

計算機システム概論 システム開発と運用 2011/5/18

門林雄基

NAIST 奈良先端科学技術大学院大学

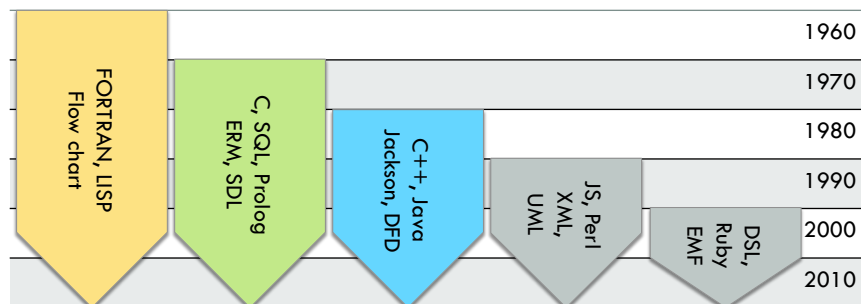
講義のポイント

2

- システム開発における設計(モデリング)とは？
- システム開発では、どんなプログラミング言語で書くのか？
- システム開発は、どのように進めるのか？
- システムの運用とは？

システム開発における イノベーションの累積

3



- ITは最新技術だが、我々は過去の設計手法、実装言語、ソフトウェア資産の上に生きている
- イノベーションの累積を活かせるようになろう

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

プログラミング言語と
モデリング手法の進化

1960年代: プログラミングの源流

5

- COBOL (The Short Range Committee, 1960)
 - 事務処理
- Fortran (John W. Backus 1957)
 - 科学計算
- Lisp (John McCarthy 1960)
 - 人工知能
- モデリング手法:
 - フローチャート (John von Neumann 1947)

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

1970年代: 今日のシステムの基礎

6

Pascal
(Niklaus Wirth 1971)

- 構造化プログラミング

```
program copy (f, g);
var f, g : file of real;
begin reset(f); rewrite(g);
  while not eof(f) do
    begin g! := f!; get(f); put(g)
  end
end.
```

Source: Pascal ISO 7185

C
(Dennis Ritchie 1973)

- 構造化プログラミング
+ 低レベルプログラミング

sh, sed, awk
(Bell Labs 1970s)

- シェルプログラミング

```
echo -n "Enter search pattern: "
read pattern
awk "/$pattern/" "{ matches++ }
END { print matches, "found" }" /path/to/data
```

Source: The GNU Awk User's Guide

SQL
(IBM 1976)

- データベースプログラミング

ML
(Robin Milner 1975)

- 関数型プログラミング

```
parent(bob, mary).
parent(jane, mary).
parent(mary, peter).
parent(paul, peter).
parent(peter, john).

anc(X, Y):-
  parent(X, Y).

anc(X, Z) :-
  parent(X, Y),
  anc(Y, Z).
```

Source: GNU-Prolog Manual

Prolog
(Philippe Roussel 1972)

- 論理プログラミング

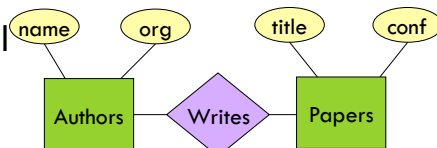
Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

1970年代のモデリング手法

7

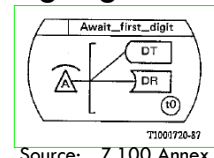
- Jackson Structured Programming (Micheal A. Jackson 1975)

- Entity-Relationship Model (Peter Chen 1976)



- SDL: Specification and Description Language (ITU-T 1976)

- .



