

システム開発入門

システム開発とプログラム作成(概要)

■プログラム作成(基礎)

プログラム

プログラムの一生

プログラム作成から実行まで

外部データと管理

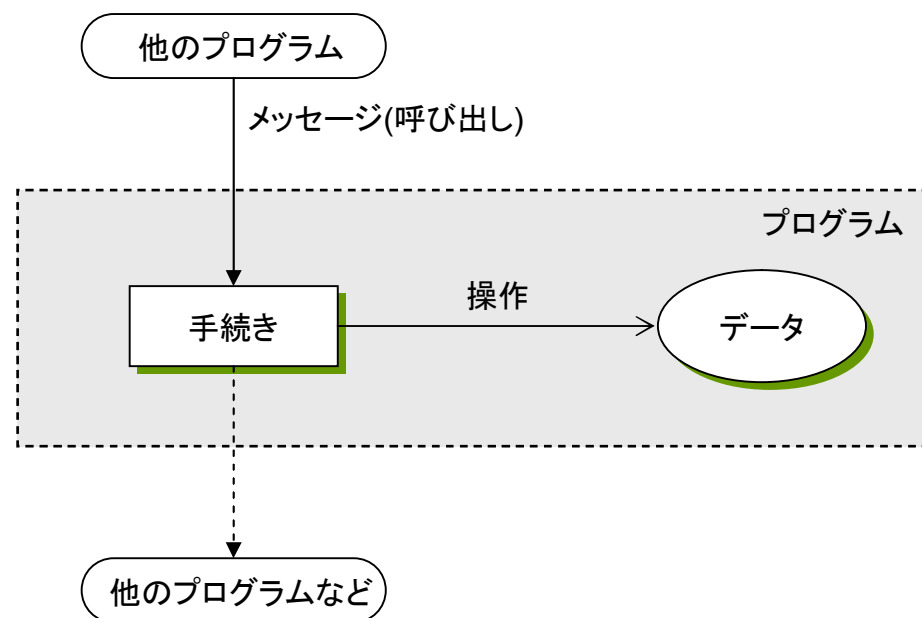
データ型

業務システムの例、方向性

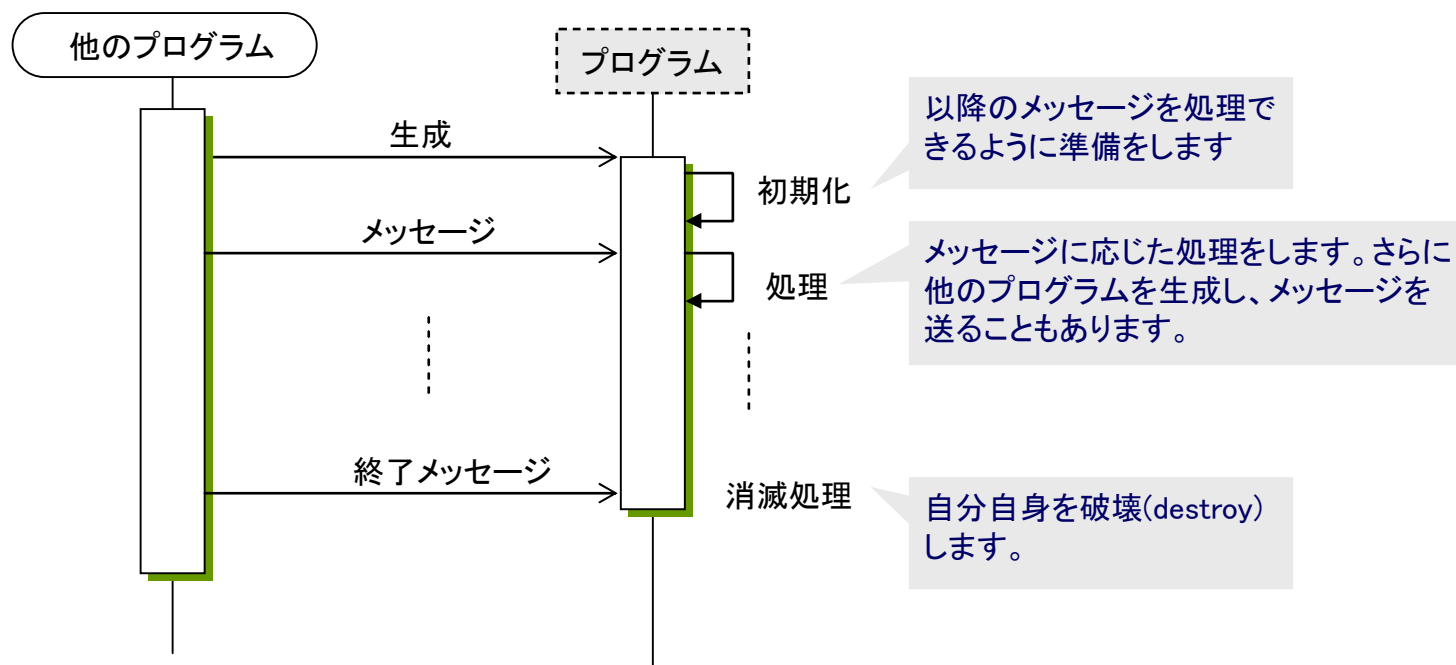
