IPA® Software Engineering Center
Information-technology Promotion Agency, Japan

システム障害とディペンダビリティ

2009年11月20日

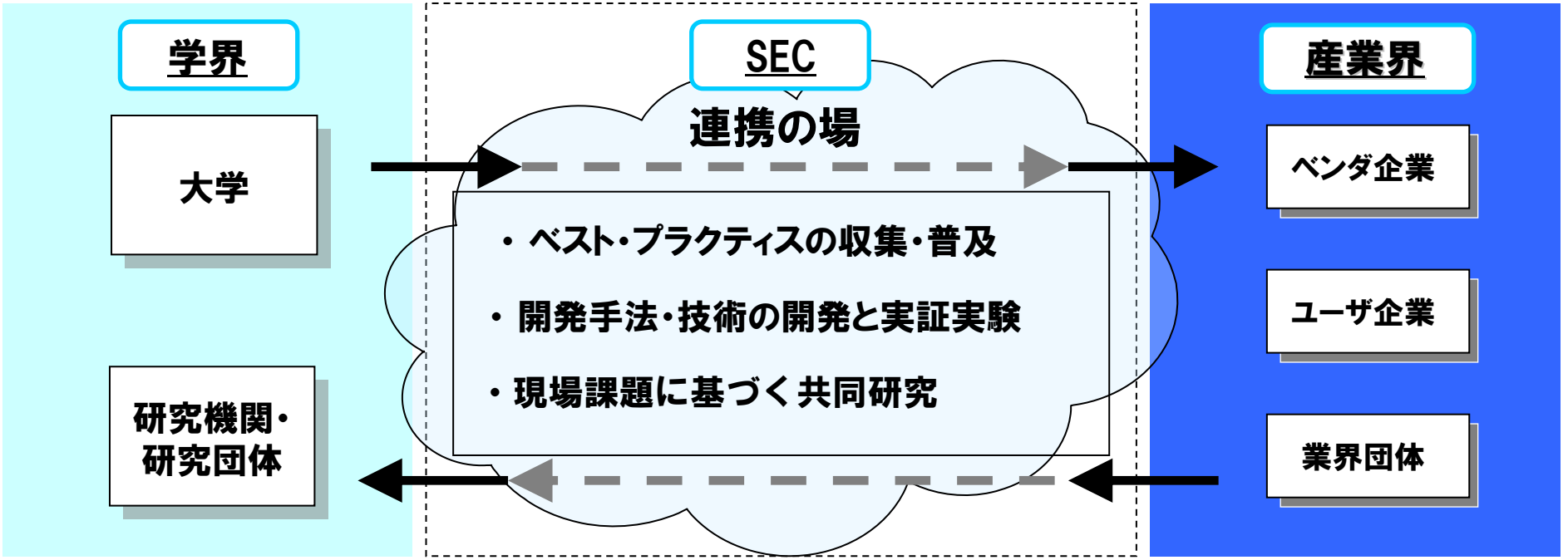
独立行政法人 情報処理推進機構
ソフトウェア・エンジニアリング・センター
所長　松田　晃一