Source: Z.100 Annex E

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

1980年代: オブジェクト指向

8

- Smalltalk (Alan Kay 1980)
 - オブジェクト指向プログラミング
 - Smalltalk仮想マシン
- C++ (Bjarne Stroustrup 1986)
 - オブジェクト指向プログラミング
 - 手続き型言語 Cとの融合
 - 機械語へのコンパイル
- Java (James Gosling 1989)
 - オブジェクト指向プログラミング
 - 広範囲にわたる標準クラスライブラリ
 - Java仮想マシン

```

(x < y) ifTrue: [
  max := y.
  i := j
] ifFalse: [
  max := x.
  i := k
]
  
```

Source: Smalltalk: A White Paper Overview

```

System.out.println("abc");
String cde = "cde";
System.out.println("abc" + cde);
String c = "abc".substring(2,3);
String d = cde.substring(1, 2);
  
```

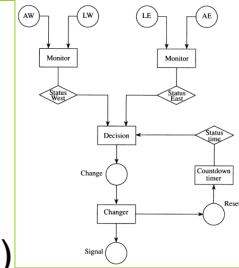
Source: Java Platform, Standard Edition 6 API Specification

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

モデリング手法(1980年代)

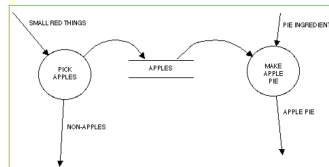
9

- Jackson System Development (Micheal A. Jackson 1982)



Source: Development Methods for Real-Time Systems

- Dataflow Diagram (Yourdon 1983)



Source: Just Enough Structured Analysis Chapter 9: Dataflow Diagrams

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

1990年代: 異種交配

10

Perl 5
(Larry Wall
1994)

- オブジェクト指向
+ シェルプログラミング

JavaScript
(Brendan Eich
1995)

- オブジェクト指向
+ Web UI

```
var httpRequest;
if (window.XMLHttpRequest) { // Mozilla, Safari, ...
    httpRequest = new XMLHttpRequest();
} else if (window.ActiveXObject) { // IE
    httpRequest = new ActiveXObject("Microsoft.XMLHTTP");
}
```

Source: MDN DOCS JavaScript

Objective Caml
(INRIA 1996)

- オブジェクト指向
+ 関数型プログラミング

XML
(Tim Bray 1998)

- 準構造化データベース
プログラミング

```
<?xml version="1.0"?>
<!DOCTYPE greeting SYSTEM "hello.dtd">
<greeting>Hello, world!</greeting>
```

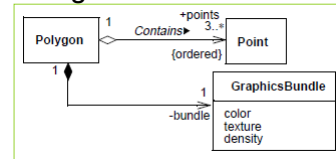
Source: Extensible Markup Language (XML) 1.0 (Fifth Edition)

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

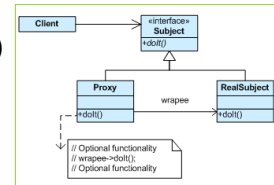
モデリング手法(1990年代)

11

- OMT: Object-modeling technique (J. Rumbaugh et al. 1991)
- OOSE: Object-oriented software engineering (Ivar Jacobson in 1992)
- Booch method (Grady Booch 1993)
- UML: Unified Modeling Language 1.1 (The UML Partners 1997)
- Design Patterns (Erich Gamma et al. 1994)
- XML Schema (W3C 2001)
- RDF: Resource Description Framework (W3C 1999)



Source: UML Notation Guide version 1.1



Source: SourceMaking Design Patterns

Data modeling languages

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

2000年代: スクリプト言語

12

- Domain Specific Languages (circa 2003)
 - 業務ドメイン記述からのプログラム生成
- JavaScript (ECMAScript, circa 2007)
 - オブジェクト指向 + 関数型プログラミング + Webプログラミング
- Python, Ruby
 - オブジェクト指向 + 関数型プログラミング + シェルプログラミング
- SPARQL (W3C 2008)
 - データベースプログラミング + Webプログラミング

```

class Post < ActiveRecord::Base
  validates :name, :presence => true
  validates :title, :presence => true,
            :length => { :minimum => 5 }
end
  
```

Source: Rails Guides, Getting Started with Rails

```

PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { <http://example.org/book/book1> dc:title ?title }
  
