SEA西安フォーラム2012

数学を応用する高信頼性開発

富士通株式会社・法政大学客員准教授　平田貞代

サマリー

巨大化で複雑化する情報を扱う情報システムの
信頼性に関する要求はますます高まっている。

もはや従来通りの“有識者に依存するレビュー”や
“テストに依存する修正”では信頼性の維持は難しい。

集合論や論理演算といった伝統的な数学を応用し
“社会”と“情報”の本質的な関係性を捉え、
情報システムを開発する前から
正確に検証する開発技法について、研究と実用化に取り組む。

ソフトウェア開発における問題の先送り

問題の7割は要件や仕様につくられ、5割がテストまで見逃されている

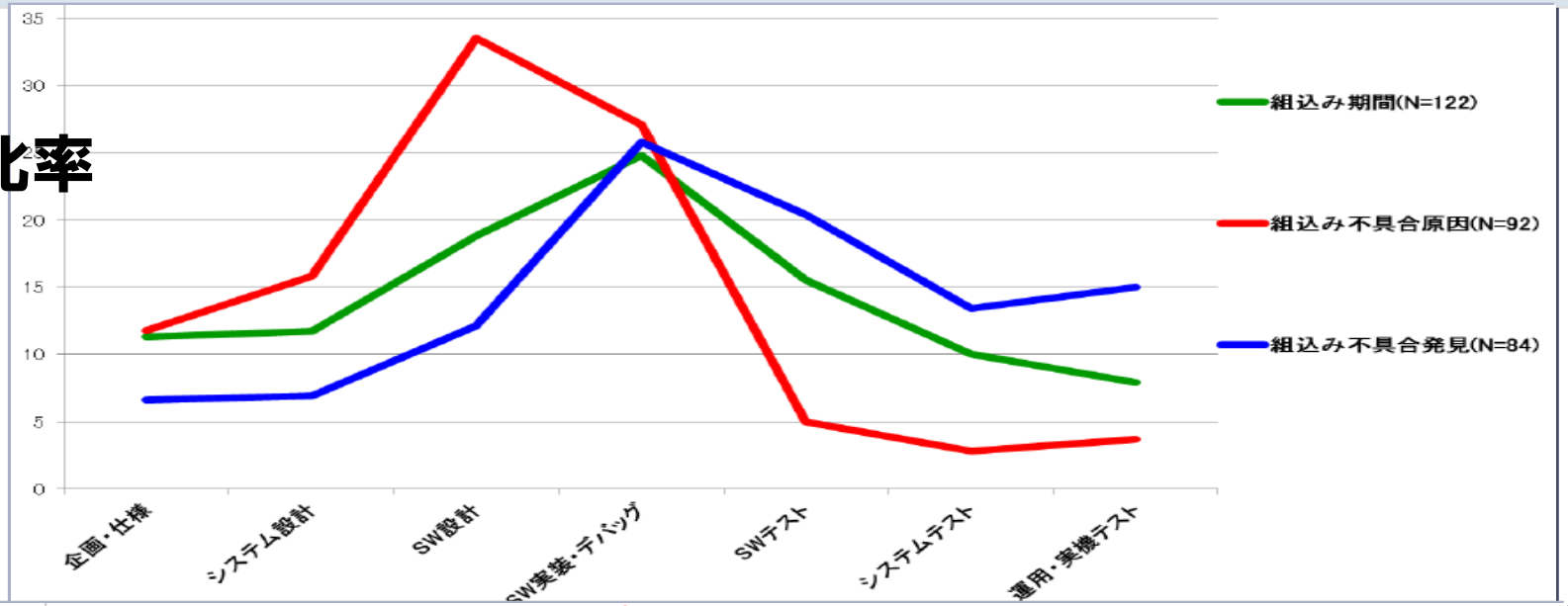
Table 5-3. Example of the Frequency (%) of Where Errors Are Found, In Relationship to Where They Were Introduced

Where Errors are Introduced (%)	Where Errors Are Found					
	Requirements Gathering and Analysis/ Architectural Design	Coding/ Unit Test	Integration and Component/ RAISE System Test	Early Customer Feedback/ Beta Test Programs	Post-product Release	Total
Requirements Gathering and Analysis/Architectural Design	3.5	10.5	35	6	15	70
Coding/Unit Test		6	9	2	3	20
Integration and Component/RAISE System Test			6.5	1	2.5	10
Total	3.5	16.5	50.5	9	20.5	100%

