

外部設計

第1回 2日目

大阪情報コンピュータ専門学校

山本 隆之

第1回

<内容>

I．科目オリエンテーション

II．2年～3年次授業「プログラム設計」「内部設計」
「内部設計演習」の復習

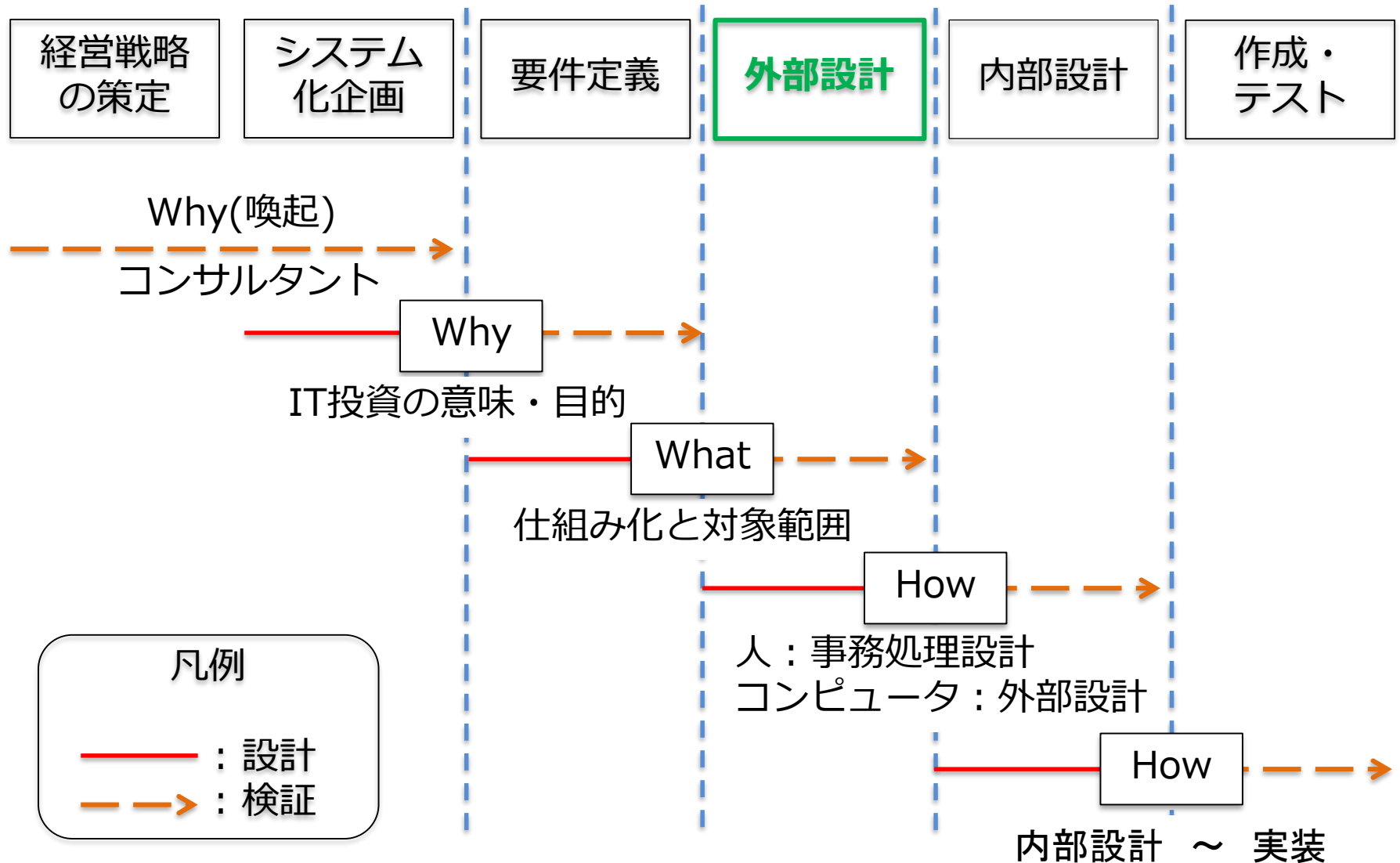
III．実習課題

<内容>

Ⅱ. システム開発基礎の復習

- ・ プログラム開発の流れ
- ・ 開発手法
- ・ 外部設計の目的と手順

プログラム開発の流れ



開発手法

開発の流れを規約化したものの例
(本当はもっとありますが, 有名どころだけ)