システム開発のプロセスの例

品質システム

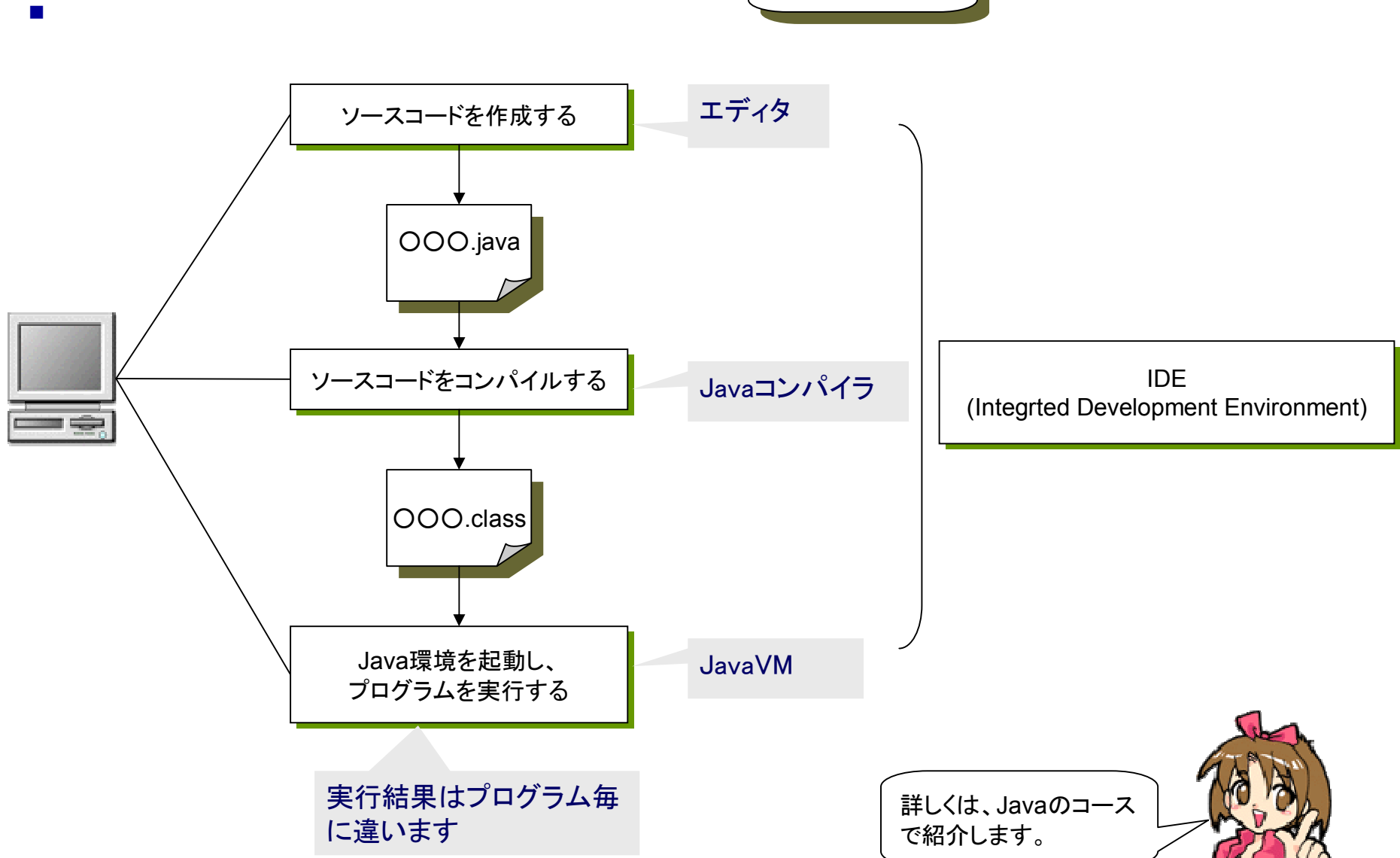
- コンピュータ環境でなんらかの目的を達成するために動作するソフトウェア
- 手続き(処理)とデータから成る
- 呼ばれて動作する
- 手続きはデータを操作(生成、参照、更新、削除)する