```

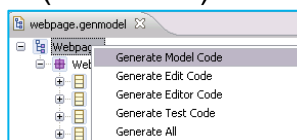
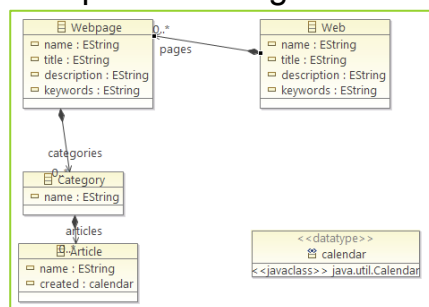
Source: SPARQL Query Language for RDF

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

モデリング手法(2000年代)

13

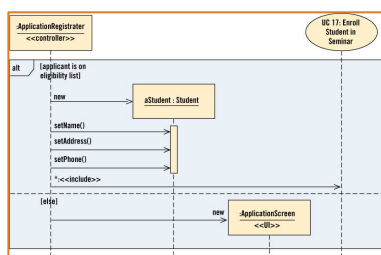
□ Eclipse Modeling Framework (IBM 2003)



Source: EMF Tutorial, Lars Vogel

□ UML 2.0 (OMG 2005)

Source: Dr. Dobbs Journal, Feb 2004



Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

2010年代: ポストモダン開発

14

- 目的に応じたモデリング手法・実装言語・ライブラリを選択する能力が求められる
- 例)
 - ユーザーインターフェース: JavaScript
 - ビジネスロジック: Ruby
 - 行列計算ライブラリ: C++
- もはや1からプログラムを書くわけではない
 - ソフトウェア部品再利用は常識に (ライブラリ、フレームワーク等)
 - プログラミング言語の文法に加えて、語彙も重要に

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

ソフトウェア層 vs 設計方法論 vs 実装言語

15

	設計方法論	実装言語
アプリケーション		
ミドルウェア		
OS		
ハードウェア		

- OS 向けの実装言語で Web サイトは作らない。
- 業務プログラム向けの設計方法論で OS は作らない。
- 雑誌や Web メディアには一番簡単どころしか載っていない。
- 雑誌や Web メディアだけでは正しい技術選択は無理。
- まじめにシステム開発を志すなら、在学中にグレートブックを 10 冊は読破しよう。
- 現場に入ったら 1 から基本を勉強しなおす暇はない。

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

システム開発の方法論

システム開発の方法論

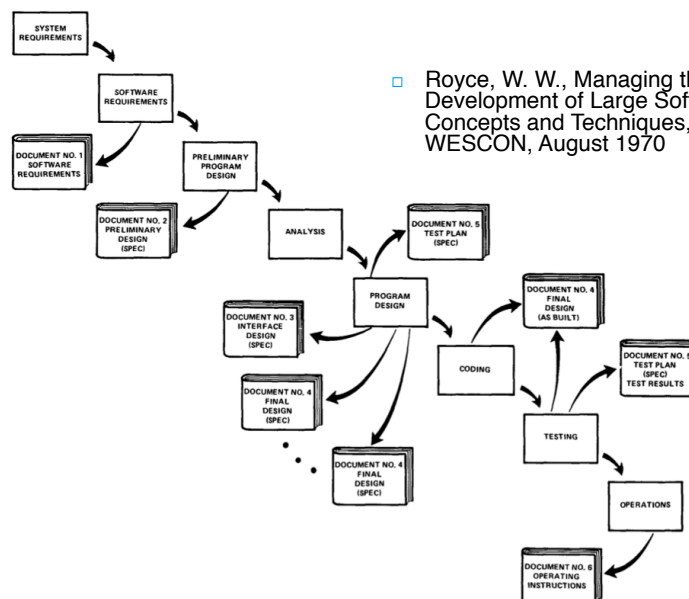
17

□ Waterfall model	1970s
□ Spiral model	1980s
□ Iterative and incremental development	1990s
□ Agile development	2000s