- ・ **ウォーターフォール型**
- ・ **スパイラル型**
- ・ **アジャイル型**



ウォーターフォール開発

開発手法

ウォーターフォール型

上流工程

要件定義

システム化要求の調査と分析
システム要件の定義

ソフトウェアの動きを外から見た際にどういう動きをするのか？(What)を決めたもの

外部設計

サブシステムの定義
システム機能仕様の作成
ユーザー・インターフェース設計
データベース設計
移行・運用障害設計

外部設計で決められた動きをどうやって実現するか？(How)を決めたもの

内部設計

プログラム構造の設計
プログラム機能仕様の作成
モジュールの定義
テスト方針書の作成

モジュール仕様作成
コーディング
単体テスト
コンポーネント間統合テスト

開発実施

サブシステム間統合テスト

総合テスト

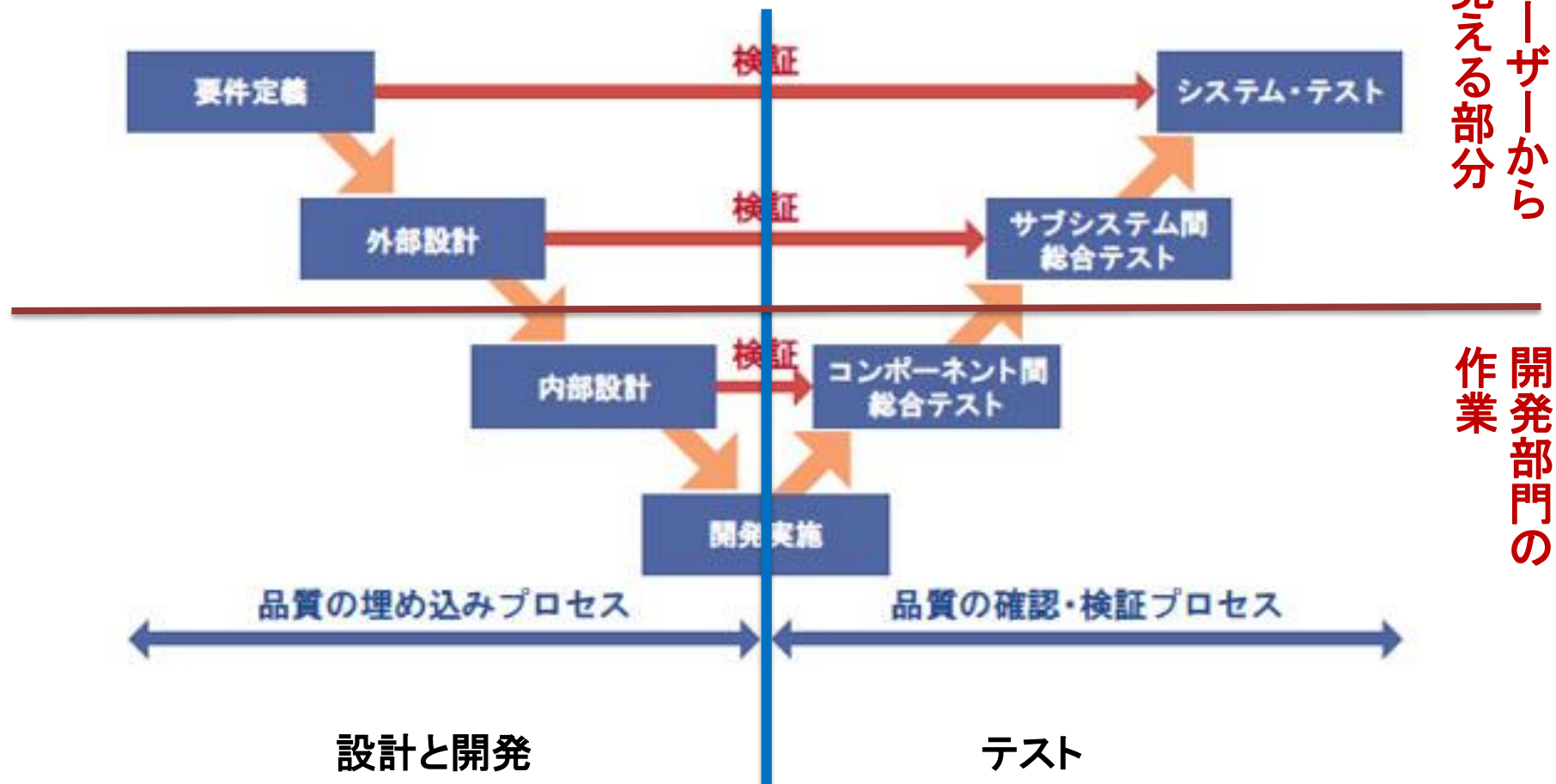
システム・テスト
運用テスト

システム・テスト

下流工程

開発手法

ウォーターフォール型（V字型モデル）



設計手法・図式化手法

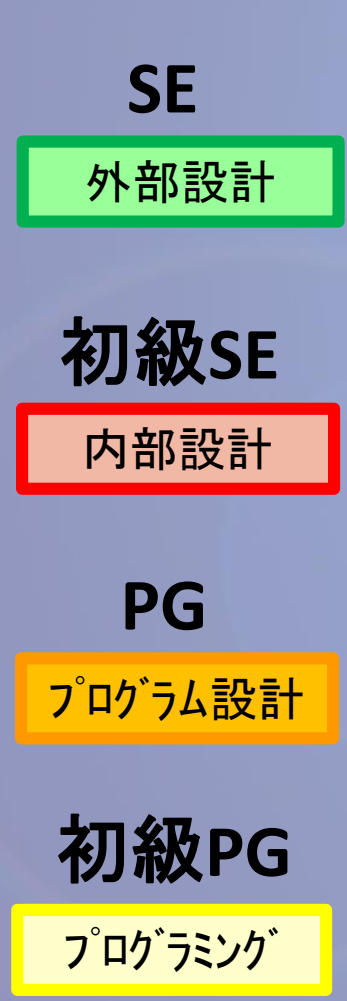
- 基本的な構造化分析・構造化設計を採用

設計書で使用する図式

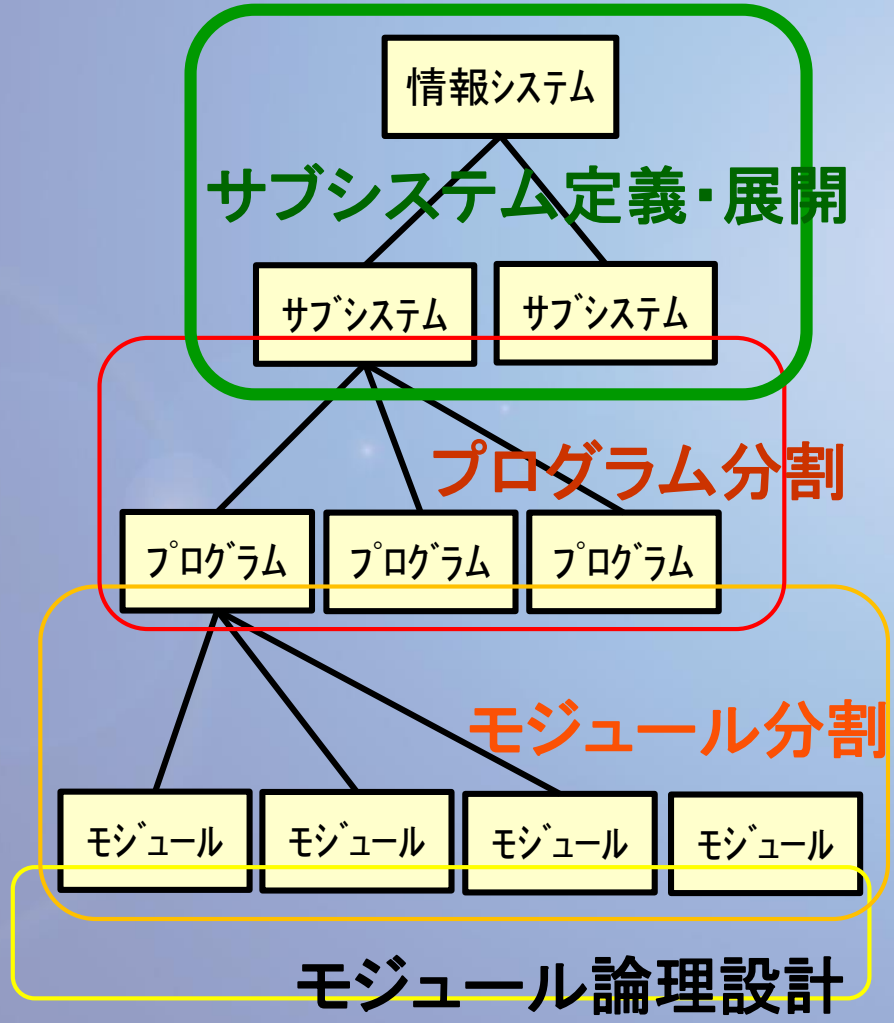
開発工程		外部設計	内部設計	プログラム設計
設計対象		サブシステム	プログラム	モジュール
機能	システムフロー	○		
	DFD	○概略	○詳細	
	HIPO(図式目次) HIPO(IPO)	○サブシステム階層 ○サブシステム仕様	○プログラム階層 ○プログラム仕様	○モジュール階層 ○モジュール仕様
入出力	画面遷移図	○概略	○詳細	
	画面レイアウト図	○概略	○詳細	
	帳票レイアウト図	○概略	○詳細	
データベース	ER図	○概念モデル	○詳細モデル	
	テーブル定義	○概略	○詳細	
	コード一覧	○		