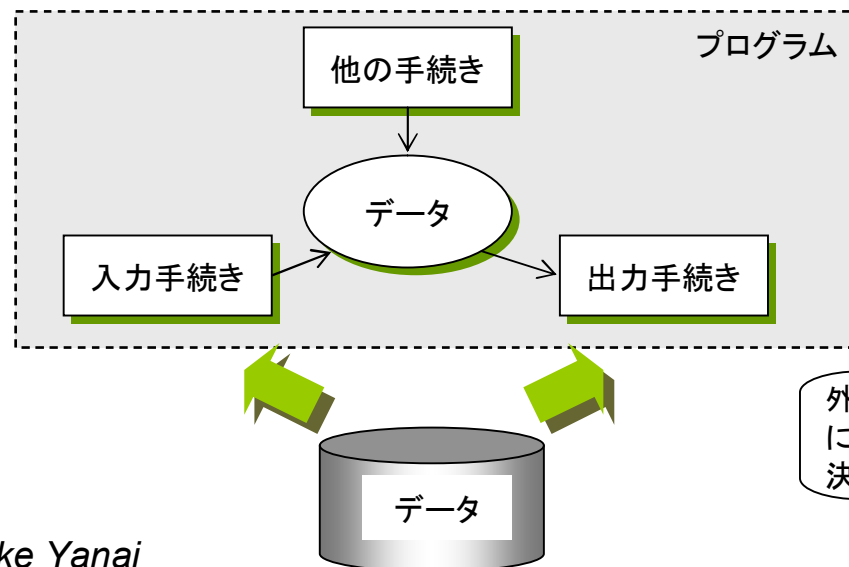
- コンピュータ環境で生まれてから消滅するまで
 - ◆ 誕生する(生成される)
 - ◆ メッセージを受け取り、処理(データを操作)をする
 - ◆ 消滅する



Javaの場合



- データは、プログラムに含まれるものとプログラムとは独立して外部システムに存在するものとがあります
- 長く使用するデータを扱うためには外部システムでの管理が必要です
 - ◆ そのようなプログラムではデータの入出力処理が必要です
- 外部データ管理
 - ◆ ファイル管理
 - ◆ データベース管理
 - ★ RDB(Relational DataBase)
 - ★ OODB(Object Oriented DataBase)
 - ★ XMLDB



外部データ管理の方式に応じて入出力方法が決まります。



データにはいろいろな型があります

- ◆ 標準的なデータ型 文字列、数値、配列、表、画像、音声など
- ◆ 個別業務用のデータ型 帳票、アプリケーション間データなど
- ファイル管理の対象となるデータとしては、文字列が良く使われます
 - ◆ CSV(Comma Separated Value)形式
 - ◆ XML形式
 - ◆ 独自形式

