

# IPA/SECにおける形式手法に関する 活動状況と課題

名古屋大学 情報連携統括本部 情報戦略室  
教授 山本修一郎

IPA/SEC形式手法導入プロセス・実証評価WG主査

## IPA/SECにおける形式手法についての主な活動

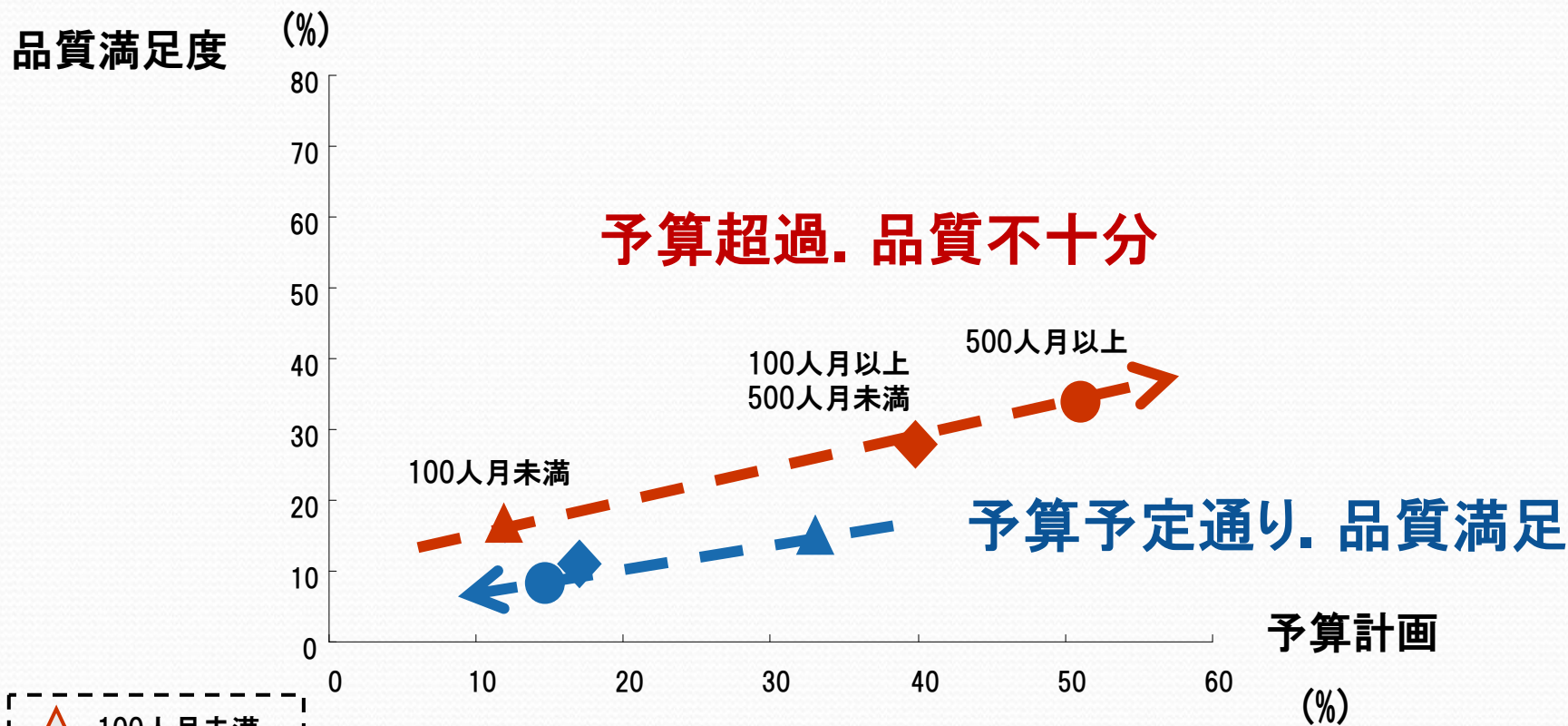
年度	会合	イベント・報告書	委員構成
2007	調査検討会7回	<ul style="list-style-type: none"> <li>・高信頼性システム開発手法フォーラム</li> <li>・高信頼ソフトウェア構築技術に関する動向調査報告書</li> </ul>	産:7 学:5 官:6
2008	高信頼性システム技術WG 6回	<ul style="list-style-type: none"> <li>・日中高信頼性システム検討会</li> <li>・第7回WOCs-Workshop Of Critical Software (JAXA/IPA-SEC)</li> </ul>	産:13 学:5 官:5
2009	高信頼性システム技術WG 6回 PT 3回	<ul style="list-style-type: none"> <li>・高信頼システム開発技術の動向(予定)</li> </ul>	産:16 学:7 官:8
2010	形式手法導入プロセス実証評価WG	<ul style="list-style-type: none"> <li>・ソフトウェアジャパン2010</li> <li>ソフトウェア開発のパラダイム・チェンジinマインドーIPA/SECにおける新しい取組みー</li> </ul>	

