第27講　システム開発（１）

システム開発とは？

第27講　システム開発（１）

- システム開発とは
- 開発手法
- システム開発の流れ

- システム開発とは？

システム開発とは

ビジネス上の目標や課題を解決する

しくみ（=ソリューション）を

ITで実現すること。

企業の課題、目標

新しい事業を興したい

コストを削減したい

業務を効率化したい

品質を向上させたい　など、様々。

ITで解決・・・

アプリケーション開発

ウェブサイト構築

データベース構築

サーバ構築　など。

システム開発の前に・・・

ビジネス課題は何か

最終目標（着地点）

優先順位　などをはっきりさせる。

システム開発の事例

1. 勤怠管理システム

   出退勤時刻、出社日数、有給休暇日数など

2. 在庫管理システム

   在庫数、価格などを管理

3. 生産管理システム

   生産工程や品質などを管理

- システム開発の手法

システム開発の基本的な流れ

1. 要件定義
2. 外部設計（基本設計）
3. 内部設計（詳細設計）
4. 開発（プログラミング）
5. 評価・テスト
6. リリース
7. 運用

1. 要件定義

・システム開発の目的や必要な機能、
　希望の納期などを明確化

・SEが発注元にヒアリング
　求められる要件を確認　**＜要件定義書＞**

2. 外部設計　（基本設計）

・サーバ、クライアントマシンなど
　ハードウェアの選定

・操作画面、システムの「見た目」を設計

・ユーザインタフェースを設計

3. 内部設計　（詳細設計）

・プログラムのための設計

・データ構造

・使用するサーバ、DB、API、

　作成すべき関数、入力・出力　等

４．開発（プログラミング）

・設計に従ってコーディング

５．テスト（評価）

・プログラムの挙動をチェック

・不具合があればデバッグ

６．リリース（引き渡し）

・システムをクライアントに引き渡す

７．運用（稼働）

・システムを動かす

・必要に応じてサポート

　トラブルへの対処　等

代表的な開発手法

1. ウォーターフォール
2. アジャイル
3. スパイラルモデル
4. プロトタイプ

ウォーターフォールとは

- 古くからある手法
- 上流工程から下流工程まで
  順序に従って進める
- 後戻りしない

ウォーターフォールの利点

- 各工程で進捗管理
- スケジュール管理、コスト管理が容易
- 比較的大きなシステムでも
  効率的に進められる

ウォーターフォールの弱点

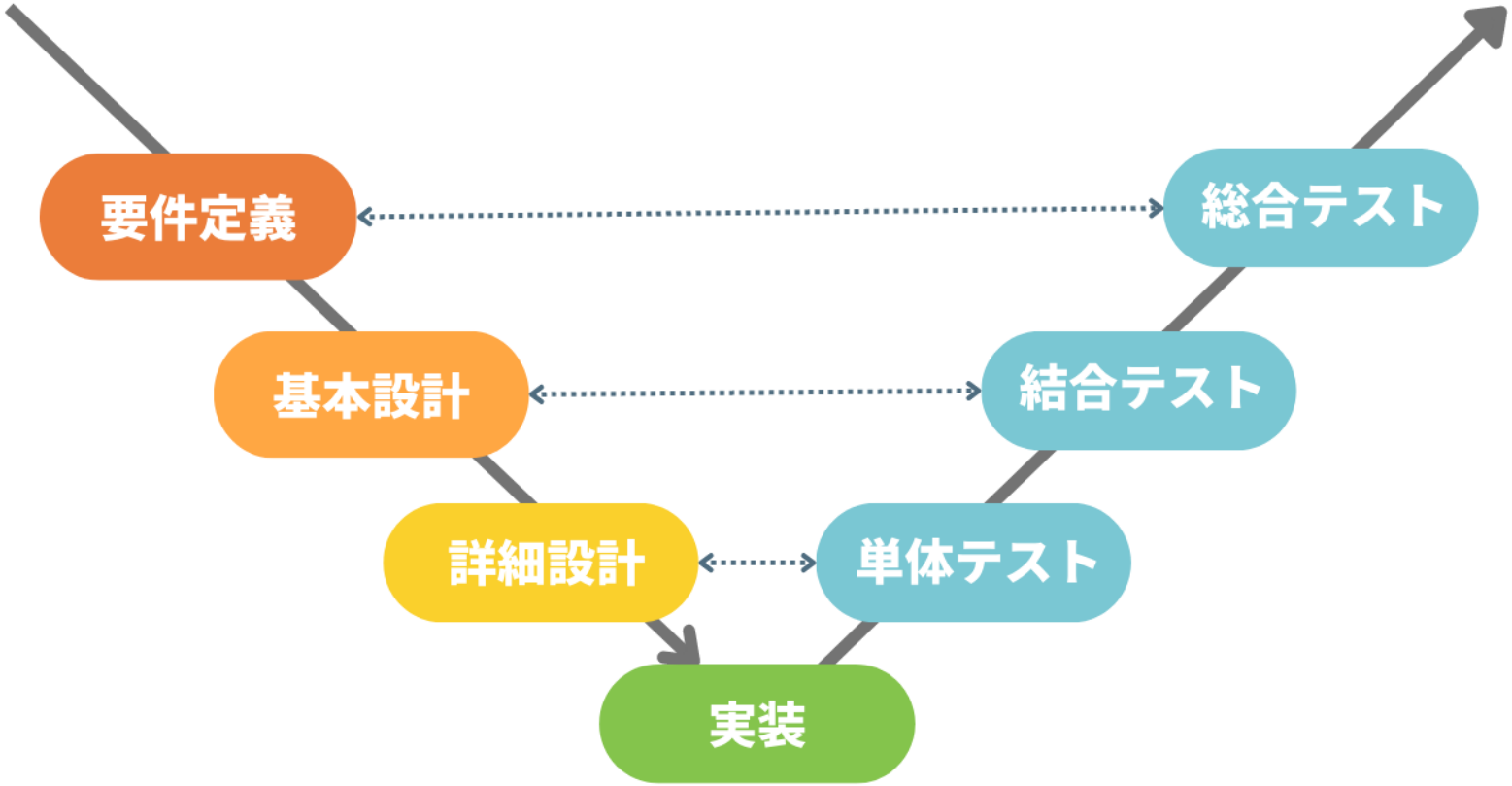
- 仕様は上流工程で決まるため

  下位工程で**仕様変更が難しい**

  → 運用部門のヒアリング

  確実な目標設定　が重要

ウォーターフォールのV字モデル



アジャイル

・上流工程では仕様を確定せず

　　だいたいの方向性だけ決める

・機能ごとに分割して開発・リリース

・近年の主流

アジャイルの利点

・仕様変更が柔軟に行える

　　仕様変更の可能性があるプロジェクト向き

・開発途中でクライアントと連携できる

　　現場で「使える」システムになる

アジャイルの弱点

・一貫性が欠ける恐れ　＜目的を明確に＞

　　当初の方向性とのズレが生じる

・全体の把握、スケジュール管理が困難

・クライアントとの連携が重要

アジャイル

・ユーザニーズへの柔軟な対応

ウェブサービス

アプリ開発　などに向く

スパイラル

・機能ごとに分割（サブシステム）

・サブシステム毎に開発を進め、
　その都度、クライアントが確認

・全体が完成したらリリース

プロトタイプ

・開発（プログラミング）前

　試作をクライアントに体験させる

・ウォーターフォールの改良型

・プロトタイプ制作に手間取る

第27講　システム開発（１）

- システム開発とは
- 開発手法
- システム開発の流れ

第27講はここまで