


K●GEIコンピュータ応用学科




## 授業予定(4回分)

9/22 情報システムの開発プロセスとそのモデル  
 9/29 システムの運用と管理の方法について  
 10/6 プログラム言語の概要、プログラム開発の手順  
 10/13 アルゴリズムについて+まとめのテスト

10/20~11/24 「コンピュータの構成」など(行谷)  
 12/1~12/22 「オペレーティングシステム」など(荒井)  
 1/12~ 1/26のいつか? 定期試験

K●GEIコンピュータ応用学科



## 第1回:システム構築(1)

- 情報システムの開発プロセスとそのモデルについて
- 小テスト

K●GEIコンピュータ応用学科

## システム開発プロセス

- 要件定義(基本計画):情報システムが満たすべき要件(システム要件仕様)を決める
- 外部設計
- 内部設計
- プログラム開発
- システムテスト
- システム導入

※上記の順に開発を進める開発プロセスを **ウォーターフォール型**という

K●GEIコンピュータ応用学科

## 要件定義(基本計画)(1)

- (前提)情報システムは利用者のニーズに応えなければならない
- このニーズを満たすため**システムから見たとき**にどのような**機能**を有すべきか?を明確にする
- 業務分析(業務同士の関連、個々の業務が使用するデータ、業務のやり方の変更の有無)を通じて、個々のプロセスとデータを明らかにし、モデルを作成。**プロセスモデルとデータモデル**。

K●GEIコンピュータ応用学科

## 要件定義(基本計画)(2)

- プロセスモデルとデータモデルは互いにその機能を補完し合う面を持っている
- こうした分析、モデルに制約条件を加味して、**システム要件定義書**を作成する

K●GEIコンピュータ応用学科

## 外部設計とデータベース設計

- ユーザ側の視点で
  - システムと利用者(ユーザ)との間の**インターフェース**(システムの使い方)を設計
  - **システム機能設計**:  
プロセスモデルを詳細化→基本機能に展開
  - データ入出力の設計
  - データモデルを詳細化し、**論理データベース**(利用者から見たデータベース)を設計
  - **外部設計仕様書**を作成

K<sup>o</sup>GEI コンピュータ応用学科

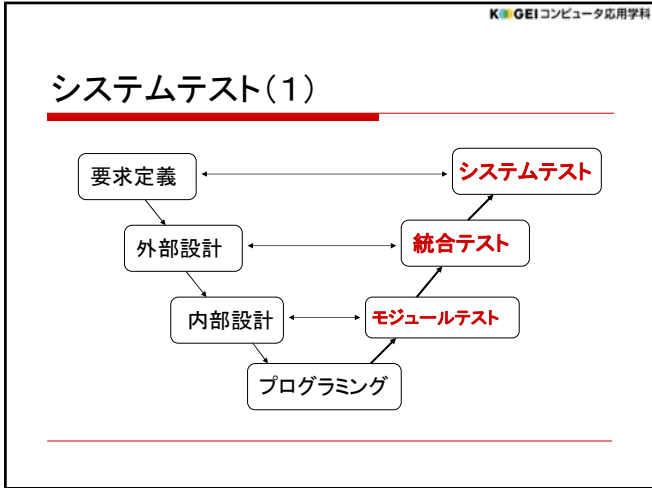
## 内部設計

- 開発者側の視点で
  - 外部設計仕様書に記述されたシステム機能設計と入出力設計からプログラム設計を行い、**プログラム仕様書**を作成する
  - 外部設計仕様書に記述された論理データベース設計から**物理データベース**を作成する  
(データ検索方式、データ管理方法、セキュリティ)
  - **内部設計仕様書**(プログラムとデータベースの内部構造の設計書)を作成

K<sup>o</sup>GEI コンピュータ応用学科

## プログラム開発

- プログラム仕様書にのって、**コーディング**を実施(プログラム言語を用いてプログラムを書くこと)
- プログラムは単体、および、モジュール毎(内部設計で決められたプログラム群の単位)にテストする
- 成果物は**プログラム**



K<sup>o</sup>GEI コンピュータ応用学科

## システムテスト(2)

- 単体テスト(モジュールテスト)
  - **ホワイトボックス法**: 内部動作まで正否を検証
- 結合テスト(統合テスト)
  - 結合テスト以降は**ブラックボックス法**: 入力に対する出力結果だけで判断
- 総合テスト
- 承認テスト
- 運用テスト

