

情報システム学会 第一回「私の主張」の会発表

## 機能記述技法の提案

～ソフトウェア産業の3K化阻止のために～

2012年11月10日  
歴史をエンジニアリングする歴史工房  
明智憲三郎  
ブログ「明智憲三郎の世界 天下布文！」  
Facebook「本能寺の変 四二七年目の真実」

1

### 目次

1. 機能記述の重要性
  - 1.1 プロジェクト崩れのパターン
  - 1.2 機能記述不備の伝搬
2. 機能記述の現状
  - 2.1 外部仕様書での位置付け
  - 2.2 認識されていない記述原則
  - 2.3 認識されていない理由
3. 機能記述技法の概要
  - 3.1 機能定義法の要点
  - 3.2 機能構造化法の要点
  - 3.3 他の技法との比較
4. 機能記述技法の普及に向けて

2

## 1. 機能記述の重要性

### 1.1 プロジェクト崩れのパターン

単体試験終了までは問題を起こしながらもなんとか進んだ。  
結合試験に入った途端に問題噴出し、工程が大幅に遅延。  
要員を増員するも問題収束せず遅延拡大。

**【私の主張】**  
この現象は40年間変わっていない。  
近年、非機能要件が注目されているが、そもそも機能要件の問題が解決されていない。

3

### 1.2 機能記述不備の伝搬

**整理不足で膨らんだ機能  
あいまいな機能  
要求を満たさない機能**

外部設計	外部仕様確認に手間取る 整理不足で共通化不十分
内部設計	仕様確認に手間取る 整理不足で共通化不十分 大量の不良の混入
製造	網羅できないテスト 大量の不良の取り残し
単体試験	大量の不良による混乱 網羅できないテスト
結合/総合試験	不良の続発による混乱
運用試験	大量の仕様問題の発生

4

## 2. 機能記述の現状

### 2.1 外部仕様書での位置付け

1. システムの概要
2. ハードウェア構成
3. システムの位置付け
4. 機能
5. 画面・帳票仕様
6. データベース仕様
7. 非機能要件

5

### 2.2 認識されていない記述原則

仕様(外部仕様)とは、機能と性能から成る  
**機能とは、入力(x)と出力(y)の変換関係(y=f(x))**  
性能とは、機能を達成する効率  
処理とは、機能・性能を実現するやり方(内部仕様)

**X** ⇨ **y=f(x)** ⇨ **Y**

**機能の定義とは、**  
全ての出力データの内容とその出力条件が  
入力データ(と定数)を用いて記述されること  
例: 画面Aの項目aがファイルBの項目bに等しい場合は、画面Cの項目cを帳票Dの項目dに出力する。

6

●機能の記述例

【機能概要】  
1ヶ月料金滞納者へは利用停止予告書を発行し、2ヶ月料金滞納者は利用停止する。

【詳細機能】(注)下線はデータベース仕様で定義済みのデータ項目  
料金滞納者ファイルの滞納開始年月がシステム年月+1である滞納者名に対して、当該者の契約者ファイルの契約者名・住所宛の利用停止予告書を印刷する。  
料金滞納者ファイルの滞納開始年月がシステム年月+2以上である滞納者名に対して、当該者の契約ファイルの利用許可に利用停止コードを書き込む。

【私の主張】  
ここで脱落した管理者・技術者が先憂後楽の代わりにプロジェクト崩れを選択している。

7

●記述の効率化・正確化の工夫  
(1)文章記述の排除(決定表や記号の使用)

	データ	ケース1	ケース2
入力	料金滞納者ファイル		
	滞納開始年月	=システム年月+1	>システム年月+1
	滞納者名	(A)	(A)
	契約ファイル		
出力	契約者名	=A	
	住所	(B)	f
	利用停止予告書	印刷	
	宛名	A	
	住所	B	
	契約ファイル		更新
	使用許可		使用停止コード

8

(2) 構成の最適化(構造化)による記述量削減

MECE: Mutually Exclusive and Collectively Exhaustive  
「相互に排他的な項目」による「完全な全体集合」

```

    graph LR
      0[0 人事] --- 1[1 厚生]
      0 --- 2[2 評価]
      0 --- 3[3 出退勤]
      1 --- 11[1.1 健康]
      1 --- 12[1.2 事務]
      2 --- 21[2.1 昇格]
      2 --- 22[2.2 処分]
      3 --- 31[3.1 報告]
      11 --- 111[健康管理]
      12 --- 121[健康診断]
  
```

9

2.3 認識されていない理由

●機能の定義が認識されていない理由

- 仕様、機能、処理の認識があいまい (外部/内部、基本/詳細の混用も)
- 定義の仕方の教科書がない
- 欧米では「機能とは $y=f(x)$ 」が自明(?)

●機能の構造化が認識されていない理由

- 構造化が下流(プログラム)を対象
- 構造の作り方の教科書がない
- 欧米では学校教育で「文書の構造化」(?)