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

Waterfall model

18



Waterfall model

19

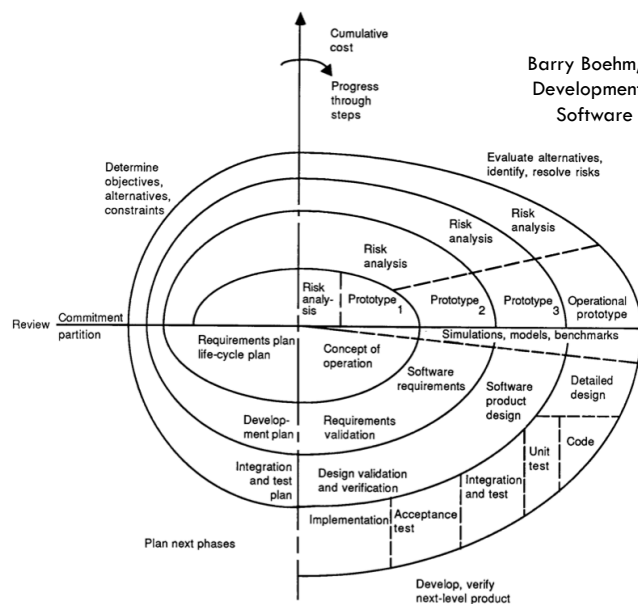
- 時代背景
 - ▣ 計算機の時間単価が人件費の数十倍～数百倍
 - ▣ 事前計画が必須
- 欠点
 - ▣ 文書中心
 - ▣ 開発が遅い、高つく

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

Spiral model

1980年代: Waterfall model の失敗が明確に意識される

20

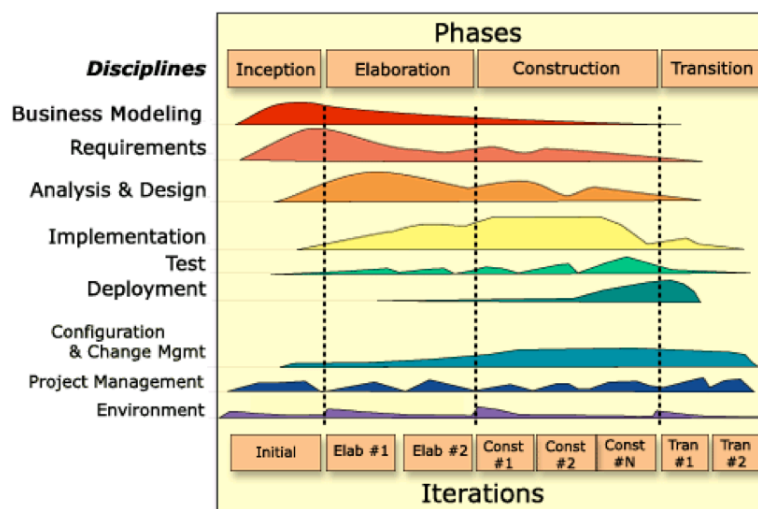


Barry Boehm, A Spiral Model of Software Development and Enhancement, SIGSOFT Software Engineering Notes 11, No. 4, August 1986.

Iterative and incremental development

例 : Rational unified process (Rational 1998)