CSV形式(例)

```
部門名, 社員番号, 役職名, 氏名, 生年月日  
第一営業部, 120, 部長, 高橋康史, 1963-2-11  
第一営業部, 237, 山田雄一, 1978-5-10
```

独自形式(例)

```
部門名: 第一営業部  
社員番号:120, 役職名:部長, 氏名:高橋康史, 生年月日:1963/2/11  
社員番号:237, 役職名:, 氏名:山田雄一, 生年月日:1978/5/10
```

XML形式(例)

```
<?xml version="1.0" encoding="Shift_JIS"?>  
<division name="第一営業部">  
<person no="120" title="部長">  
  <name>高橋康史</order>  
  <birthday>1963-02-11</birthday>  
</person>  
<person no="237" title="部長">  
  <name>山田雄一</order>  
  <birthday>1978-05-10</birthday>  
</person>  
</division>
```

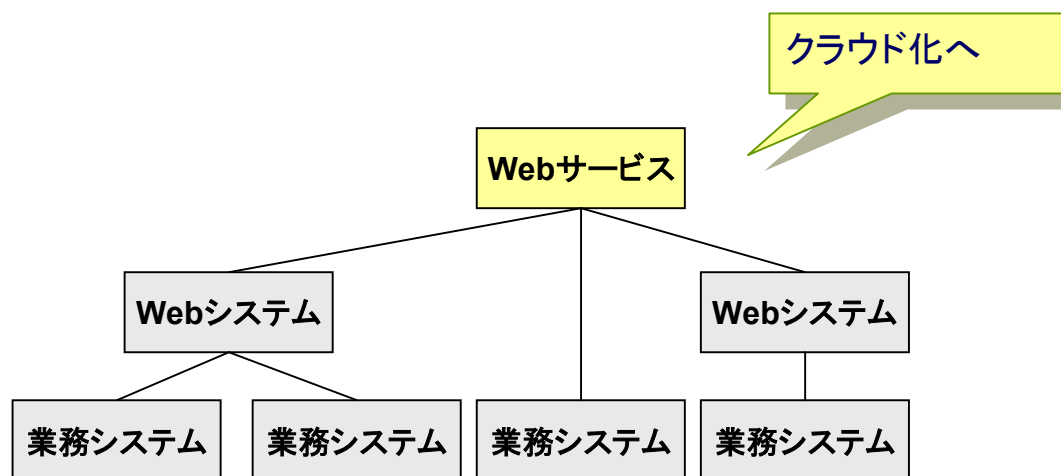
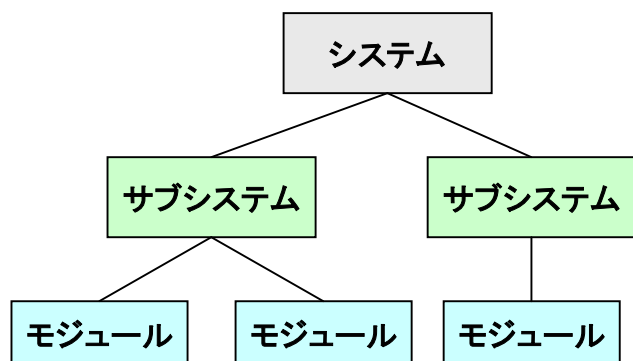
- 業務システムの統合化、異種システムの統合化
- Webサービス

業務システム

受注、発注、在庫管理、販売管理、
財務会計、債務管理、人事管理、
文書管理、設計変更管理、
製造管理、CAD図面管理、
工場管理、製造計画、作業管理、
医療事務、リサイクル物品管理、
地図情報システム、などなど