情報システム階層構造と職種・開発工程との関係

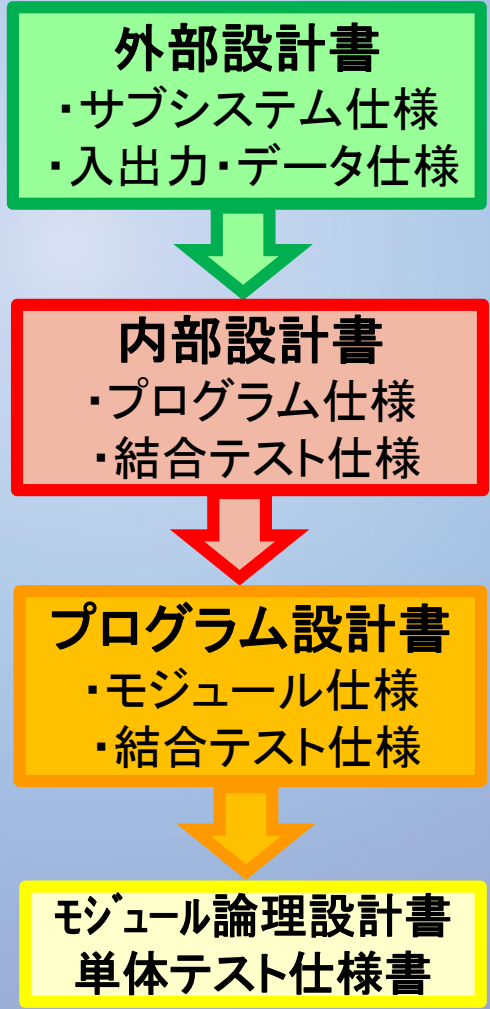
AP職種・工程



情報システム階層構造



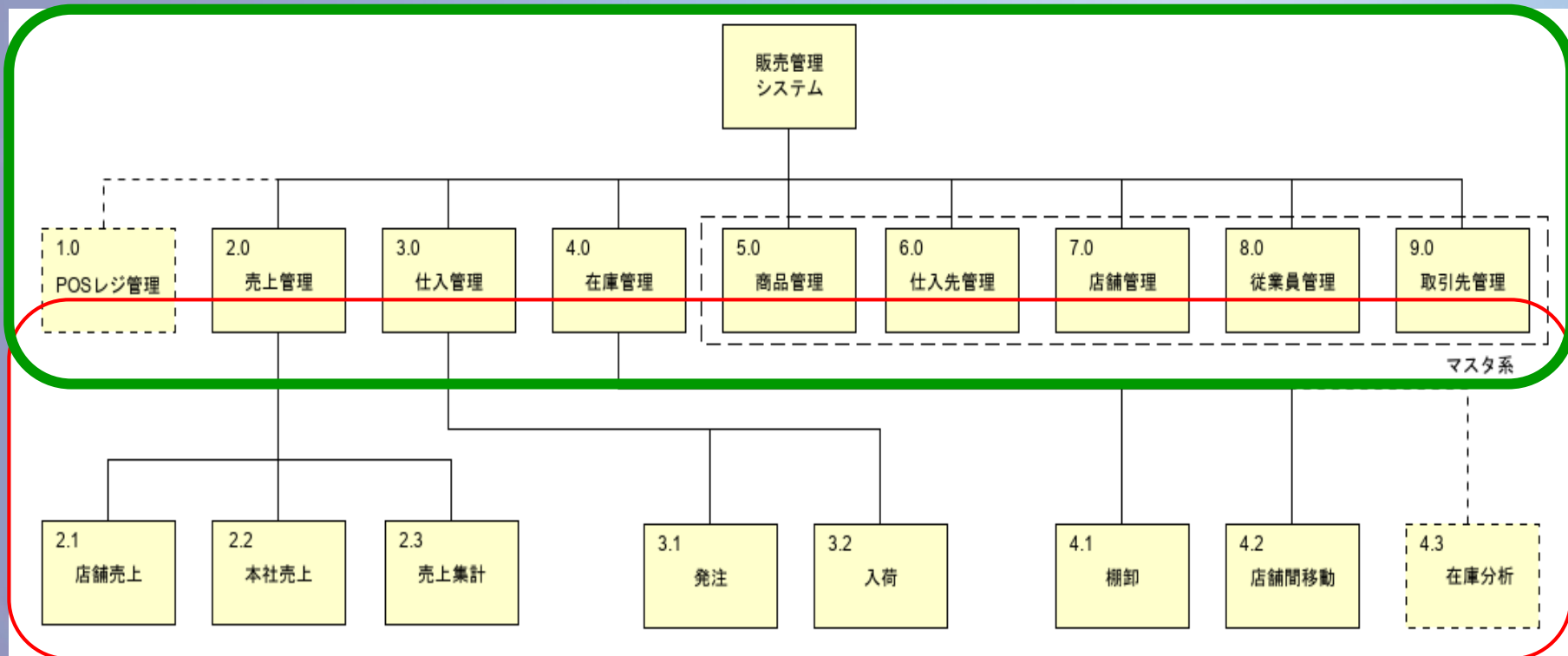
主なドキュメント



サブシステム構造図の例

- ・ 上流工程から構造化設計を進めて内部設計を実施中

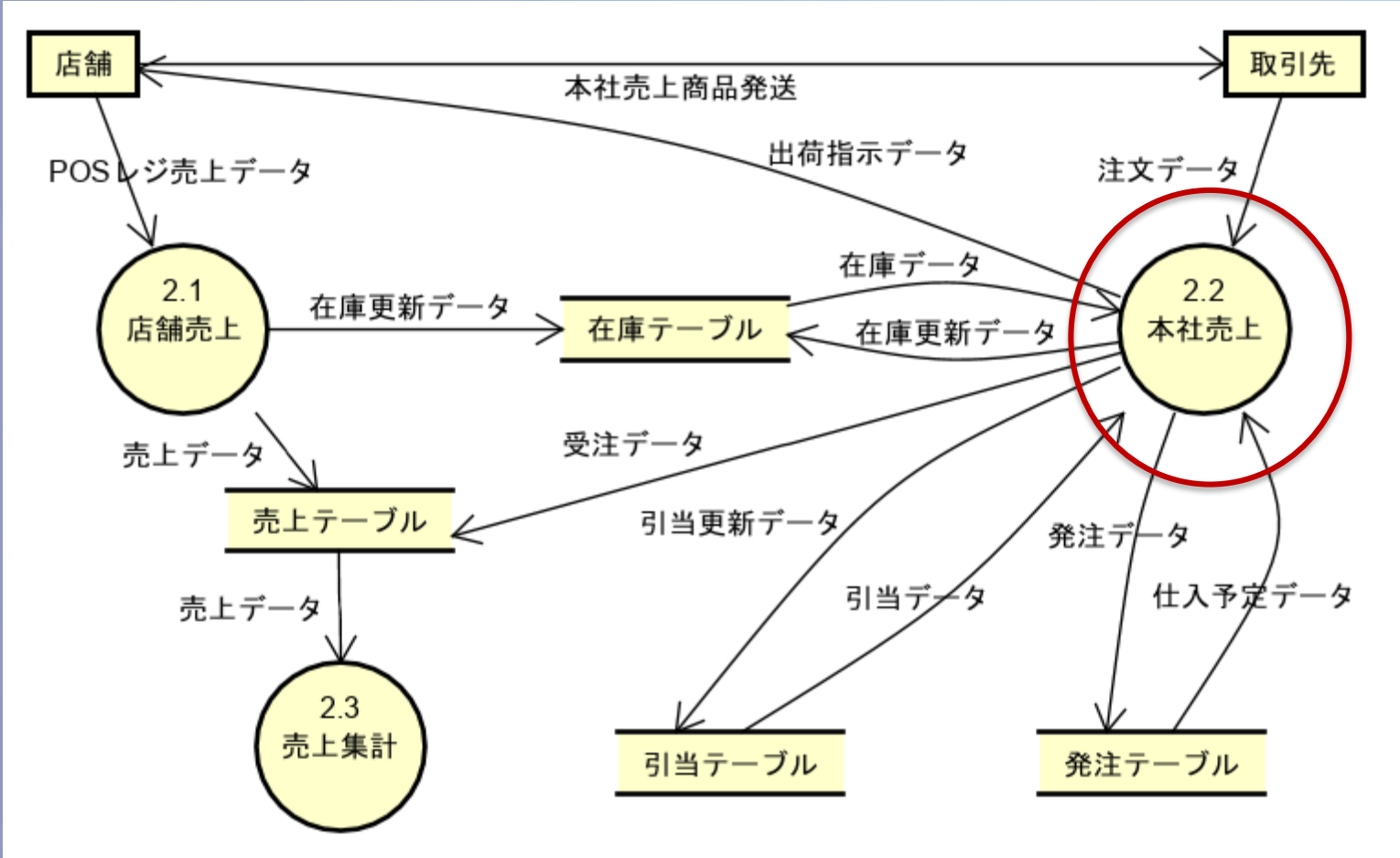
販売管理システム サブシステム構造図



— 外部設計
— 内部設計

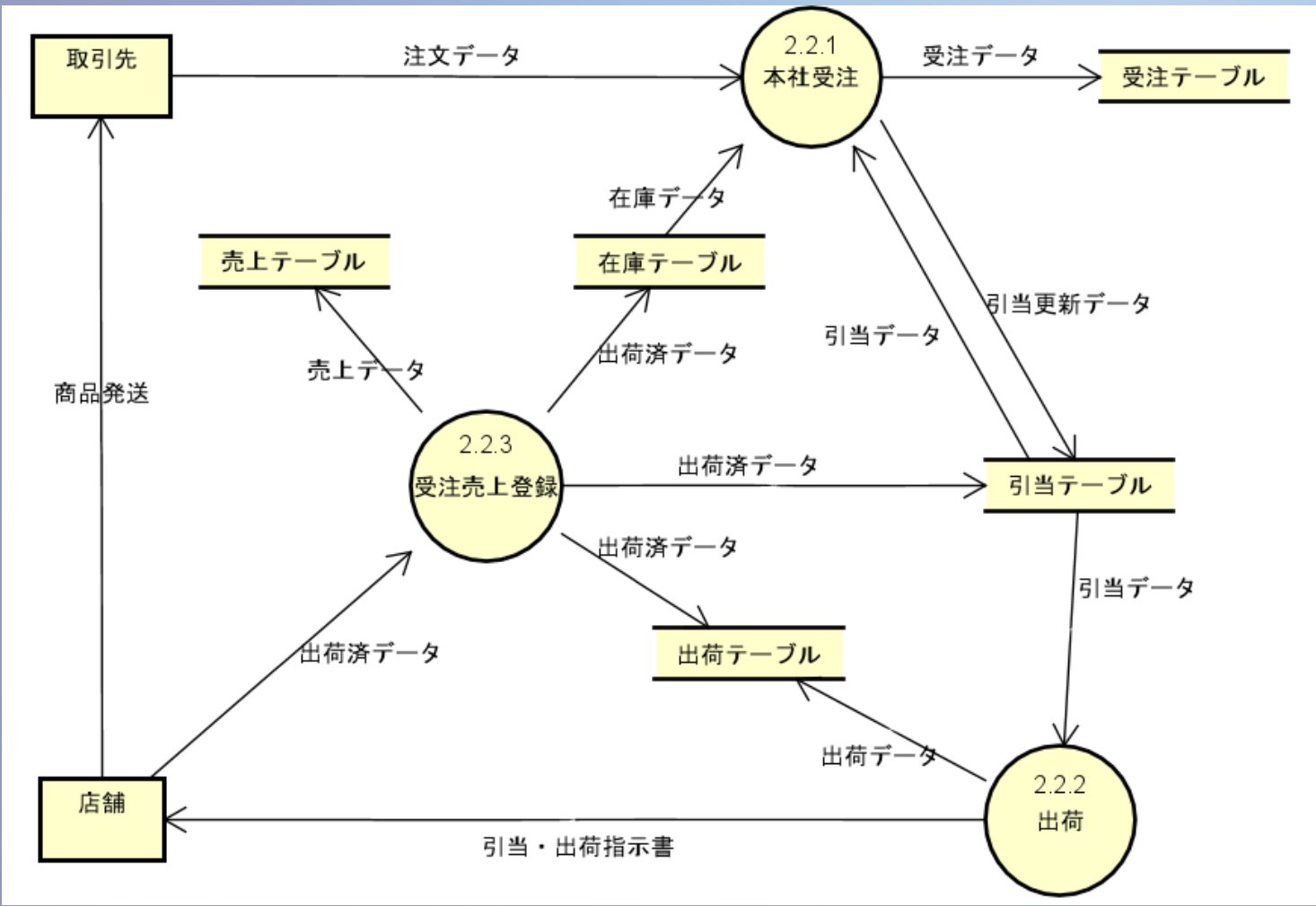
DFDの例 (外部設計範囲)