# 主要な形式手法の適用事例

	SCADE	Z, Isabelle	VDM	B, Event-B	SPIN	SMV
航空	安全性分析				テスト	設計, テスト
宇宙				探査衛星仕様	<b>要求, 設計</b>	
原子力				設計		
鉄道				仕様, 設計, 実装		状態設計
船舶					設計, テスト	
公共				設計, 実装		
医療		設計仕様記述				
社内システム			<b>金融系仕様</b>	SOA仕様	HMIイベント	<b>HMIタイミング</b>
セキュリティ		状態設計検証		運用設定設計		
OS				カーネルテスト		
ICカード					<b>AP設計</b>	
携帯電話			<b>ICチップファームウェア仕様</b>	プラットフォーム設計		
電力						<b>設計, テスト</b>
組込機器				仕様	<b>テスト</b>	<b>設計</b>

(参考)高信頼ソフトウェア構築技術に関する動向調査報告書, <http://sec.ipa.go.jp/reports/20080606.html>

# 開発規模増による予算超過と品質低下



- △ 100人月未満
- ◇ 100人月以上  
500人月未満
- 500人月以上

(参考)JUAS, 企業IT動向調査2007

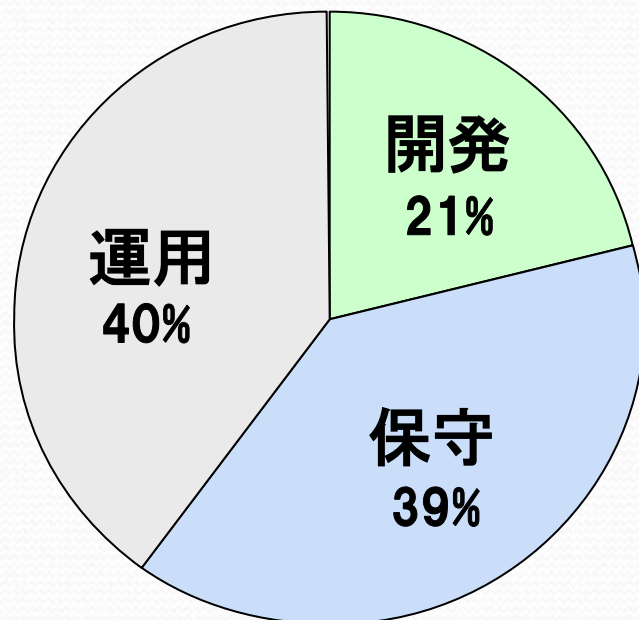
# 重要インフラ障害の原因分類

社会的に問題とされマスコミが取り上げた事例85件を分類  
(2006年12月～2007年10月)

運用性



開発時の十分な考慮



保守性

(参考)経済産業省, 重要インフラ情報システム信頼性研究会 報告書, 2009, IPA

# ディペンダビリティ

人間行動  
コンポーネント  
法制度  
自然物理現象

## ディペンダビリティ

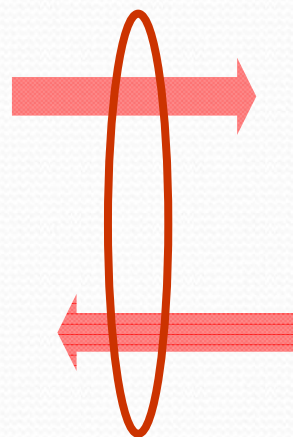
- アベイラビリティ性能及びこれに影響を与える要因、すなわち信頼性性能、保全性性能及び保全支援能力を記述するために用いられる包括的な用語

