計画書の作成