開発部門で実施

システムテスト: 開発部門+シスアド担当

K<sup>o</sup>GEI コンピュータ応用学科

## 結合テストの手法

- トップダウンテスト
  - 最上位のモジュールから順次下位へ
- ボトムアップテスト
  - 最下位のモジュールから順次上位へ
- サンドイッチテスト(折衷テスト)
- ビックバンテスト
  - モジュール単体のテストをすべて終えてから、モジュールをすべて一度に結合してテスト

K<sup>o</sup>GEI コンピュータ応用学科

## テストの目的

- システム導入後のトラブルの防止のため、テスト段階でできるだけ多くの**バグ**を**発見**して修正しておくことが目的である

累積バグ件数

信頼度成長曲線(ゴンペルツ曲線)  
 $y = c \exp(-a \exp(-bx))$

テスト項目消化件数

こうならないときは、バグ出しが適切になされていない可能性が高い!!

## システムテスト(3)

- (狭義の)システムテスト  
(総合テスト+承認テスト+運用テスト)
  - 実際の運用を想定しての模擬使用
  - データランザクション(データの作成、更新、消去)を大量に発生させてテスト
  - 応答時間、異常時からの回復処理もテスト
  - 導入後もしばらくは継続することが多い(運用テスト)

## 総合テストの中身

- 機能テスト
- 操作性テスト
- 性能テスト: 処理能力(スループット)や応答時間(レスポンスタイム)  
ベンチマークテスト、ターンアラウンドタイム
- 負荷テスト: データ量などの**負荷を変化**させて
- 異常時テスト: エラー処理や例外処理をテスト
- レグレッションテスト(退行テスト):  
システムの修正・新機能の追加をしたときにほかの部分に影響が及んでいないかどうかをテスト

## テストデータの用意

- ブラックボックス法でテストを行うときは、用いるテストデータが重要となる
  - 同値分割  
範囲内と範囲外から**代表的な値**を使って分析することで、例えば、正しい入力データの値が1~99のとき、テストデータとして50、-10、110などを設定する。
  - 限界値分析  
テストの際、範囲内の最大値・最小値と範囲外の最大値・最小値を使って分析すること。上の例なら0, 1, 99, 100を使うことになる。

## 仮想事例による要求定義プロセス(1)

- 【桜自動車株式会社の仮想事例】(pp.103-104)
- システム開発で最も難しいのは要求定義プロセスと外部設計プロセス: 個別要素が多い!!
  - プロジェクト全体の工数(内部設計以下の後続するプロセスの作業負荷)や人員が決まってしまう!
  - 「**問題解決**に至る道筋をいかにつけるか?」が決め手  
「どこに着眼点をおいて、ものごとを捉えるか?」

## 仮想事例による要求定義プロセス(2)

- システムへの要望
  1. 顧客に、車検や定期点検などのときにタイムリーにアナウンスしたい
  2. 提携しているレンタカー会社の顧客リストも利用
  3. 各営業所のある顧客データを本社でも活用
  4. 本社でも各営業所でも共通に使えるような顧客データベースがほしい
  5. 顧客データベース構築に関する投資の効果...

## 仮想事例による要求定義プロセス(3)

- 顧客データベースを必要としていることが明白  
以下の展開を段階的に進めることが必要
  1. 販売...活動の中で顧客に関する情報を収集
  2. 情報を全社で共有するための業務の仕組みを構築
  3. カーライフ全般にわたる各フェーズで顧客が必要とするときに必要なサービスを提供
  4. こうした営業展開を進めて、収益向上を

※データベースの詳細設計へ!

## システム開発の規模と要員

---

- システム開発見積方法
    - プログラムステップ法  
開発するプログラムのステップ数で見積もる
    - ファンクションポイント法  
外部入出力の数や内部論理ファイルの数と、  
開発の難易度を掛け合わせて見積もる方法
  - 開発規模(工数)の単位は人月
    - 2人で3ヶ月かかるなら「6人月の仕事」
- 

## 次回(第2回)は

---

- システム構築の続き(入力項目など)と  
システムの運用と管理の方法について
-