JIS Z 8115

イベント



● 期待する性能

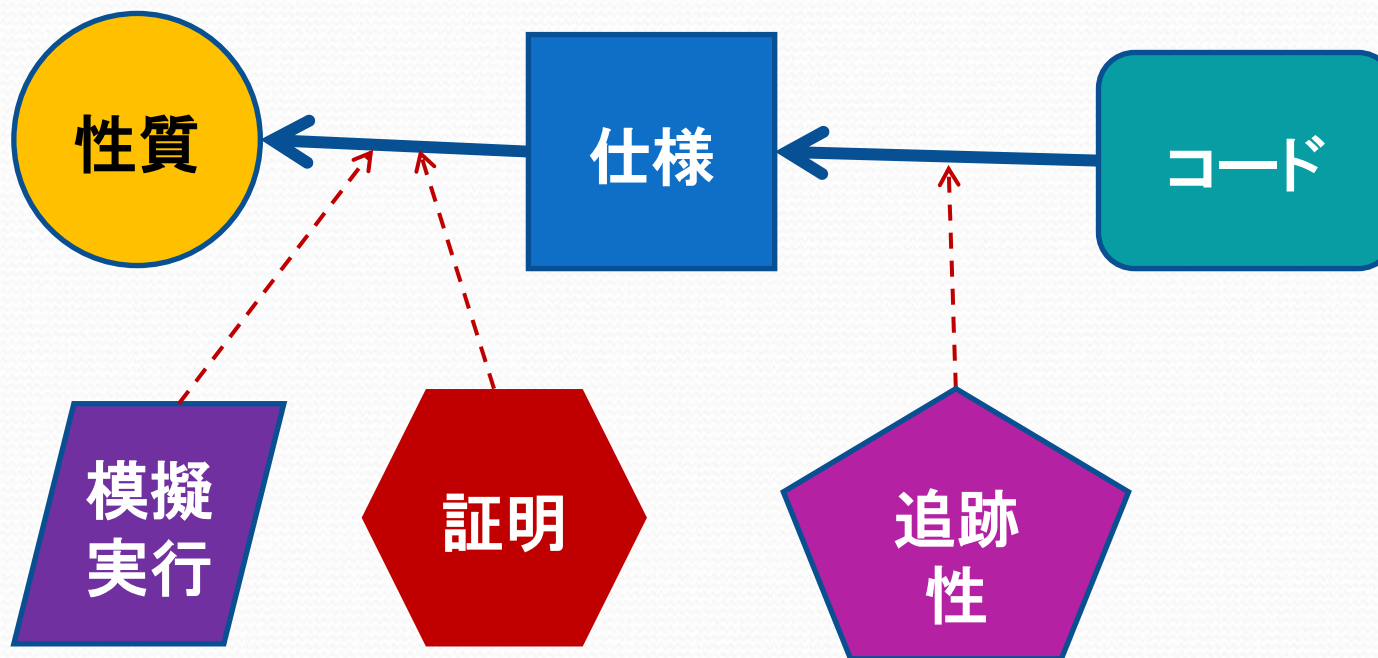


● 信頼性, 保全性

# 要求される性質の論証構造

仕様が満たすべき性質

形式的に記述された仕様



仕様が性質を満たすことを模擬的に確認

仕様が性質を満たすことを数学的に保証

仕様をコードが実現することを保証

# 形式手法の導入プロセス

観点	取り組み内容
技術面	<ul style="list-style-type: none"><li>・高信頼性開発技術フレームワークの策定</li><li>・開発ライフサイクルに対する統合的開発手法の構築</li></ul>
普及面	<ul style="list-style-type: none"><li>・実証研究に基づく成功事例蓄積と知識の体系化</li><li>・国内研究機関との連携</li><li>・国際連携の取り組み</li></ul>
教育面	<ul style="list-style-type: none"><li>・教材開発</li><li>・スキル認定</li></ul>