業務システム間統合/Webサービス

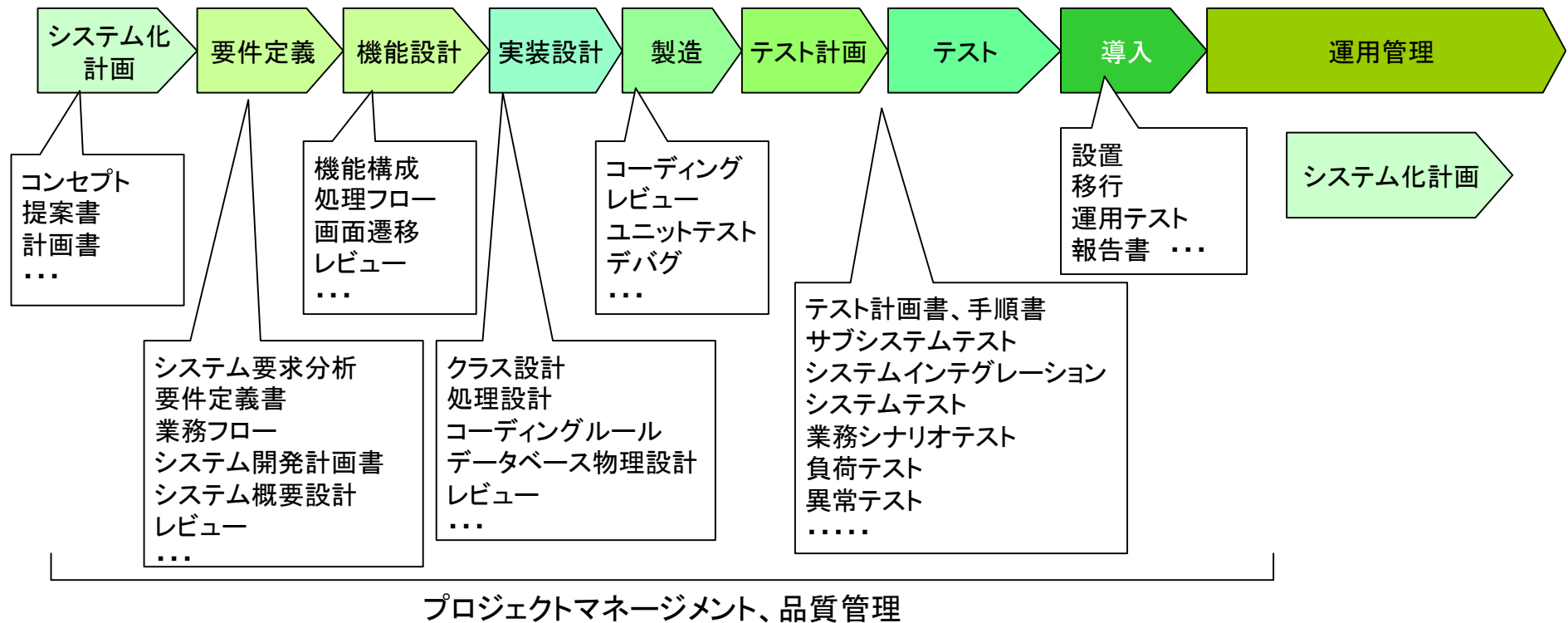
- プロセス統合 (複数の企業や組織の業務プロセスを統合)
 - SCM: 商品調達、在庫管理、配送貨物追跡 など
- ポータルサイト、マッチングサイト、ワンストップサービス
 - 通信販売、旅行予約、工事入札、中古車販売、就職支援、行政手続き など
- その他のWebシステム統合



- システム開発は設計やプログラムを行うだけではなく、たくさんの工程を経て行われる。

Start

End Start...



※詳細項目は、企業やプロジェクトによって
異なります。

- 要求仕様
 - ◆ 開発開始時点で決まっていない、開発中にゆれ動く
- 採用技術
 - ◆ 開発中に新たに登場、改訂がある
- 外部I/F
 - ◆ 他システムや端末がさまざまなI/Fを持ち、変化も激しい
- 開発期間
 - ◆ 相対的に短い
- システムベンダーの責任が大
 - オープンなコンポーネント/プラットフォームから顧客個別のシステムを組み上げる