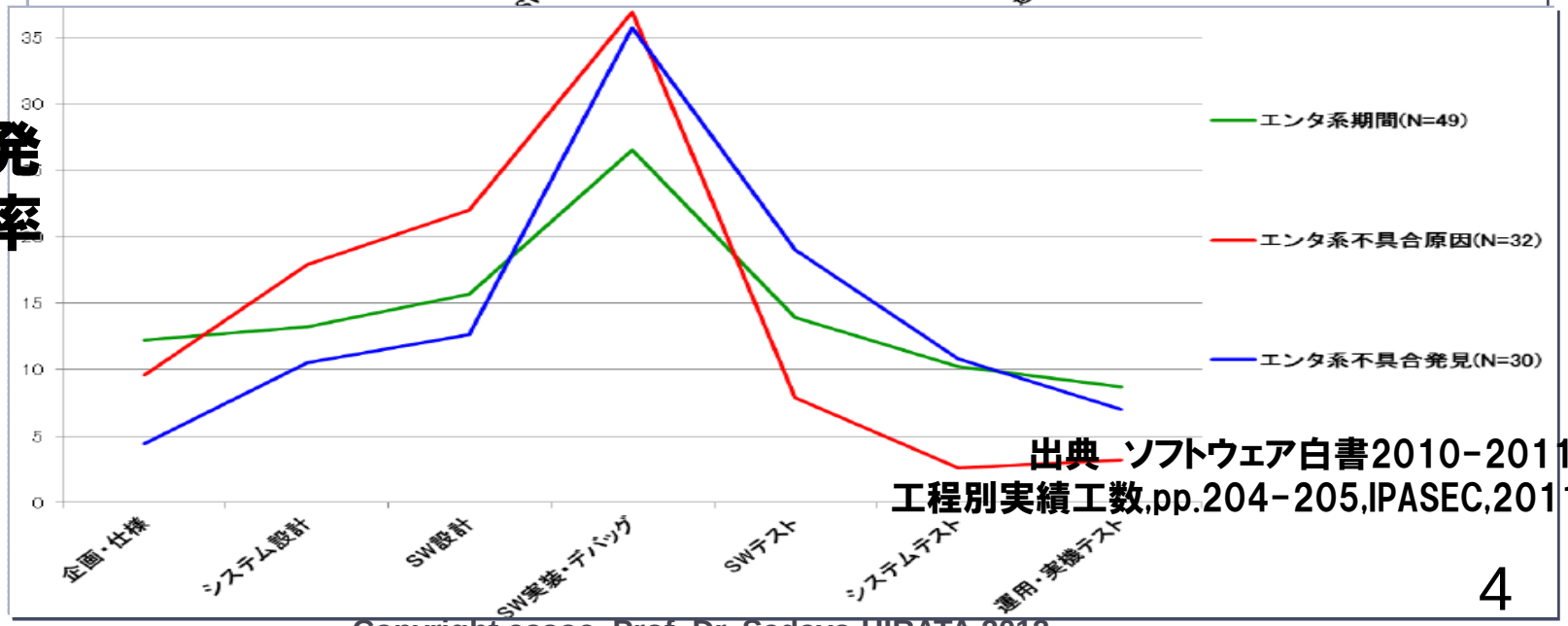
出典　新谷,形式手法の勧め,IPASEC,2011

問題を先送りする業務系開発

組込開発
問題検出比率



業務系開発
問題検出率



出典　ソフトウェア白書2010-2011,
工程別実績工数,pp.204-205,IPASEC,2011

テスト実施以前の検証技術

テストはバグが有ることを検証するが、
バグが無いことは検証できない
（ダイクストラ）

テスト実施以前の検証は
有識者の勘による
ドキュメントのレビューに依存。
他に方法は無いの？
（発注企業）

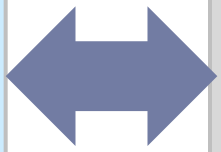
どんな優秀な人でも
ドキュメント無しに
合意や分業はできない
（パーナス）

要求や仕様の段階で科学的に検証する技法のひとつとして
「形式手法」が注目されている

要求・仕様における“作業の目的化”現象

【理想】

要求・仕様の漏れ・曖昧・矛盾を予防したい



【現場観察から得られる実態】

どの業務にも対応できるよう、あらゆる情報を集約して標準化したドキュメント・フォーマットやチェックリスト。それらの空欄を埋めるために、思いつく限りのID、カード番号、製品番号といった幾つものKeyを示し、その組合せを説明するための、膨大な帳票を作成。