ソフトウェア・エンジニアリング・センター（SEC）

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

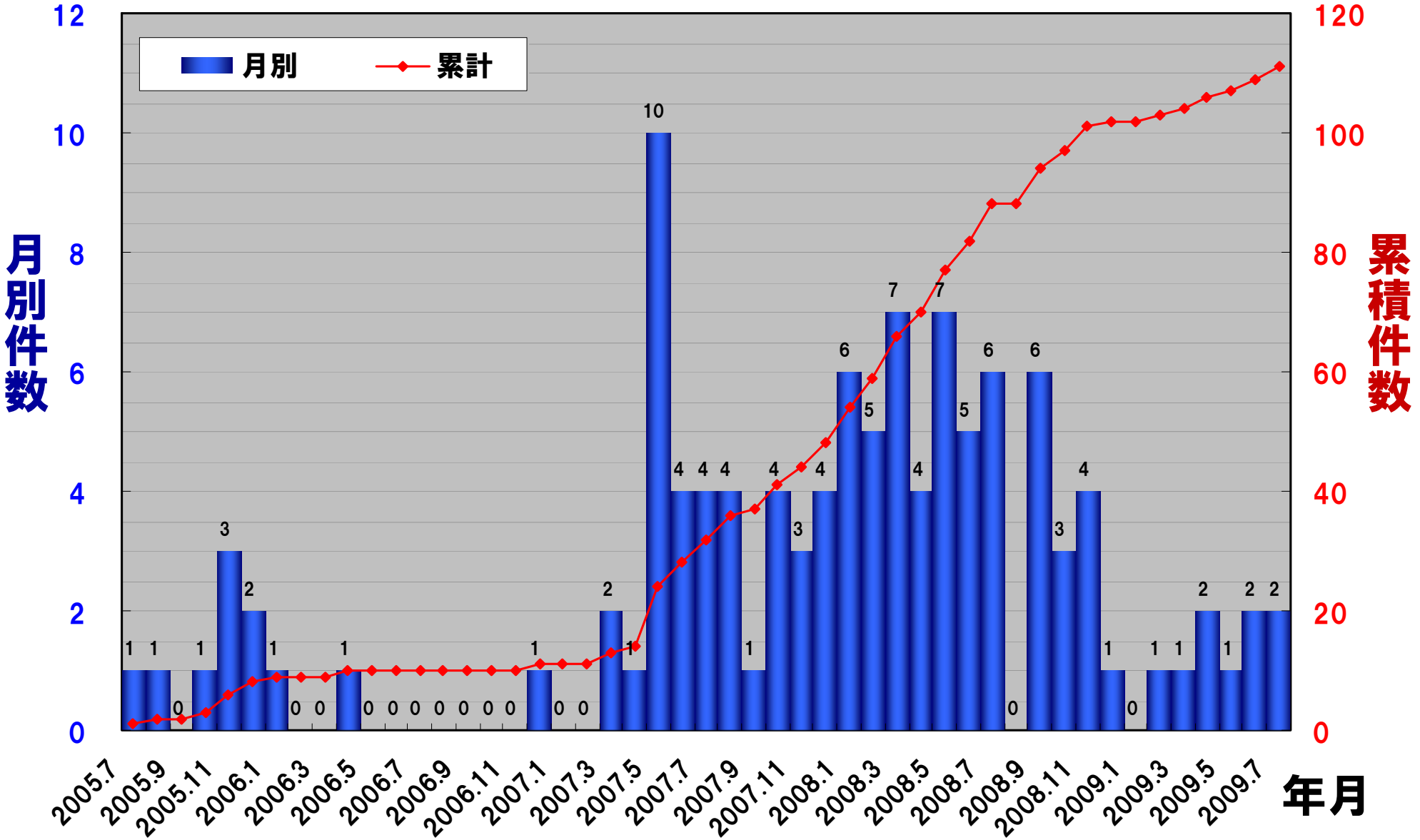
■ ソフトウェア開発力強化にむけて、技術と人材の開発のための産学官連携の拠点

■ 情報システムの信頼性確保にむけたソフトウェア・エンジニアリングの推進



情報システムの障害に関する報道件数の推移

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri



講演の内容

1．情報化社会が直面するリスク

2．安心・安全な情報化社会へ向けて

3．DEOSへの期待

情報社会が直面するリスク

■ システムの大規模化・複雑化：

- 機能の複雑さ、規模の増大
- 独立に設計されたシステムが相互に繋がったネットワークシステム
  例：三菱東京UFJ統合システムとセブン銀行システムの接続事故

- 社会が大きく依存しているITシステムの全容を把握しきれていない

情報社会が直面するリスク

■ システムの大規模化・複雑化：

- 機能の複雑さ、規模の増大
- 独立に設計されたシステムが相互に繋がったネットワークシステム
- 社会が大きく依存しているITシステムの全容を把握しきれていない

■ 情報量の爆発・ボーダーレス化：

- 大量に生産され続ける巨大な情報が国境を越えて蓄積、流通
- 重要な財産である情報が手元に無い、雲のかなたに
  例： SaaSによる定額給付金管理システム

クラウド・コンピューティング

コンピュータ（ハード・ソフト）をユーザが所有

**ユーザは、コンピュータを必要な量だけ利用し、
利用量に見合った料金を支払う**

- ○ 早くて、安くて、手間要らず
- ○ 伸縮自在

- ● 重要な財産である「情報」が手元に無い
- ● サービスの信頼性

情報社会が直面するリスク

システムの大規模化・複雑化：

- 機能の複雑さ、規模の増大
- 独立に設計されたシステムが相互に繋がったネットワークシステム
- 社会が大きく依存しているITシステムの全容を把握しきれていない