売上管理サブシステム 概略DFD



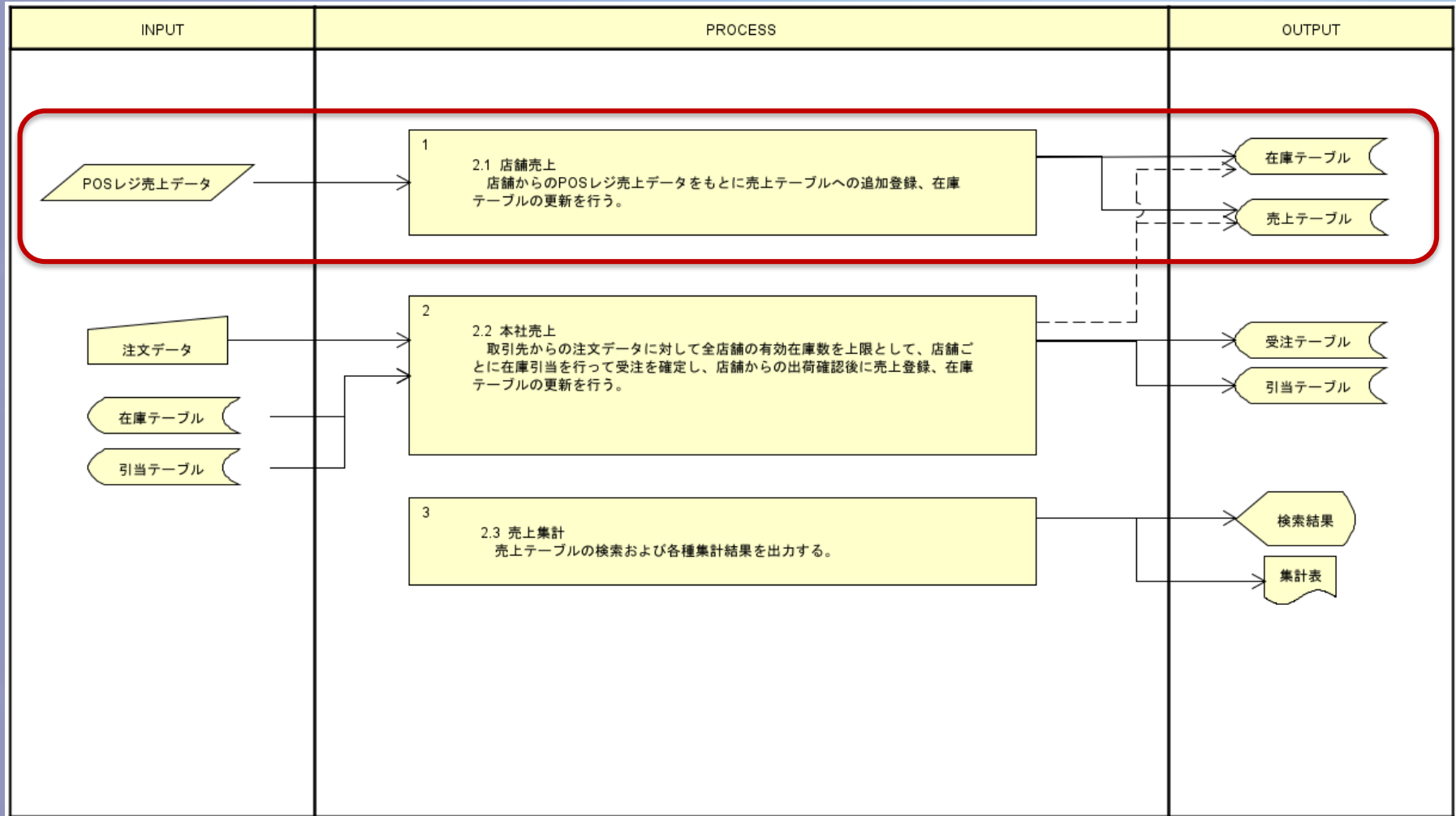
DFDの例 (内部設計範囲)

売上管理サブシステム 詳細DFD



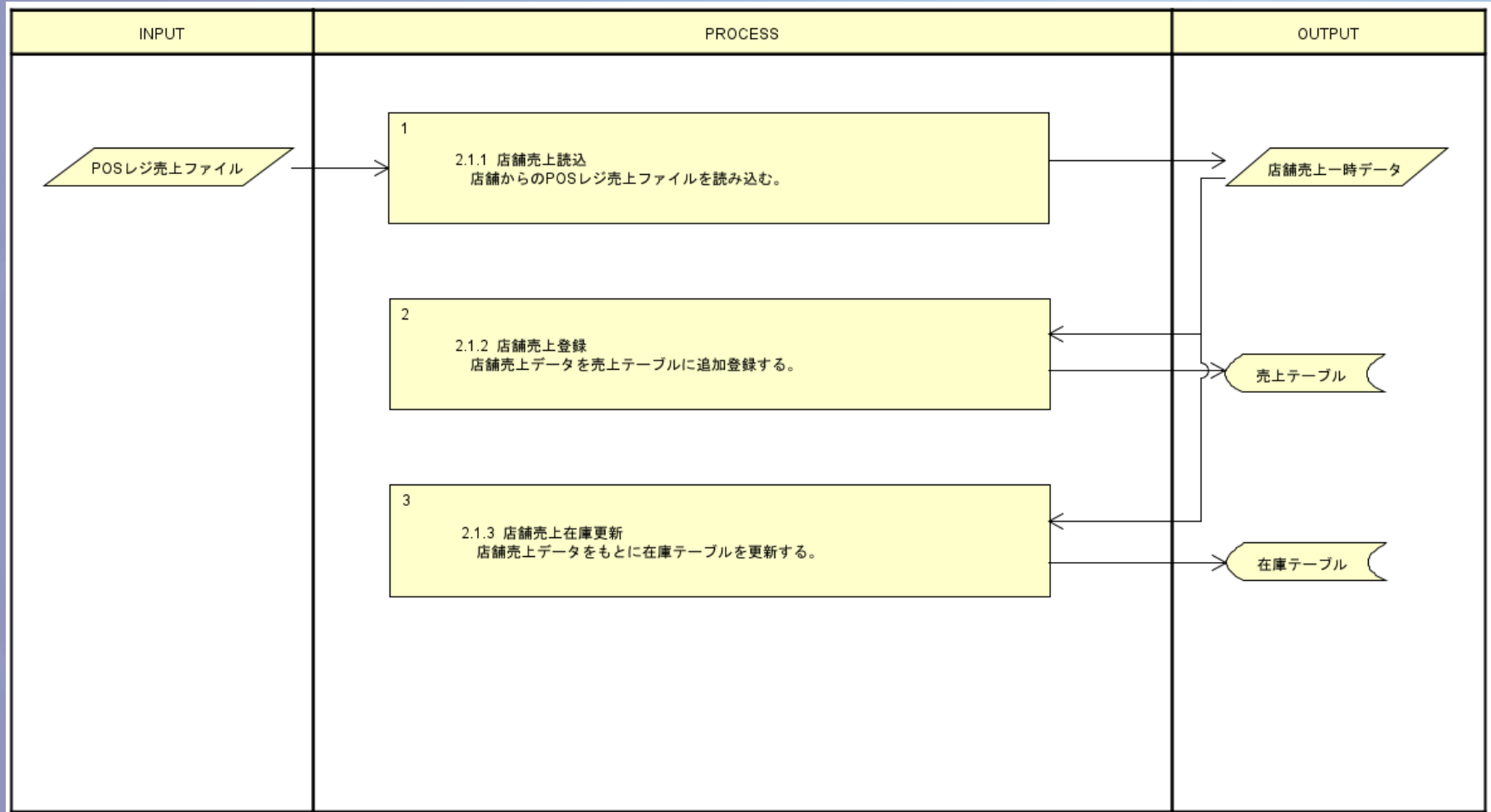
IPOダイアグラムの例 (外部設計範囲)