シーケンス図を完成するために、順序を限定する必要がない振る舞いにも順序を示す。膨大な枚数のシーケンス図を作成。

膨大なドキュメントの判断や修正に限界

要求・仕様の本質だけを漏れ・曖昧・矛盾なく記述する技法が必要

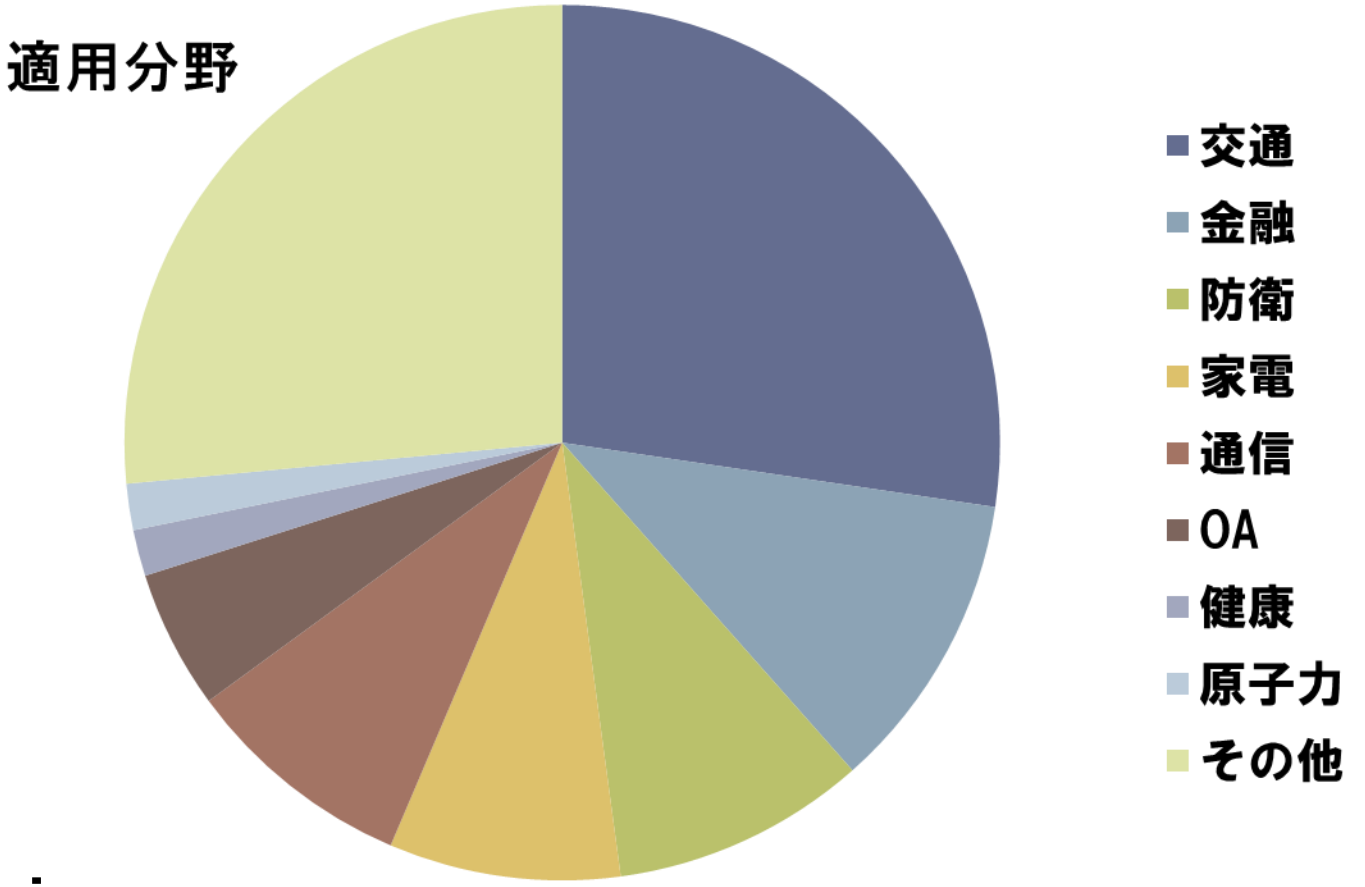
形式手法とは

言葉や図の代わりに、
And、Or、Not、->、Exist、Allなどの論理演算、
集合や写像などの離散数学、
といった“形式”を用いて表現。

基となる数学	数学の効用	機能	開発にとってのメリット
論理演算	真か偽のいずれかを明白に算出	無矛盾	・コンパクトな実装 ・属人性の排除 ・常に同じ結果の導出が可能 ・合意,意思決定の迅速化 ・問題の潜む範囲,修正箇所の限定 ・チームワーク
集合	要素を隙間なく重なりなく明確に分類	抽象化	
写像	必ず唯一の要素を示す	一義性	

形式手法の適用箇所

【適用分野】



【適用目的】

- Inspection
- Static Analysis
- Testing
- Model Checking
- Proof

出典　ラーセン,形式手法98導入事例の調査・分析から見る高信頼性ソフトウェア開発の現状,IPA,Oct.2012を基に編集

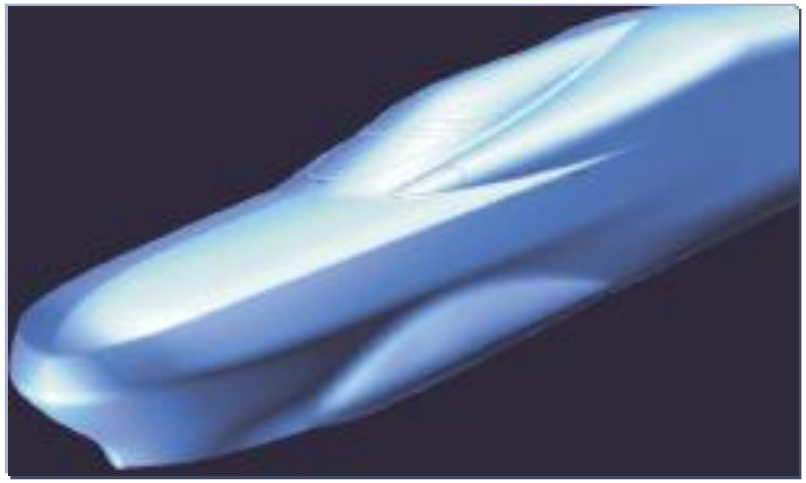
形式手法の導入方法

Levels	導入方法
1	コンセプトの適用
2	ツールを利用した仕様の形式記述
3	仕様の検証
4	全開発工程における形式記述

ソフトウェア開発におけるモデル

Howにこだわりすぎて**What**がわからなくなることを阻止する。