リスク管理

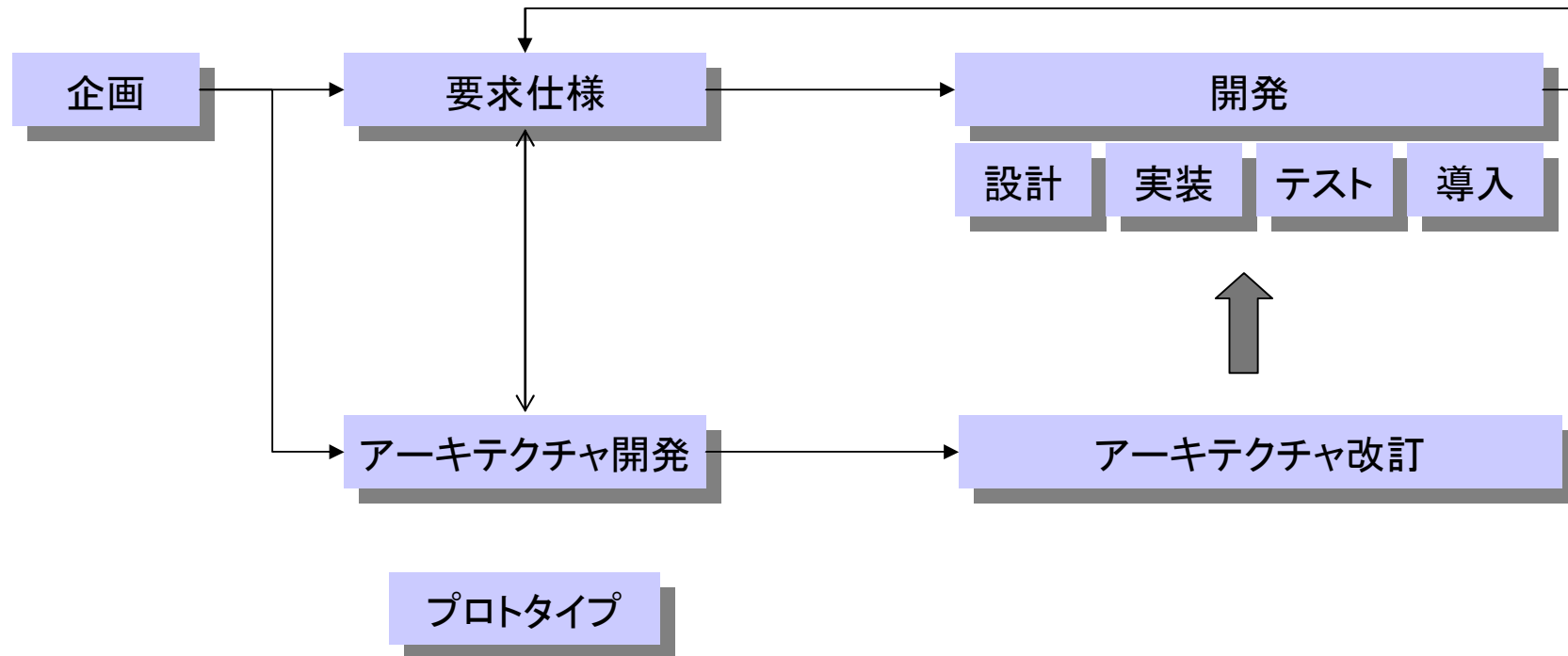
(ウォーターフォール型開発では、プロジェクトの終了間際にリスクが表面化することがある)