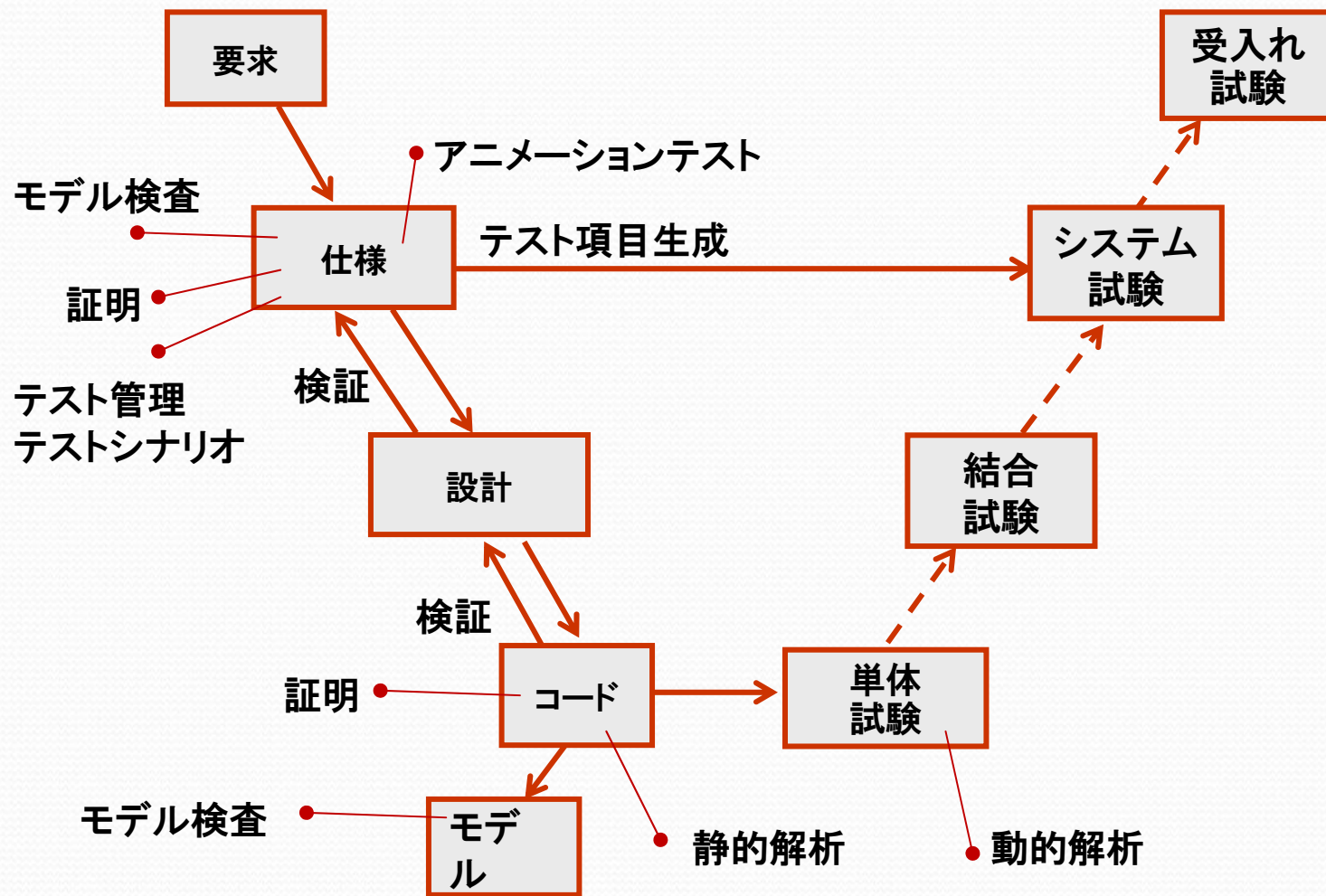
# 導入プロセスの構成要素例

- ①ソフトウェアをコンポーネントに分解して仕様を記述する方法
- ②運用環境や保守環境でソフトウェアが外部とどのような相互作用をするのかを記述する方法
- ③仕様が満たすべき性質を記述する方法
- ④形式的に仕様を記述する方法
- ⑤仕様が性質を満たすことを模擬的に確認できる方法
- ⑥仕様が性質を満たすことを数学的に保証できること
- ⑦仕様をコードが実現する追跡性を確認できること
- ⑧上記の方法を開発プロセスに統合すること

# RASC: 高信頼性システム記述参照フレームワーク

対象	関係	言語例
要求	アクタ間の依存関係	ゴール指向要求
アーキテクチャ	コンポーネント間の接続関係	AADL SysML UML2.0
仕様	コンポーネントの状態関係	Z, VDM B, event-B CTL, LTL
コード	オブジェクトの状態関係	プログラミング言語

# モデル駆動早期テスト型V字モデル



(参考)ROBERT M. HIERONS, et.al., Using Formal Specifications to Support Testing, ACM Computing Surveys, Vol. 41, No. 2, pp. 9-76, 2009

# モデル検査の効果

テストとコードインスペクションによる従来の品質保証プロセスを、経済性と有効性の点で、モデル検査技術が補完し、効率化できることが明らかになった

しかし、高信頼ソフトウェアの開発は、技術面だけでなく社会的な側面を持つ複雑すぎる問題

ソフトウェア開発プロセスにおけるモデル検査ツールの活用が、コードの部分的な性質の検査を自動化することで、プログラマがより複雑な課題に時間をかけられるようになり、プログラマの生産性が向上する

(参考) R. Jhala and R. Majumdar, *Software Model Checking*, ACM Computing Surveys, Vol. 41, No. 4, Article 21, pp. 21-54, 2009.