関係ないことをそぎ落として、目的に関することだけを検証する。

風道の検証



車内のユーザービリティの検証

他の産業では当たり前の手順である一方、
ソフトウェア開発では標準化されていない

形式手法の普及

ロバート・フロイドから40年、レイモンド・アプリエルのZおよびVDMから35年経過したにもかかわらず、形式手法が産業界に広く使用されるに至らない。
パーナス,IEEE,2010

2009年以降、98プロジェクトへ形式手法を導入。適用を成功に導くのは、ツール・コンサル・チャンピオンの確保。
A fool with a tool is still a fool!
ラーセン,IPA,2012

日本は欧米に比べ普及は遅延

仕様記述の実践的な活用の工夫

A Semi-formal Mathematioal Expression

$$f(x,y) = \begin{cases} x^2 - y^2 & \text{if } (((y<0) \wedge (x<0)) \vee ((y<0) \wedge (x>0)) \vee ((x=0) \wedge (y=0))) \\ x+y & \text{if } (((y=0) \wedge (x<0)) \vee ((y=0) \wedge (x>0)) \vee ((y>0) \wedge (x=0))) \\ x^2 + y^2 & \text{if } (((y<0) \wedge (x=0)) \vee ((x<0) \wedge (y>0)) \vee ((x>0) \wedge (y>0))) \end{cases}$$