売上管理サブシステム 概略IPOダイアグラム



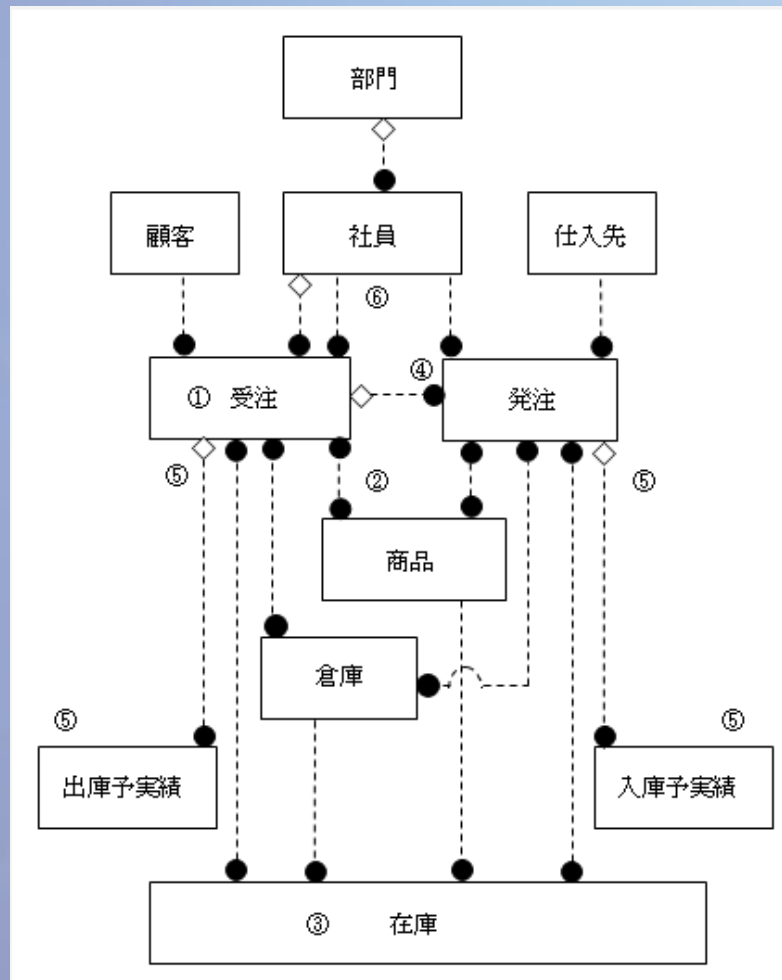
IPOダイアグラムの例 (内部設計範囲)

売上管理サブシステム 詳細IPOダイアグラム



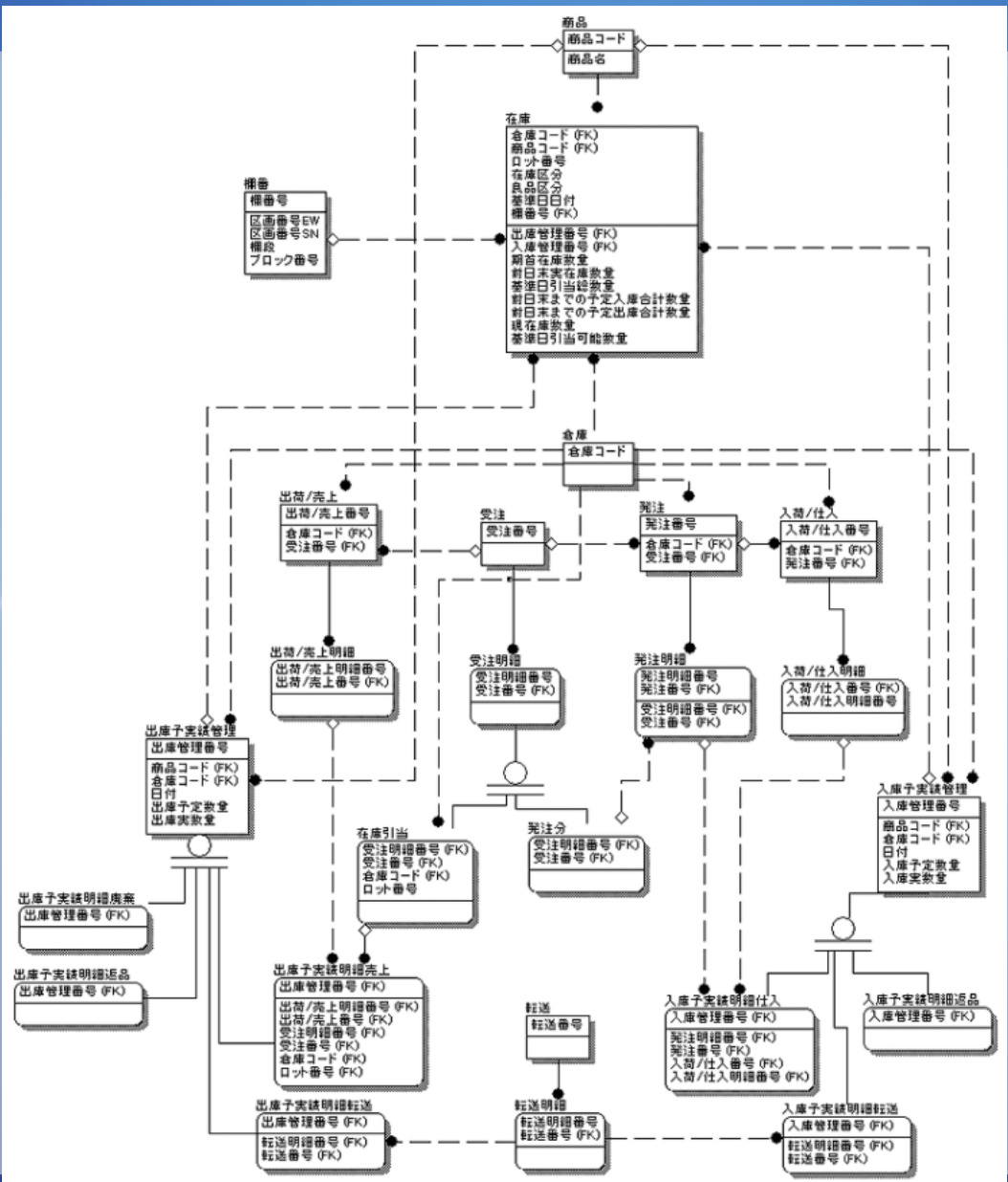
ER図の例 (概略ER図)

販売管理システム ER図



ER図の例 (詳細ER図)