スパイラル型開発/イテレーション型開発

少しずつ分析・設計・実装・テストする。
リスクの高いものを優先して手をつける。

具体的なやり方は、プロジェクトの状況/特徴によって選択する。

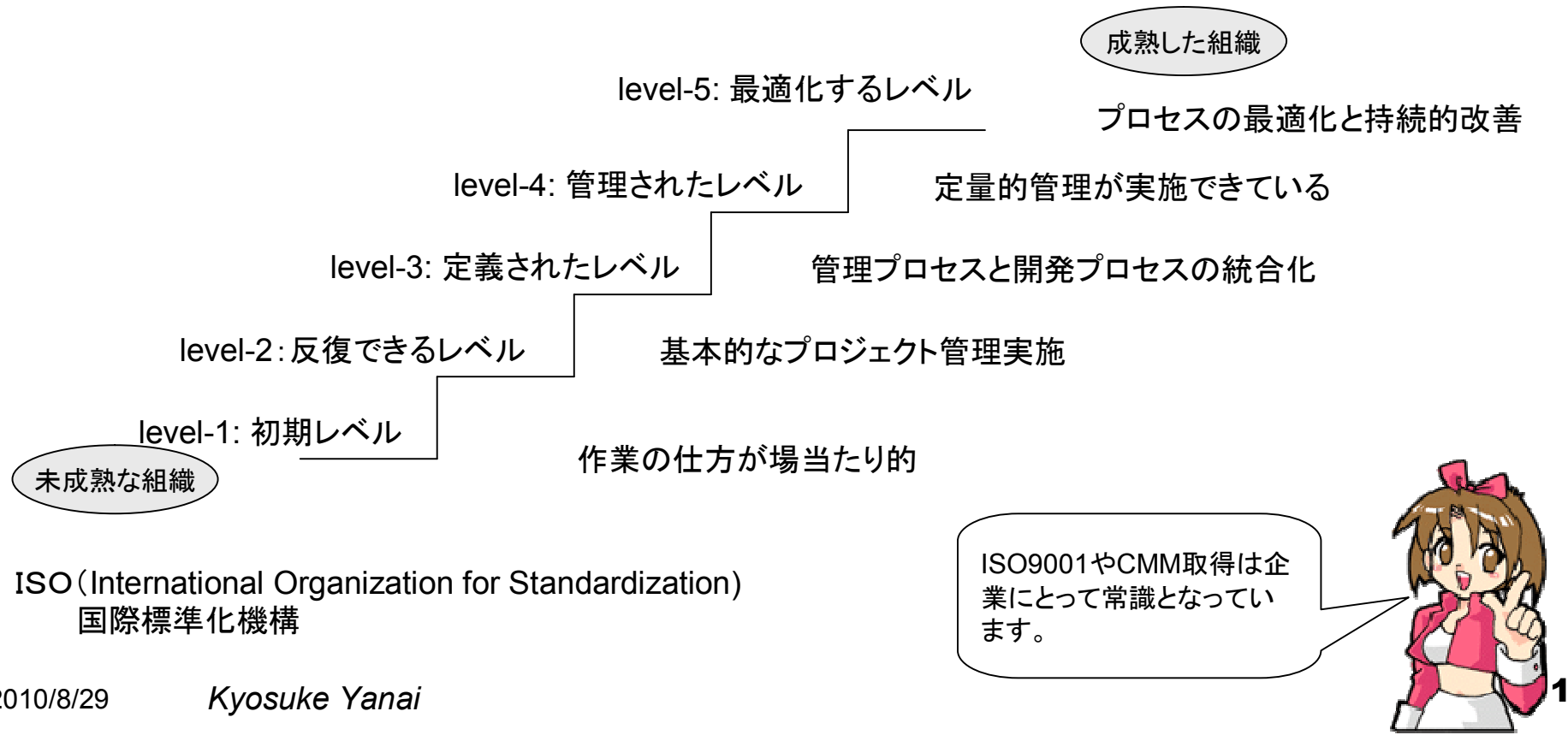
- イテレーション型開発のパターン例(一例です)



ポイント

- ・ユースケースを管理単位として計画・実施管理する
- ・プロジェクトの早期段階でアーキテクチャの設計、プロトタイプによる実現性の検証を行う
- ・変化/変更を恐れない

- 基本的な考え方: 品質は開発プロセス全体によって確保される
要件管理、プロジェクト計画、進捗管理、外注管理、品質保証、構成管理、レビュー、教育訓練など
- ISO 9001:2000 QMS(Quality Management Systems).....JISでは、JIS Q 9001-2000
 - ◆ ISO 9001:2000 は、製品やサービス(情報サービス含む)を購入者が安心して購入するために、供給者側の品質 管理／品質保証の仕組み・体制を規定した国際規格
- CMM(Capability Maturity Model)
 - ◆ 組織のシステム開発プロセスの成熟度モデルをベースとしたプロセス評価を実施



- おつかれさまでした。



これで、
システム開発入門 システム開発とプログラム作成
は終わりです。