David Lorge Parnas

T[2]

	x < 0	x = 0	x > 0
y < 0	$x^2 - y^2$	$x^2 + y^2$	$x^2 - y^2$
y = 0	$x + y$	$x^2 - y^2$	$x + y$
y > 0	$x^2 + y^2$	$x + y$	$x^2 + y^2$

T[1] T[0]

A “Normal Funotion Table”

出展：David Lorge Parnas,” Making Mathematical Methods Practical for Software Development ”, IPA SEC セミナー, 2009

仕様記述の例題１

主な集合(関数)の種類

出典:銀林,経営工学会平成24年度春季大会基調講演,2012

集合の種類	表記	意味
total function	f:A → B	f はAからBへの全域関数
partial function	f:A ⇸ B	f はAからBへの部分関数
binary relation	f:A ↔ B	f はAとBの間の2項関係
total injection	f:A ↣ B	f はAからBへの全域単射
total surjection	f:A ↠ B	f はAからBへの全域全射

【演習2】 友人の氏名と誕生日のリストを作る場合はどの関数？

[氏名, 日付]

誕生日f : 氏名 関数？ 日付

【正解】 部分関数　誕生日f : 氏名 ⇸ 日付

(同性同名はないとすると)一つの名前には一つの誕生日が対応。

全氏名の誕生日が登録されているわけではないから。

仕様記述の例題2

主な集合(関数)の種類

出典:銀林,経営工学会平成24年度春季大会基調講演,2012

集合の種類	表記	意味
total function	f:A → B	f はAからBへの全域関数
partial function	f:A ⇸ B	f はAからBへの部分関数
binary relation	f:A ↔ B	f はAとBの間の2項関係
total injection	f:A ↣ B	f はAからBへの全域単射
total surjection	f:A ↠ B	f はAからBへの全域全射

【演習2】友人の氏名とEメールアドレスのリストを作る場合はどの関数？

[氏名, Emailアドレス]