21



Source: The Rational Edge, November 2003

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

Agile development

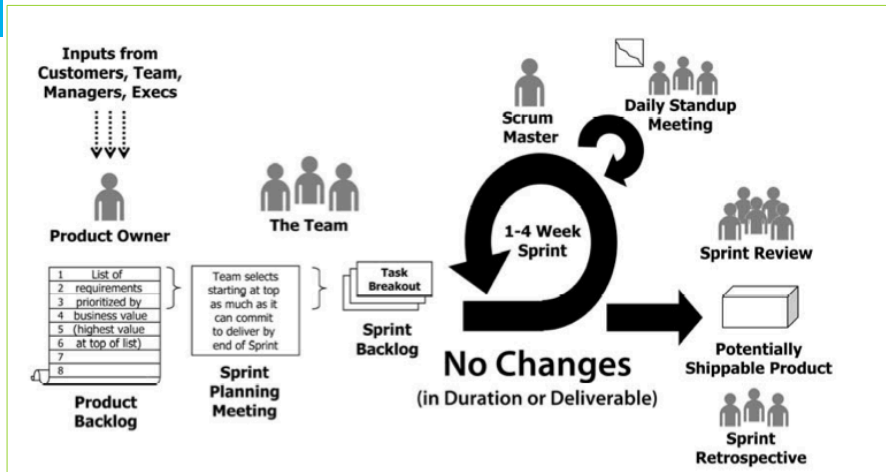
22

- Agile manifesto (2001) 発表後にそう呼ばれるようになった
 - Rational unified process も含まれる
- Agile development
 - A. Cockburn, "Agile Software Development", Addison-Wesley, 2001.
- XP
 - K. Beck, "Extreme Programming Explained: Embrace Change", Addison-Wesley, 2000.
- Scrum
 - 発祥は竹内弘高・野中郁次郎による日本の製品開発のケーススタディ (1986)
 - Pete Deemer and Gabrielle Benefield, "SCRUM PRIMER: An Introduction to Agile Project Management with Scrum", available online.

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

Scrum

23



Pete Deemer and Gabrielle Benefield, "SCRUM PRIMER: An Introduction to Agile Project Management with Scrum", available online, 2007.

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

方法論に溺れる事なかれ

頭でっかちの罠

24

- プロセス偏重
 - Robert Schafer, "Deeper Questions: The Metaproblem of Large Organizations Developing Complex Systems and the Limits of Process", SIGSOFT Software Engineering Notes, Volume 30 Issue 4, July 2005.
- モデル偏重
 - A. E. Bell, "Death by UML Fever", ACM Queue Vol.2, No.1, March 2004.
 - A. E. Bell, "UML Fever: Diagnosis and Recovery", ACM Queue, Vol.3, No.2, March 2005
- アーキテクチャ偏重
 - J. Spolsky, "Don't Let Architecture Astronauts Scare You", April 2001, available online.
- ツール偏重
 - CASE?