販売管理システム ER図



<内容>

Ⅱ．システム開発基礎の復習

- ・ プログラム開発の流れ
- ・ 開発手法
- ・ 外部設計の目的と手順

外部設計とは

＜外部設計の目的＞

- － システムの機能やヒューマンインターフェースについて**ユーザの立場で定義**する

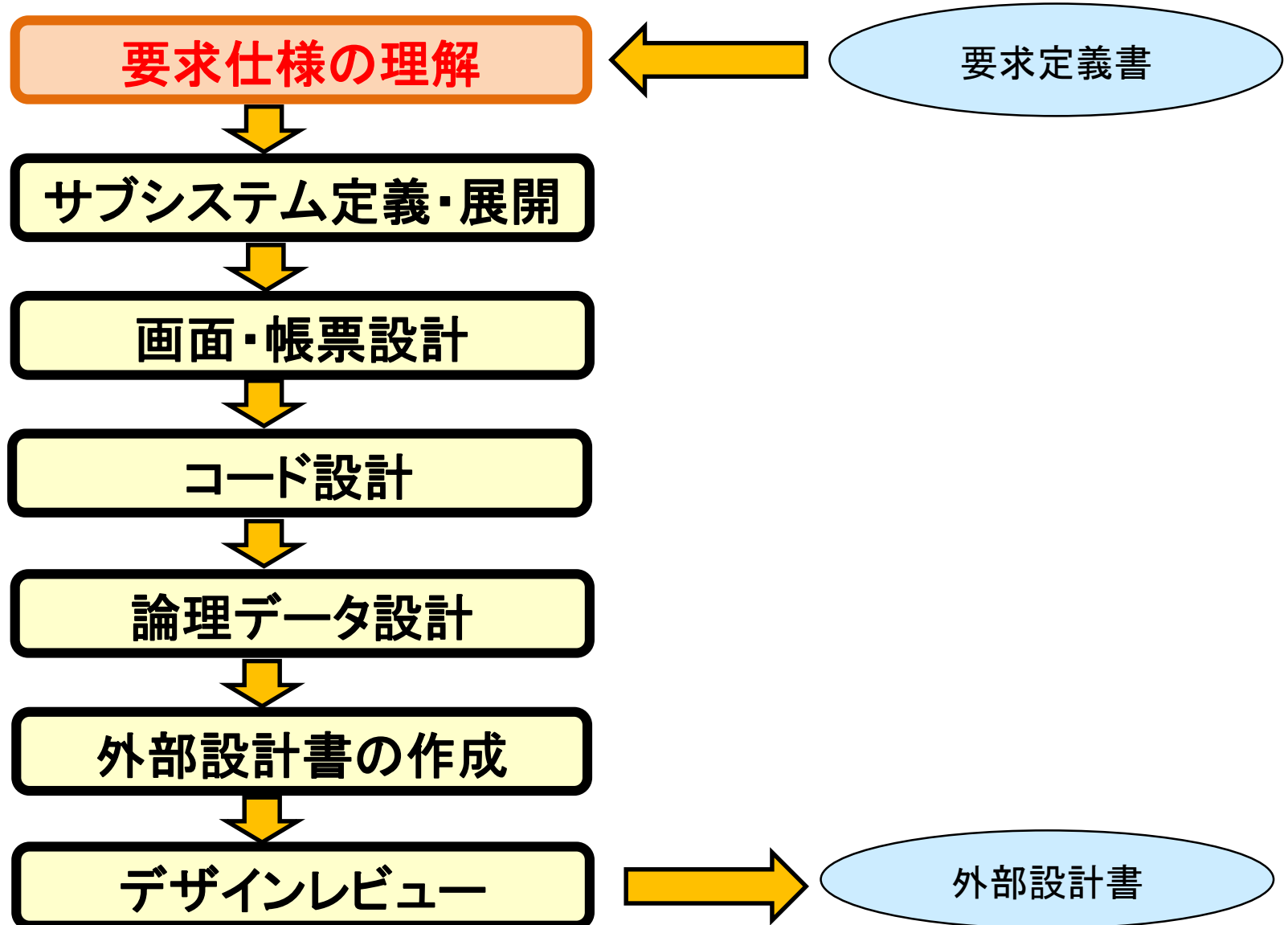
プログラムのスキル不要

（外部設計はユーザの立場から見てシステムを設計する工程）

- － ユーザの意見を十分に反映し、「**何を入力**」すれば「**どのような出力**」が「**どのようなタイミング**」で得られるかということを理解できるようにする

外部設計とは

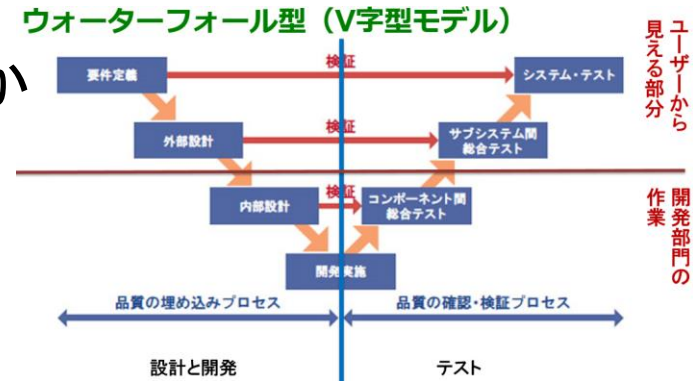
＜外部設計の手順＞



外部設計(要求仕様の理解)

外部設計で決めること

- ・どのような業務をシステム化するか
- ・どのような業務の流れをシステム化するか
- ・利用者にどんな機能を提供するか
- ...等



ユーザーが実際操作して確認できるのは、テスト**工程の最終段階**！
きちんと設計しておかないとテストや本稼働後の運用で大きな**手戻り**
が発生し、システム利用者に**多大な負担を強いる**ことになる。

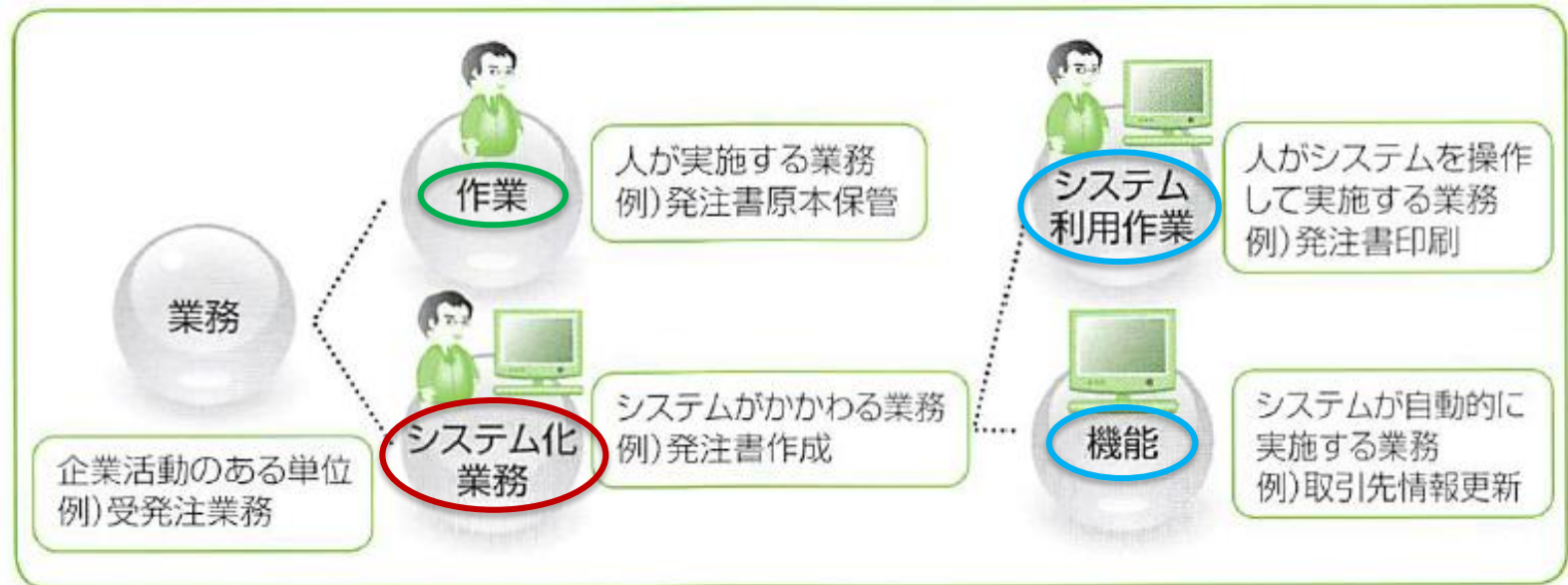
ユーザー:「**動くものを見ないとわからない**」

システム導入後の業務の流れをユーザーがイメージできるように
設計書を作成し、**レビュー**することが不可欠です。

外部設計(要求仕様の理解)

業務は大きく「**作業**」と「**システム化業務**」に分かれる。
「**作業**」は人が実施する業務で、「**システム化業務**」はシステムがかかわる業務です。

「**システム化業務**」はさらに、
人がシステムを操作実施する「**システム利用作業**」と
システムが自動的に実施する「**機能**」に分かれる。



外部設計(要求仕様の理解)

・システム化業務の明確化

1) システム化業務一覧の作成(システム化業務の洗い出し)

No.	機能名	機能概要
1	発注書作成/更新/削除	発注書を新規作成/更新/削除する
2	各種日時更新	作成日時、登録日時、変更日時を自動更新する
3	支払条件入力	支払条件を入力する
4	発注金額算出	発注金額を自動算出する
5	各種選択	発注先、納品場所、担当者、部品情報は、入力したテキストから 選択リストをリアルタイムで絞り込む
6	発注先表示	部品に対応する発注先を部品マスターに登録された情報から自動表示する
7	発注書一時保存	作成中の発注書を一時保存する(1 ユーザーにつき1 つまで)
8	発注書複写	発注書を複写して新規登録する
9	発注先設定	発注先を部品名、納品場所、担当者から自動設定する
10	発注書変更	発注書を変更する
11	マスター取込	各種マスターから自動的に既定値を取得する
12

どの業務がどこまで抽出できたのか？

階層構造に分けていないため、多数の業務から重複した機能を発見しにくい

外部設計(要求仕様の理解)

・システム化業務の明確化

1) システム化業務一覧の作成

No.	大機能	中機能	小機能	機能概要
1	発注業務	発注書作成	発注書作成/更新/削除	発注書を新規作成/更新/削除する
2			発注書変更	発注書を変更する
3			発注書一時保存	作成中の発注書を一時保存する(1 ユーザーにつき1つまで)
4			各種日時更新	作成日時、登録日時、変更日時を自動更新する
5			発注金額算出	発注金額を自動算出する
6			発注先表示	部品に対応する発注先を部品マスター情報からを自動表示する
7		発注書再利用	発注書複写	発注書を複写して新規登録する
8		入力支援	各種選択	支払条件を入力する
9			支払条件入力	発注先、納品場所、担当者、部品情報は、入力したテキストから選択リストをリアルタイムで絞り込む
10			発注先設定	発注先を部品名、納品場所、担当者から自動設定する
11			マスター取込	各種マスターから自動的に既定値を取得する
12		
13		