10

●文書化を軽視する文化  
行間を読み！  
資料ばかり書いてないでプログラムを作れ！  
文書化はソフトウェア技術者の仕事ではない！

【私の主張】  
文書化技術 ≠ 日本語文章技術(テニヲハ)  
文書化技術 ≡ 設計技術

11

その結果、オフショア開発先からの指摘  
@IT「いいかげんにして！日本企業」

日本人発注者への不満  
仕様をまとめる能力が不足  
が圧倒的に多い。具体的には、

- 要件の網羅性が悪く、論理的にすっきりまとまった資料が少ない
- 「重複なく・漏れなく」考える論理的思考(MECE)に欠ける仕様書が目立つ
- 共通化すべき事項が設計書に点在する
- 画面間の関連性があいまいなため、全体の業務フローが把握しにくい

12

一方で日本側の理解は？ 【IT人材白書2012】  
IPA2010年度の257社アンケート調査結果

- コスト増加要因(国内発注に比し)
    - 1位 仕様に関する説明、合意形成 56%
    - 2位 詳細な仕様書の作成 44%
  - トラブル発生要因
    - 1位 コミュニケーション不足(言語・文化)70%
    - 2位 発注仕様に対する理解不足 44%
- これに対して、ソフトウェア業界の取り組みは？  
⇒ 「現地会社にブリッジSEの育成・常設」
- 【私の主張】 国内問題を放置して国外の設計力を強化  
→ 3K化への道を着実に辿る

### 3. 機能記述技法の概要

- 3.1 機能定義法の要点
- 文章記述を極力減らし、正確に機能を記述でき、かつ、ユーザーにも理解できる表記方法を定める。
- ・決定表の活用
  - ・箇条書きの活用
  - ・条件表記の記号化(等号・不等号など)
  - ・規定外入力への扱いの網羅
  - ・禁止事項の明示(手順的記述、未定義データの使用、「など」の使用)
  - ・チャンキングの活用

14

#### 3.2 機能構造化法の要点

設計原理は「上位要素を目的として、それを実現する手段を下位に展開した系統図を作り、結果として重複なく・漏れなく(MECE)、かつ共通要素がまとめられた関係を作る」と定める。

設計結果の評価基準は「モジュールの強度・結合度」をアレンジして設定する。たとえば、

- 機能的強度 例: 年間計画書を作成する機能
- 手順的強度 例: 画面表示して帳票出力する機能
- 暗号的強度 例: その他機能

構造の作り方は？  
ソフトウェア技術分野に良い教科書がない！

15

そこで、文書の構造化技術(ロジカル・ライティング)を参考にして構造の作り方を整理する。

- 文書の構造化技術の参考例
- 『理科系の作文技術』木下是雄、中央公論新社、1981  
パラグラフの立て方、文の構造の作り方
- 『文書の書き方・作り方』高橋伸治 PHP研究所、1998  
文書モジュールを構成していく方法

たとえば、  
ステップ1 機能要素の抽出  
業務フローを参考にして書き出した機能項目を整理して、ただ1つの目的をもった機能を実現するように機能要素を抽出する。この機能要素に適切な機能名称を付ける。

16

#### ステップ2 機能構成図の作成

機能要素を階層構造に整理する。上位と下位との関係性を点検し、必要ならば機能要素の再編成や機能名称の変更を行う。原則として下位の機能要素の数は7個以内にする。整理した結果は機能名称を使った構成図に書く。構成図に書いた構成がそのまま機能を記述するドキュメントの章・節・項に対応する。

野球の「ゴロをとる技術」と同様に原理は簡単だが実践は難しく、奥行きのある技術。  
したがって、良い例／悪い例、演習問題を多数そろえる必要がある。

17

#### 3.3 他の技法との比較

- (1)機能定義法を形式手法と比較  
目標が明快で、習得・実用が容易  
ユーザーも理解しやすくなる  
形式手法の記述へも書き直しやすくなる
- (2)機能構造化法をオブジェクト指向設計と比較  
目標が明快で、習得・実用が容易  
従来の外部仕様書形式を継続使用できる  
オブジェクト指向設計をも補完する

18

ここで、拙著の目次を例に考えると、・・・

『本能寺の変 四二七年目の真実』

- 1章 作り変えられた歴史
  - 1.1 誰の手で史実は歪められたか
  - 1.3 作られた信長と光秀の不仲説
- 2章 謀反を決意した真の動機
  - 2.1 土岐氏再興の悲願
  - 2.4 衝撃の「家康潰し」計画
- 3章 本能寺の変は、こう仕組まれた
  - 3.1 二つの企て
  - 3.5 光秀の苦悩、そして滅亡
- 4章 新説を裏づける後日譚

19

**【私の主張】**  
設計思想や設計能力によって設計結果は各人各様。  
設計結果の良否には雲泥の差あり。

「だから、新たに設計しないでパッケージソフトウェアを利用」も一つの解だが、  
「だから、設計力を高める」解はいらないのか？  
これを放棄したら3K化の道しかないのでは。

20

- #### 4. 機能記述技法の普及に向けて
- (1) 機能記述技法の教科書の制作・出版
  - ↓
  - (2) 教育者の育成
  - (3) 情報系学校への講座設置
  - (4) 情報系企業への講座・訓練提供
  - ↓
  - (5) 学校教育での文書構造化教育実施  
欧米並みに小学校から教育・訓練
- 【私の主張】**  
普及の鍵はソフトウェア産業界の重要性認識！
- 21

● 普及による効果  
システム構築・保守・運用全般に渡る  
品質・生産性の飛躍的な向上

**【私の主張】**  
設計力の強化により  
ソフトウェア産業の3K化を  
阻止できる

22