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

バランスが大切

25

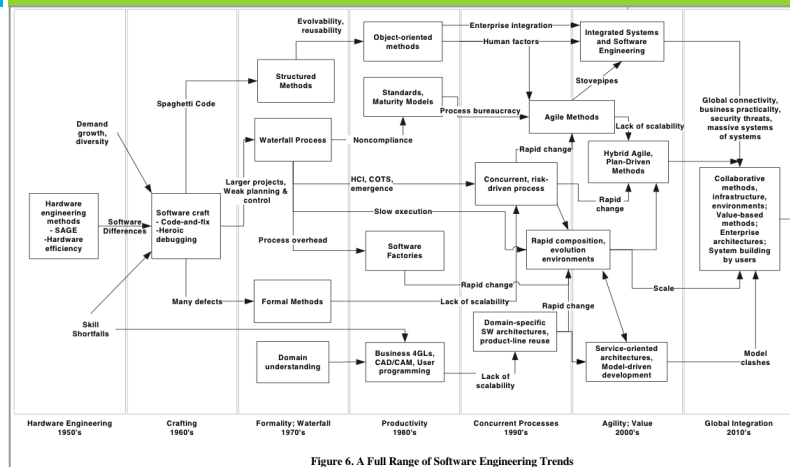
プロセス



Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

システム開発方法論の変遷

26



Source: B. Boehm, A View of 20th and 21st Century Software Engineering, ICSE'06

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

有効性は変わらない:テスト

27

- Unit test: 関数などの小さな単位で期待通りに動くことをテスト

関連する標準: IEEE 829, IEEE 1008

- Regression test: プログラム修正によるバグ発現の有無をテスト

例: Mozilla Tinderbox

- Fuzz test: 入力に異常データを混入させ、バグ発現の有無をテスト

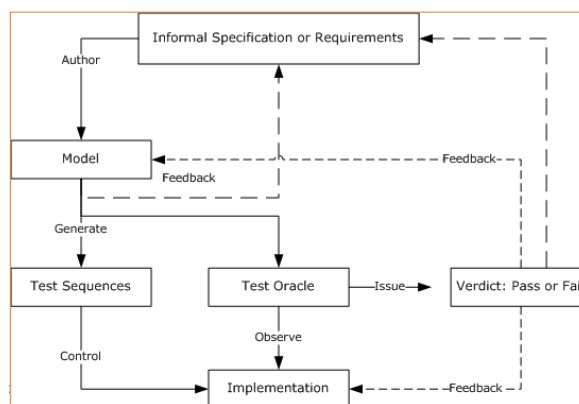
M. Sutton, "Fuzzing, Brute Force Vulnerability Discovery", Addison-Wesley, 2007
 ファジング:ブルートフォースによる脆弱性発見手法,毎日コミュニケーションズ

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

バグの根絶への取り組み

28

- モデルベース検査 (Model-based testing)
- 実装とは独立してモデルを記述し、モデルから検査手順・結果を導出することでテスト生成の自動化、網羅性の向上を目指す
- → IT-Spiral
 先端ソフトウェア工学I



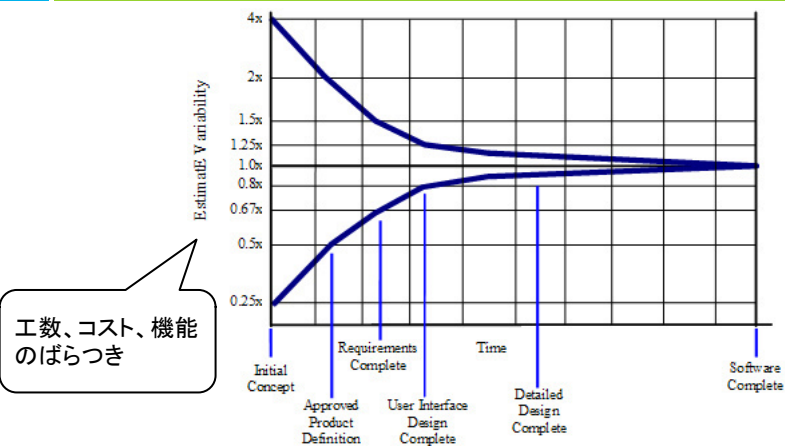
Source: Model-Based Testing,
 MSDN Library

ソフトウェア開発工数の見積り

The cone of uncertainty

不確実性の円錐

30



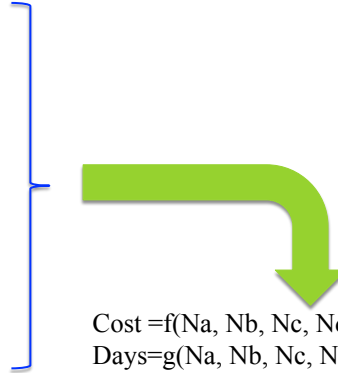
Source: <http://www.construx.com/Page.aspx?cid=1648>

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

計測可能な要素数からの推定

31

- 機能
- ユースケース
- ストーリー
- Webページ
- DBテーブル
- クラス
- レポート
- ダイアログボックス
- テストケース
- ...