システム利用作業や機能を**階層構造**に分けて記述すると、業務に近いものから段階的に確認でき、**要求定義書**からの落とし込みの段階で**抜け**や**漏れ**がないかを確認しやすくなりユーザーが理解しやすい。

階層構造に分けると重複機能を発見しやすく、足りない機能も分かりやすい

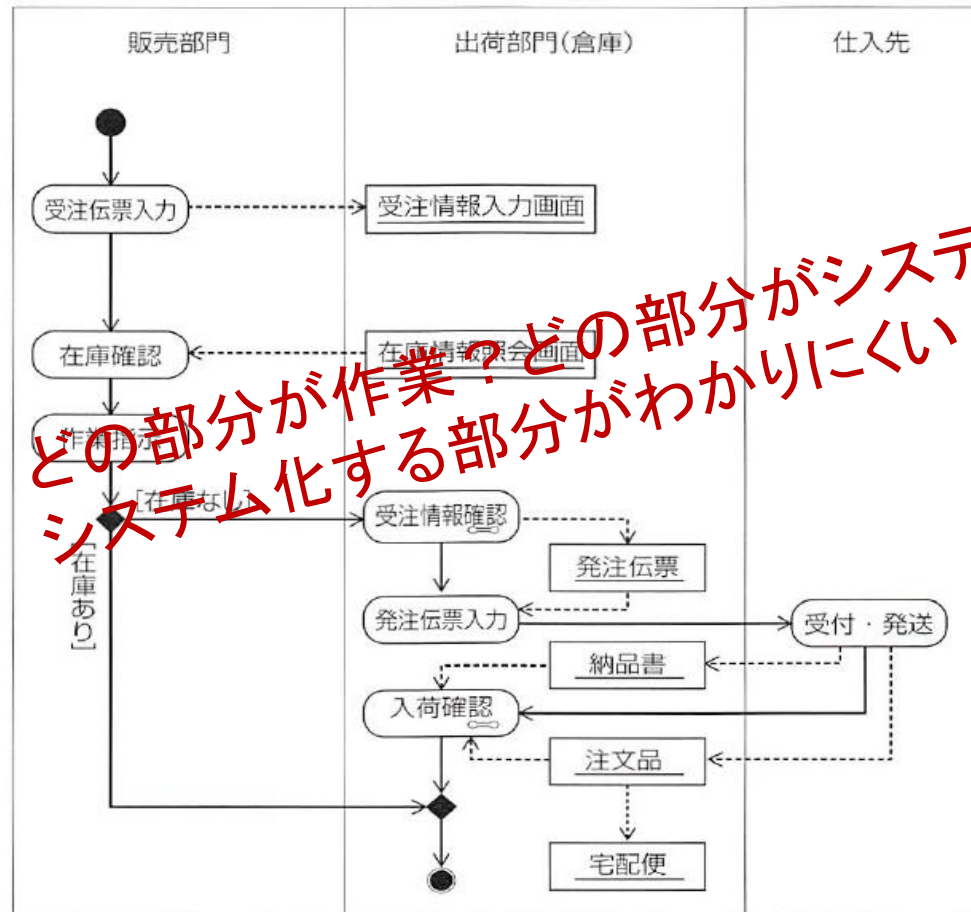
外部設計(要求仕様の理解)

・システム化業務の明確化

2) システム化業務フロー図の作成(システム化業務の流れ)

- **どのような手順**で動作し、個々の「作業」は**どのタイミング**で発生するのか？ システム全体の動きを把握できるように。

受注伝票入力業務例



どの部分がシステム化？
どの部分が作業？
どの部分がわかりにくい！
システム化する部分がわかりにくい！

外部設計(要求仕様の理解)

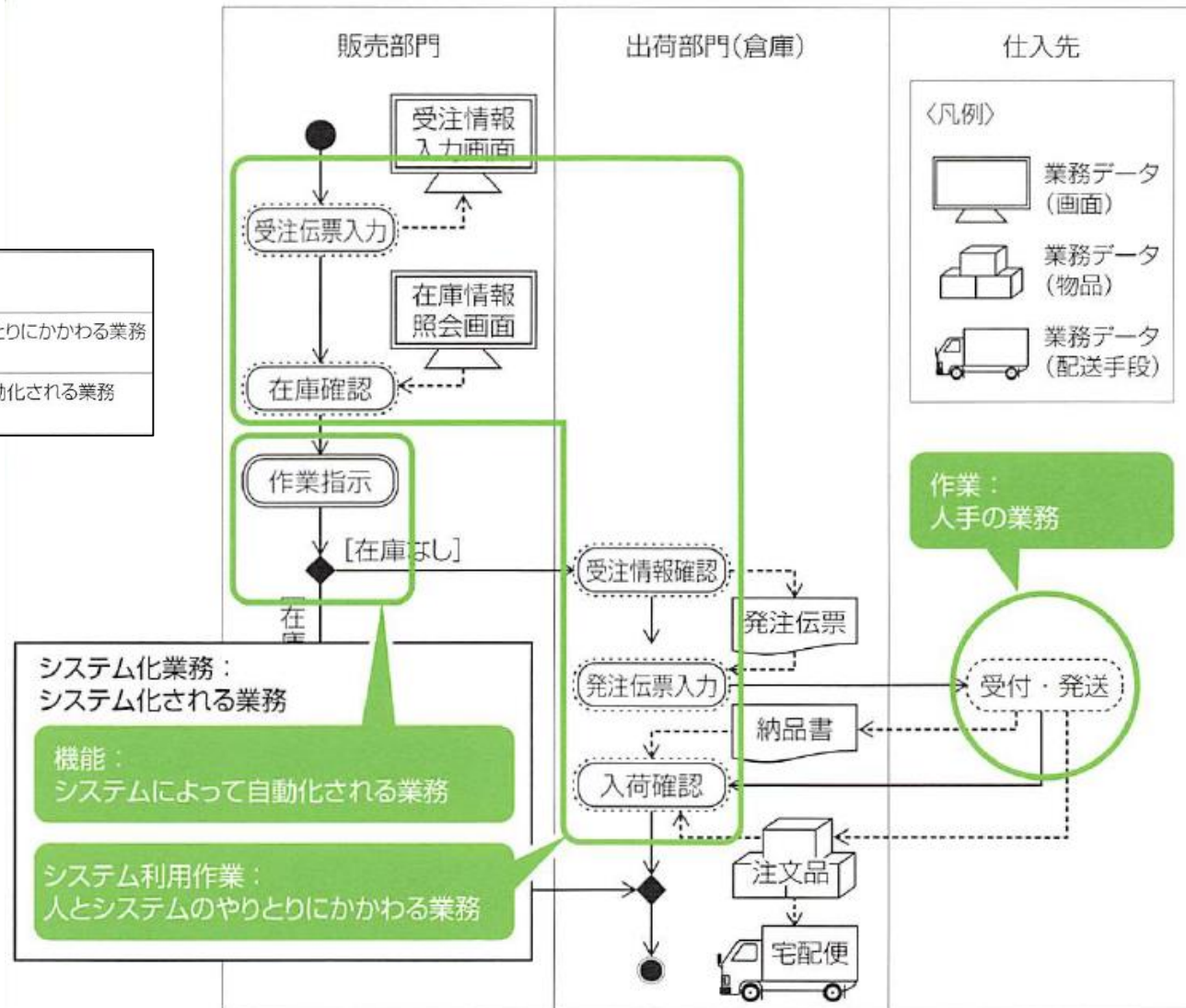
・システム化業務の明確化

2) システム化業務フロー図の作成(システム化業務の流れ)

アイコンを分類した図例

	作業		人手の業務
	業務 システム化	システム 利用作業	人とシステムのやりとりにかかわる業務
		機能	システムによって自動化される業務
			

視覚的にわかりやすく理解しやすい



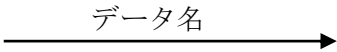
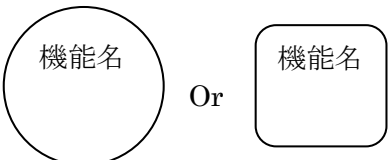
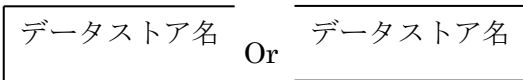
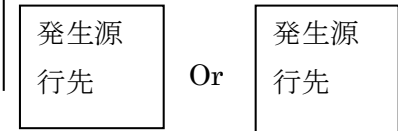
外部設計(要求仕様の理解)