Email アドレスf : ？

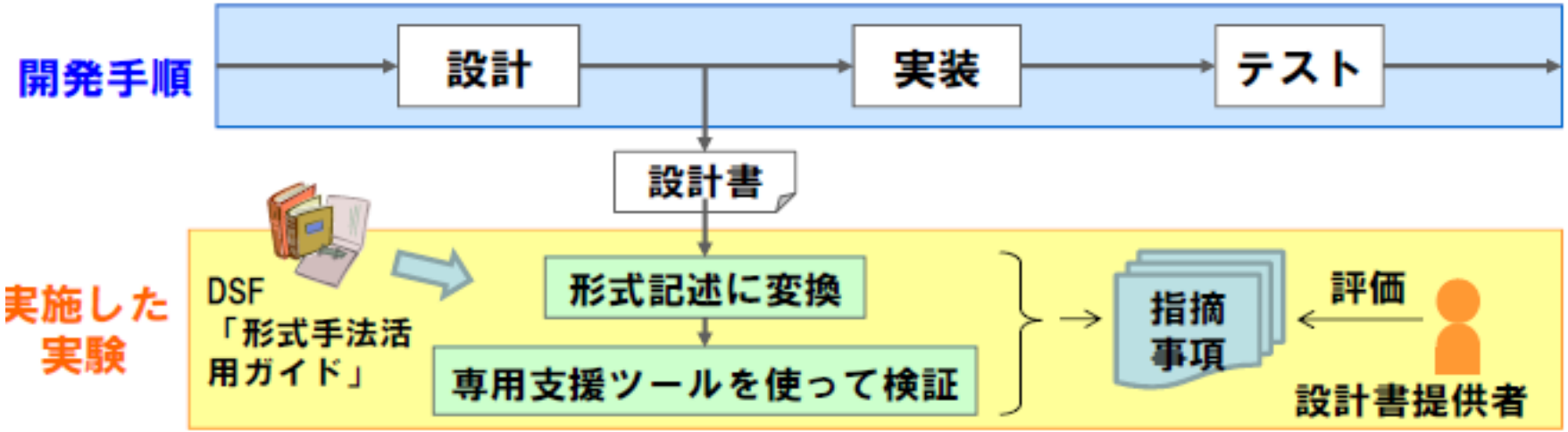
【正解】×:氏名⇸ Emailアドレス　　○: Emailアドレス ⇸ 氏名

一人が複数のEmailアドレスをもつことはあり得るが

複数の人が同一Emailアドレスを保有することはない。

企業で稼動中のシステムへの適用

- 題材とした設計書
  - 東京証券取引所で開発され、運用されている情報システムの設計書
  - 基本設計終了（レビュー完了）時点のもの
- 実験の方法
  - 設計書を形式記述(※1)に変換し、専用支援ツールを使って検証
  - この作業で見つかった指摘事項をリストアップし、設計書提供者（東証）が指摘の妥当性を評価



（※1　使用した形式仕様言語は、Event-B、SPIN、VDM++ の3種類）

5

出展：IPA SEC,"形式手法適用実証実験結果報告",記者会見,2012　http://sec.ipa.go.jp/reports/20120420.html

開発者による評価と実開発との比較

表1　実験で検出された指摘事項55件に対する設計書提供者の評価

設計書提供者による評価	件数
設計書の修正が必要 （実装に影響する可能性がある）	22件
設計書の修正が望ましい （実装に影響する可能性は低いが、修正した方が設計書を理解しやすい）	13件
設計書の修正は不要(※2)	20件
合計	55件

（※2 実験者の領域知識の不足による指摘や、業務への影響がない指摘）

表2　「修正が必要」と評価された指摘事項22件の内訳

実際の開発における発見時期	件数
後工程（実装・テスト）において発見されていた	13件
指摘されていない （ただし、実際の開発では、共通ルールとして徹底されていたため、問題とならなかった）	9件

出展：IPA SEC,
"形式手法適用実証実験結果報告",記者会見,2012
http://sec.ipa.go.jp/reports/20120420.html