# システムレベルの留意事項

項目	説明
要求の優先順位	要求を優先順位付け, 重要な <b>性質</b> を簡潔に明確化
要求と仕様化	要求: 環境へのシステムの期待効果 <b>仕様</b> : 環境インタフェースにおけるシステムの振る舞い
現実的な要求	可能な資源に基づく妥当な要求
環境仮説	システムへの要求と環境やオペレータへの要求を区別
障害解析	潜在的なハザード(障害)を解析し, 適切に緩和
ディペンダビリティ ケース	ディペンダビリティケースにより, 実行条件下で優先要求の満足性を確認
操作性テスト	ユーザインタフェースを開発の早期段階でテストし反映
形式モデリング	<b>形式手法</b> により仕様を正確に記述する
分析ツール	要求と <b>仕様</b> 上の欠陥を発見し <b>正当性に対する確信度</b> を向上
標準	アルゴリズムやプロトコルについての標準解を活用

(参考) Jackson, D. et al, Software for dependable systems– sufficient evidence?, NATIONAL RESEARCH COUNCIL, 2008

# エンジニアリング・ケース

項目	説明
情勢変化	要求モデル, アーキテクチャ記述言語, 形式手法を採用するに至った開発・技術の情勢
IT変容の期待	要求モデル, アーキテクチャ記述言語, 形式手法が対象としたITシステム変容への期待
技術課題	要求モデル, アーキテクチャ記述言語, 形式手法が達成しようとしたシステム高信頼化への貢献と解決した技術課題
記述内容	構成要素, 構成要素間関係, 評価特性
記述手順	要求モデル, アーキテクチャ記述言語, 形式手法の作成プロセス
実施組織	要求モデル, アーキテクチャ記述言語, 形式手法を用いた開発組織, 適用組織
取り組み	要求モデル, アーキテクチャ記述言語, 形式手法を記述, 適用, 教育するためのプロジェクト管理面での具体的な取り組み

# 今後の課題

- 形式手法の統合
  - 複数の形式手法の選択と結合
  - 限定性: 性質の網羅性基準, 対象範囲, 適用状況
  - 開発プロセスと開発生産物
  - 追跡性, 進化性, 開放性
  - コミュニケーション
  - 技術者コミュニティ
- 形式手法の実証評価プロジェクト

# 参考文献

1. Marko Auerswald, **Guidelines for the Development of Dependable Integrated Safety Systems – Results of the EU Project EASIS** –, [http://www.easis-online.org/wEnglish/img/pdf-files/Safetronic\\_EASIS.pdf](http://www.easis-online.org/wEnglish/img/pdf-files/Safetronic_EASIS.pdf)
2. **Deploy**, <http://www.deploy-project.eu/>
3. ROBERT M. HIERONS, et.al., **Using Formal Specifications to Support Testing**, ACM Computing Surveys, Vol. 41, No. 2, pp. 9-76, 2009
4. Jackson, D. et al, **Software for dependable systems– sufficient evidence?**, NATIONAL RESEARCH COUNCIL, 2008
5. Ciera Jaspán, Michael Keeling, Larry Maccherone, Gabriel L. Zenarosa, and Mary Shaw, **Software Mythbusters Explore Formal Methods**, IEEE SOFTWARE, November/December 2009 , pp.60-63
6. R. Jhala and R. Majumdar, **Software Model Checking**, ACM Computing Surveys, Vol. 41, No. 4, Article 21, pp. 21-54, 2009.
7. JUAS, **企業IT動向調査2007**
8. **経済産業省, 重要インフラ情報システム信頼性研究会 報告書**, 2009, IPA
9. Steven P. Miller, Michael W. Whalen, Darren D. Cofer, **Software Model Checking Takes Off**, Communications of the ACM, February 1, 2010, pp.58-84.
10. David Lorge Parnas, **REALLY RETHINKING ‘FORMAL METHODS’**, IEEE COMPUTER , JANUARY 2010, pp.28-34
11. **高信頼ソフトウェア構築技術に関する動向調査報告書**, <http://sec.ipa.go.jp/reports/20080606.html>
12. JIM WOODCOCK, PETER GORM LARSEN, JUAN BICARREGUI, and JOHN FITZGERALD, **Formal Methods: Practice and Experience**, ACM Computing Surveys, Vol. 41, No. 4, Article 19, pp.19-36, Publication date: October 2009.