・分析手法

(プログラム設計、内部設計授業の復習)

DFD

DFD の記号

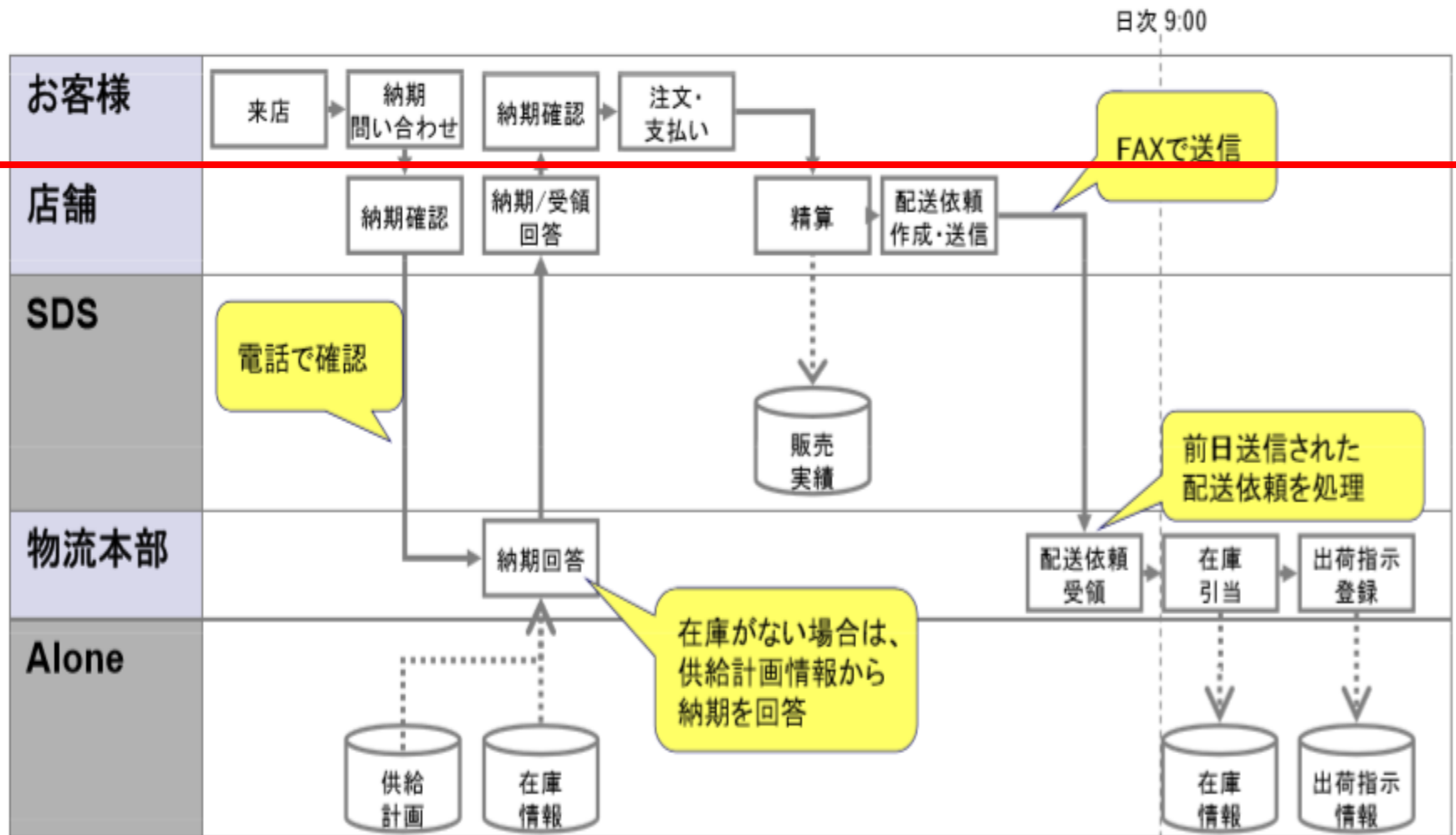
名称	記号	意味
データフロー		<ul style="list-style-type: none">データの流れを示す。データには名前を付け矢印線の上に記述。矢印線は分岐、合流させない。
機能 (処理)		<ul style="list-style-type: none">入力から出力データへの変換を示す。処理内容、処理名を記号内部に記述する。
データストア		<ul style="list-style-type: none">記述されるデータ名を記述する。データ名で示されるファイルである。
外部		<ul style="list-style-type: none">コンピュータの外部にある部門を示す。データの発生源または、行先を記述する。



参考： LOVEMチャート(Line of Visibility)

大型商品業務フロー

来店～注文～在庫引当～出荷指示登録



外部設計(要求仕様の理解)

●分析手法(プログラム設計、内部設計授業の復習)

◆HIPO

・Input(インプット)、 ・Process(処理)、 ・Output(アウトプット)

なぜ、IPOが大切なのか？

・関係部署、関係者が多い、・プロジェクト期間が長い、・仕事内容が複雑

=> 相手に伝わりにくい

<相手に伝える3つのルール>

【短く伝える】

あらかじめ、伝えることは何なのかを用意する。大切なのは、「最低限」相手に伝えるべきことが何なのかを意識すること。

【構造意識して伝える】

話の内容は論理的に話す。始めの話と結論の話をセットで伝えることで、より相手に対して伝わりやすくなる。

【イメージで伝える】

図や絵などを書いて相手にイメージをしてもらう。相手とイメージを共有することで、ミスを防ぐ。



外部設計(要求仕様の理解)

◆図式目次(VTOC)

HIPOによって記述するシステムやプログラムの機能の階層構造を表現

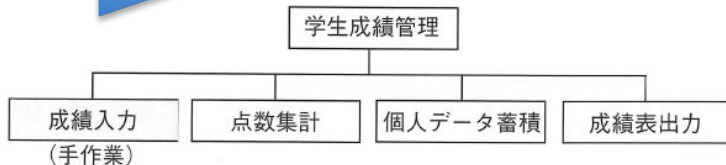
<外部設計vs内部設計作業>

作業内容	外部設計	内部設計
機能とプログラム * 1	「機能」単位でサブシステム展開	各機能を「処理」単位の「プログラム」に分割
順序	「順序付け」は行わない	「プログラム」の実行順序を定義 * 2

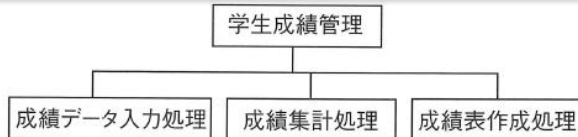
* 1

「プログラム」や「人手」等区別なし

外部設計の
サブシステム展開



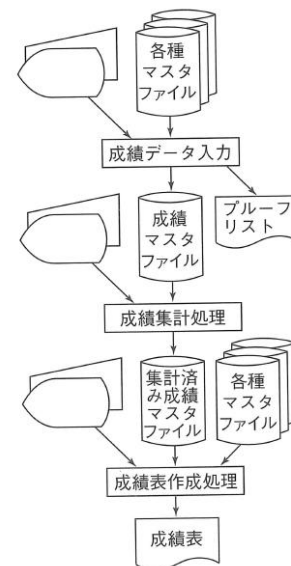
内部設計の
機能分割・構造化



「プログラム」での処理のみ

* 2

プロセスフロー



第1回

<内容>

I．科目オリエンテーション

II．2年～3年次授業「プログラム設計」「内部設計」
「内部設計演習」の復習

III．実習課題

実習課題

A情報システム専門学校の入試管理システム開発

- A情報システム専門学校の背景・環境・概要の問題点と課題をもとに入試管理システムの 外部設計の演習を設問に従って行う。
- 外部設計では以下の演習を実施する。

① 機能設計(DFD, IPO)

② ヒューマン・インタフェース設計(画面レイアウト、画面仕様、画面遷移図)

③ コード設計

④ 論理データ設計

出題先

⇒ ¥¥teasrv¥出題¥講師[Q-Z]¥山本隆之¥外部設計¥第01回¥課題¥

提出期限⇒ 授業情報を参照

提出ファイル名

2xyyzz_名前_基礎演習1DFDテンプレート.asta

2xyyzz_名前_基礎演習1-1DFDテンプレート.asta

2xyyzz_名前_基礎演習1-2IPOテンプレート.asta

2xyyzz_名前_基礎演習1-3IPOテンプレート.asta

※ 2xyyzz: クラス番号 と 名前 を必ず追記してください

提出先

⇒ ¥¥teasrv¥提出¥講師[Q-Z]¥山本隆之¥外部設計¥第01回¥