実証実験の評価

実験で得られたこと

後工程（実装・テスト）で発見された修正項目を検出

形式手法の適用により、設計段階の品質を向上でき、
後工程での修正コストを削減できる可能性を確認

参加メンバのコメント

「これほどの指摘が出るとは思っていなかった」

「設計書の品質を向上させるための方策の一つとして、従来のレビューに加えて形式手法を採用することは十分可能」

8

出展：IPA SEC,"形式手法適用実証実験結果報告",記者会見,2012 http://sec.ipa.go.jp/reports/20120420.html

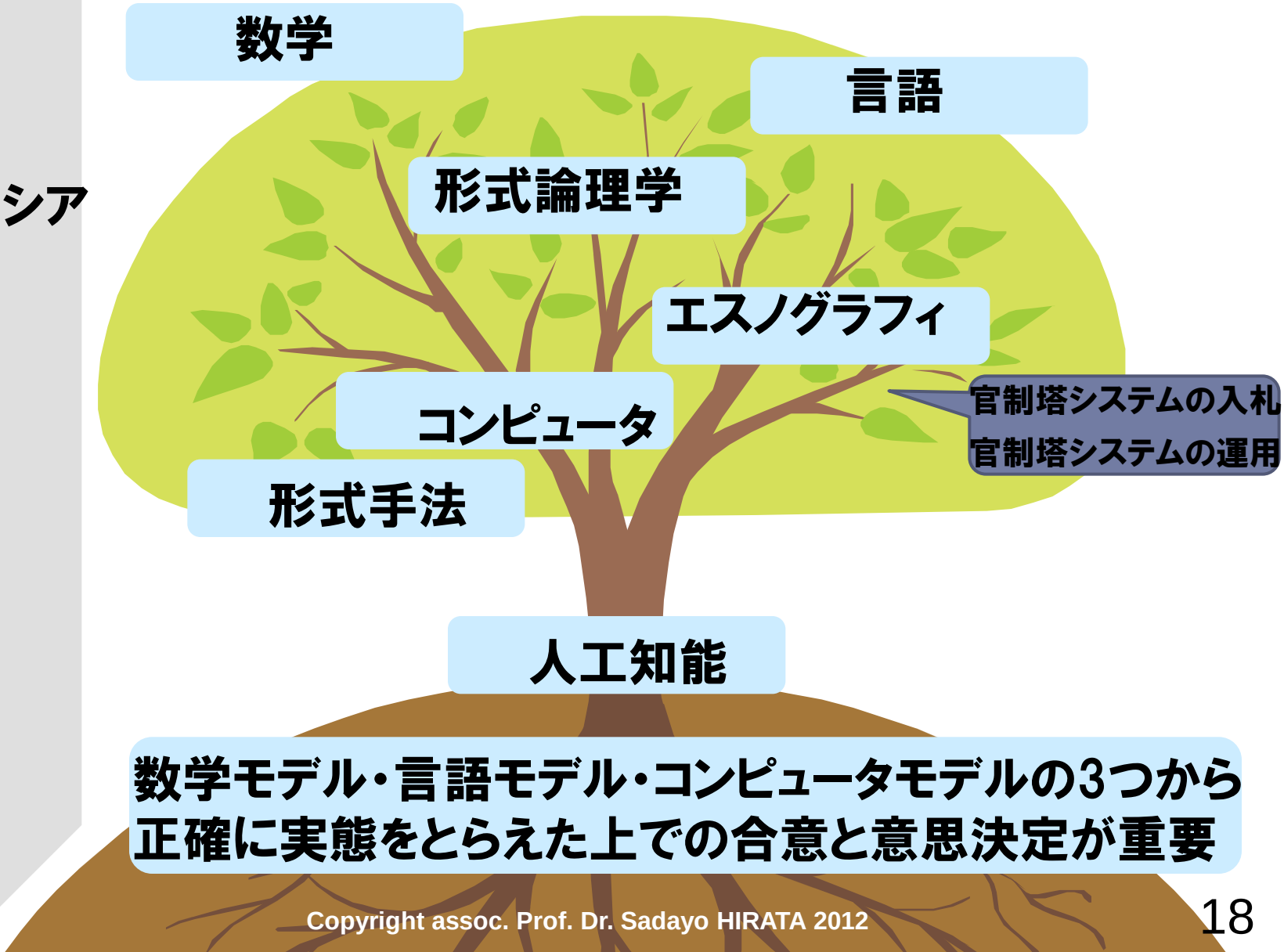
今後のソフトウェア開発

原始

古代ギリシア

20世紀

21世紀



数学モデル・言語モデル・コンピュータモデルの3つから
正確に実態をとらえた上での合意と意思決定が重要

参考文献

[1] ラーセン,形式手法98導入事例の調査・分析から見る高信頼性ソフトウェア開発の現状,IPA,Oct.2012

[2] Alexander Romanovsky, Martyn Thomas, Industrial Deployment of System Engineering Methods, Oct.2012, ISNB978-3-642-33169-5

[3] パーナス, IEEE, 2012

[4] 藤倉俊幸,「形式手法によるソフトウェアの仕様記述と検証」, CQ出版社,2012.

[5] Anthony Hall, Seven Myths of Formal Methods,IEEE Software,vol.7,no.5,pp.11-19,199

[6] 独立行政法人情報処理推進機構,「高信頼ソフトウェア構築技術に関する動向調査」報告書, 2008, http://sec.ipa.go.jp/reports/20080606.html

[7] David Lorge Parnas," Making Mathematical Methods Practical for Software Development ", IPA SEC セミナー, 2009 .

[8] IPA SEC,"形式手法適用実証実験結果報告",記者会見,2012, http://sec.ipa.go.jp/reports/20120420.html

[9] J. M. Spivey, The Z Notation: A Reference Manual, second edition, Prentice Hall, 1992.

[10] ISO/IEC 13568:2002 Information technology -Z formal specification notation- Syntax, type system and semantics.

[11] 平田・銀林,形式手法による業務系ソフトウェアの信頼性向上,jisa quaterly, bulletin 106,2012 summer,2012

[12] S.Hirata, Dependable Systems for Social Safety and Reliability, IEEE ISSRE 1201X-F6G9C7E6C8,2011.

[13] 独立行政法人情報処理推進機構,「形式手法適用調査」報告書, 2010, http://sec.ipa.go.jp/reports/20100729.html