

Imatec Inc.

SWEST13

## 構造化分析・設計の勘所

～強いプログラムを作るテクニックを学ぶ～

予告編

2011年9月2日

組込みソフトウェアギルド 幹事  
(株)イマテック 酒井 郁子




EMBEDDED SOFTWARE GUILD Copyright 2011 by Embedded Software Guild All rights reserved. P.1

Imatec Inc.

## 構造化設計は古の技術?

- 構造化設計とは
  - トップダウン式にモジュール構造を階層に分割する設計技法
  - 構造化プログラミングの考え方がベース
    - 構造化プログラミングの効果
      - コードを読みやすくする
      - 複雑さを軽減する
- 前提:仕様(なすべきこと)が定義されている

古い技術だね。今も役にたつの？



設計技法

1970年 1980年 1990年 2000年

プロセス

1970年: 構造化プログラミング

1980年: 構造化設計, 構造化分析

1990年: インタフェース中心設計, データ中心設計

2000年: オブジェクト指向分析設計

ウォーターフォール開発プロセス


反復型開発プロセス

アジャイル開発プロセス


EMBEDDED SOFTWARE GUILD Copyright 2011 by Embedded Software Guild All rights reserved. P.2

Imatec Inc.

## 構造化分析・設計の参考文献




G.J. BP 1979年



BP 1994年

現在入手可能な本も意外と少ない



SESSAME WG2 2006年


EMBEDDED SOFTWARE GUILD Copyright 2011 by Embedded Software Guild All rights reserved. P.3

Imatec Inc.

## ソフトウェア設計のポイント

1. わかりやすい境界で、処理を分割する
  - 複雑なシステムを管理しやすい形態に分割し、整理する
    - 分割したものは、それぞれ1モジュールとしてプログラム・コード上に実現
2. システムを階層構造で組み立てる
  - 大きな粒度と、小さな粒度の塊で、上位・下位の関係の階層構造をつくる
3. 各モジュールの独立性を高める
  - 独立性を高めることによる効果
    - モジュール内の処理を変更しても、ほかのモジュールに影響しない
    - モジュールをブラック・ボックスとして再利用できる

プロのソフトウェアは動けば良いものではない！  
“使える”ソフトウェアにするためには、  
これらポイントを、しっかり押えてくれたまえ。



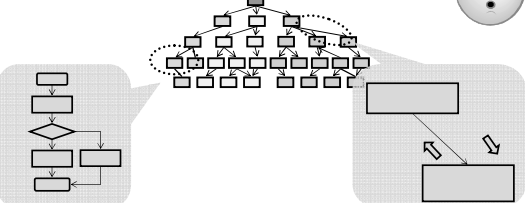
EMBEDDED SOFTWARE GUILD Copyright 2011 by Embedded Software Guild All rights reserved. P.4

Imatec Inc.

## 構造化設計の三大成果物

- プログラムの階層的モジュール構造の定義
- 各モジュールの機能定義
- モジュールインタフェースの概略的な定義

あっ、見たことある図だ

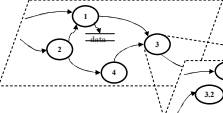


EMBEDDED SOFTWARE GUILD Copyright 2011 by Embedded Software Guild All rights reserved. P.5

Imatec Inc.

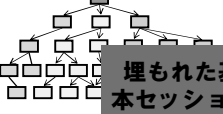
## 構造化設計の知識、使えていますか？

- DFDによる構造化分析



構造化設計習った気がする  
これらは、いつ使えばよかったの？

- 構造化設計による構造図



埋もれた基礎知識で終わらせないよう  
本セッションと一緒に考えてみましょう

EMBEDDED SOFTWARE GUILD Copyright 2011 by Embedded Software Guild All rights reserved. P.6