

システム開発

システム化計画

「業務改善」 ... 業務の情報化、システム化
↓
企業が情報効率や情報活用にコンピュータの利用を促進
業務のシステム化

要件定義

確認: ベンダー (vendor)

システムの・・・

概要 (機能・性能・内容)
予算規模 (概算コスト・開発スケジュール・開発体制)
運用サポート体制

などを決める・明確にする
注意) 「要件」いろいろ!



情報提供依頼 (RFI) ... システム化を検討しているので
関連する情報が欲しい。

提案依頼書 (RFP) ... こういう目的でこういう機能をもつシステムを
導入することにした。
その具体的な提案をして欲しい。
(要件定義ができていないと依頼できない)

提案書 ... こんなシステムはどうですか？

見積書 ... これくらいの費用がかかります。



SLCP (ソフトウェア・ライフサイクル・プロセス)

→ 「共通フレーム」

→ 企画から運用までの開発作業内容の詳細化・明確化

情報システムの開発

→ 代表的なソフトウェア開発手法・考え方

- ・ 主なソフトウェア開発手法 ← システム設計のモデル
 - ・ プロセスモデル(構造化手法)
 - ・ データモデル
 - ・ **オブジェクト指向モデル**
- ・ 主なソフトウェア開発モデル
 - ・ **ウォーターフォールモデル**
 - ・ **プロトタイプングモデル**
 - ・ **スパイラルモデル**
 - ・ RAD (Rapid Application Development)
 - ・ SLCP (Software Life Cycle Process)

関連) 最近の開発手法

- ・ **アジャイル**
 - ・ 「素早い」の意。短期間に迅速に開発を行う(イテレーション: 反復型開発)
 - ・ アジャイル開発の代表的な手法
 - ・ スクラム
 - ・ エクストリームプログラミング(XP), ペアプログラミング
- ・ **リーン**
 - ・ 「痩せた」の意。無駄なく、小さく始める



1) 構造化手法

- 構造化(ストラクチャード)定理
処理を分割し、「**適正プログラム**」の組み合わせで記述
↓
「順次」「繰返し(ループ)」「選択(分岐)」

2) オブジェクト指向

オブジェクト指向(object oriented)

オブジェクトに重点をおいた考え方, オブジェクトを組み合わせでシステムを構築

- データ・手続きなどの集合
- 複雑な仕組みを隠蔽・カプセル化

Java, C++ などモダンな言語が対応

例) テレビや車などの操作例

関連) 文書化・ドキュメント化の重要性

→ プログラミングにおけるコメントなども

情報システムの開発の具体例

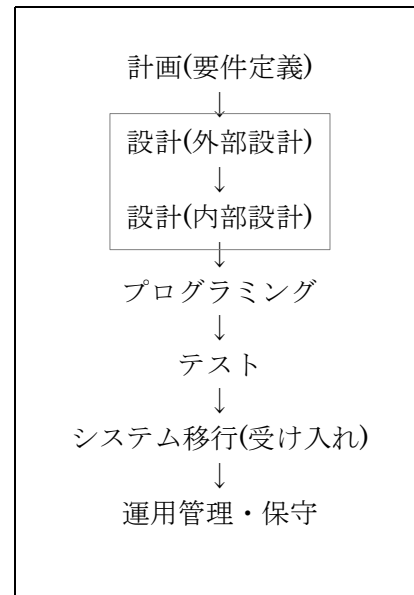
EUC で関与する部分

- ・ 基本的な設計部分
- ・ インターフェース (利用方式, 使い方)
- ・ テスト
- ・ 運用管理

システム開発の方法

工程にかかる時間と完成度の点から

- ・ **ウォーターフォールモデル**
 - ・ 管理しやすい
 - ・ ユーザの要求を反映しにくい
- ・ **プロトタイプモデル**
 - ・ 管理しにくい (費用, 時間)
 - ・ ユーザの要求を反映しやすい
- ・ **スパイラルモデル**
= ウォーターフォール + プロトタイプ



システム開発費の見積もり

ファンクションポイント法 (FP 法)

ソフトウェアの規模をソフトウェアの持つ機能から割り出す



関連) 「人月」の神話

関連) 日程管理と品質管理

設計

- ・ 外部設計
 - 画面概要設計
- ・ 内部設計
 - 詳細設計
- ・ プログラム設計
 - プログラム内の構造設計

プログラミング

- アルゴリズム
- データ構造
- プログラミング言語

テスト(検証)

- 設計通り動作するか？
- 十分な性能がでるのか？
- 負荷をかけた場合に不都合が生じないか？
- 実際に仕事で利用した場合はどうか？

作ったものはテストして動作確認しないと実際に使えない！

テストの種類

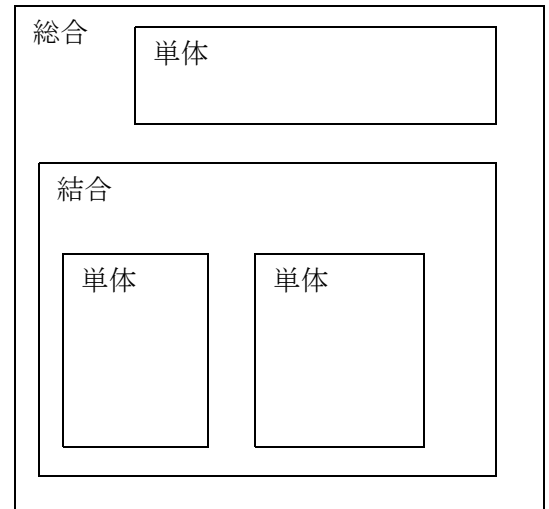
単体テスト, 結合テスト



総合テスト(=システムテスト)

テストで関係する部分の大きさ

単体テスト << 結合テスト << 総合テスト



ホワイトボックステストとブラックボックステスト

退行テスト(回帰テスト)

- デグレードしていないか？修正した部分が別の違うミスを作っていないか？

ポイント)

プログラマ, 単体テスト, ホワイトボックステスト



シスアド(エンドユーザ), 総合テスト, ブラックボックステスト

運用テスト

- システムの反応(負荷・信頼性)を見るテストのポイント
 - ・レスポンスタイム
 - ・ターンアラウンドタイム
 - ・スループット
 - ・ベンチマークテスト
 - ・RAS, RASIS

信頼度成長曲線(ゴンペルツ曲線)

- テストをいつまで続けなければならないのか？
 - エラー数がある程度落ち着くまで！

テスト用のデータ作成

- 限界値分析, ニューメリックチェックなど

検収・納品・移行

運用・保守

→ PDCA, QC

→ バックアップ

→ マニュアル管理