$Cost = f(Na, Nb, Nc, Nd, Ne, Nf, Ng)$
 $Days = g(Na, Nb, Nc, Nd, Ne, Nf, Ng)$
 過去データにもとづくパラメータ推定

様々な見積り手法: Cocomo II, Function Point, SEMA

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

ファンクションポイント法

32

- External inputs
 - 入力するフォーム、ダイアログボックス、制御信号など
- External outputs
 - 生成する画面、レポート、グラフ、制御信号など
- External queries
 - データ検索の入出力
- Internal logical files
 - このプログラムが制御するファイル
- External interface files
 - 他のプログラムが制御するファイル

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

ファンクションポイント法

ISO/IEC 20926 Software and systems engineering -- Software measurement --
IFPUG functional size measurement method

33

計測した数に複雑度に応じた乗数をかけて総和を求める

Complexity	Low	Medium	High
External Inputs	x3	x4	x6
External Outputs	x4	x5	x7
External Queries	x3	x4	x6
Internal Logic Files	x4	x10	x15
External Interface Files	x5	x7	x10

Source: Steve McConnell, Software Estimation: Demystifying the Black Art, Microsoft Press

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

システムの運用

システム運用への要請：COBIT

Control Objectives for Information and related Technology

35

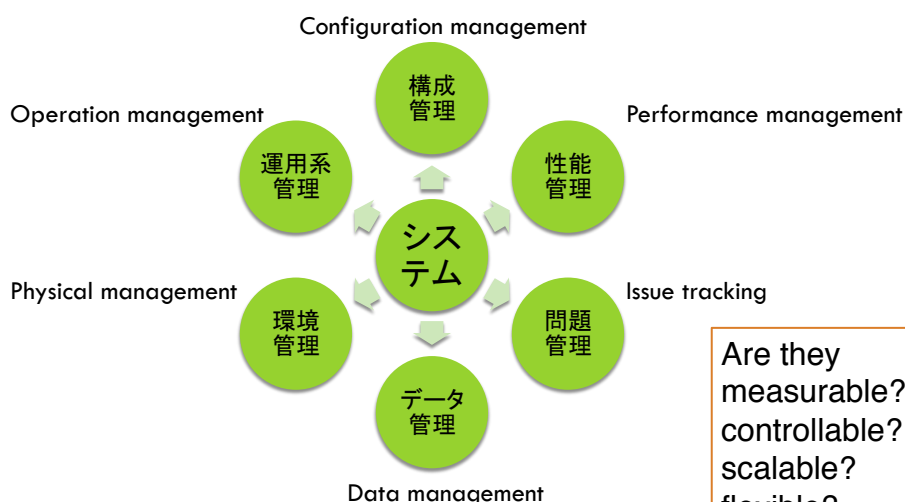
- DS1 サービスレベルの定義と管理
- DS2 サードパーティのサービスの管理
- DS3 性能とキャパシティの管理
- DS4 継続的なサービスの保証
- DS5 システムセキュリティの保証
- DS6 費用の捕捉と配賦
- DS7 利用者の教育と研修
- DS8 サービスデスクとインシデントの管理
- DS9 構成管理
- DS10 問題管理
- DS11 データ管理
- DS12 物理的環境の管理
- DS13 オペレーション管理

Source: COBIT 4.1日本語版,
日本ITガバナンス協会

<http://www.isaca.org/Knowledge-Center/cobit/Pages/Downloads.aspx>

システムの運用

36



Are they
measurable?
controllable?
scalable?
flexible?

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18

参考図書

日英併記

37

- E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley
- オブジェクト指向における再利用のためのデザインパターン、ソフトバンククリエイティブ
- K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley
- XPエクストリーム・プログラミング入門－変化を受け入れる、ピアソンエデュケーション
- Steve McConnell, Software Estimation: Demystifying the Black Art, Microsoft Press
- ソフトウェア見積り -- 人月の暗黙知を解き明かす、日経BPソフトプレス
- Per Kroll, Bruce Mac Isaac, Agility and Discipline Made Easy: Practices from OpenUP and RUP, Addison-Wesley
- アジャイル開発の6原則と20のベストプラクティス、エスアイビーアクセス
- Rick D. Craig, Stefan P. Jaskiel, Systematic Software Testing, Artech House
- 体系的ソフトウェアテスト入門、日経BP出版センター

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/18