情報量の爆発・ボーダーレス化：

- 大量に生産され続ける巨大な情報が国境を越えて蓄積、流通
- 重要な財産である情報が手元に無い、雲のかなたに

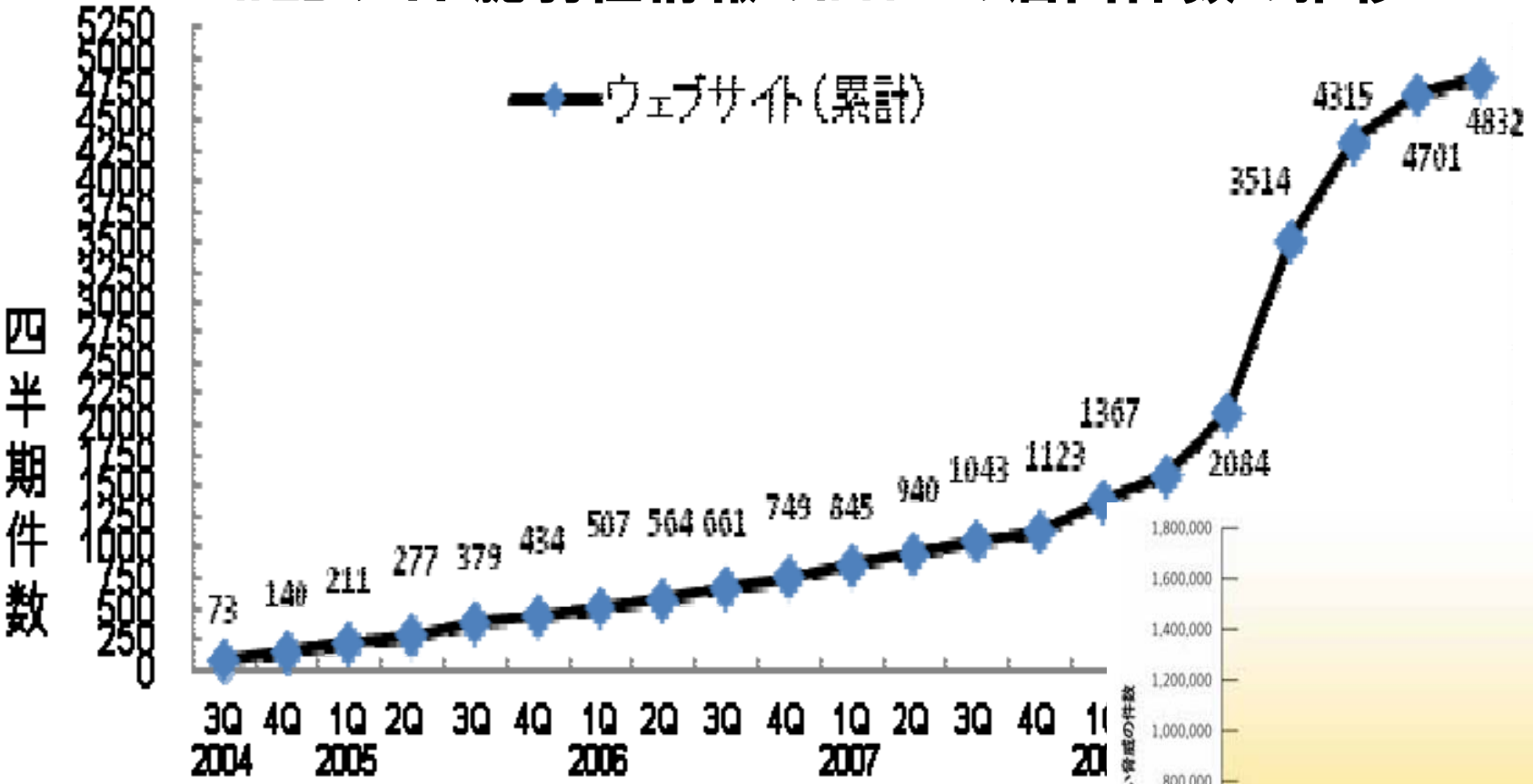
利用の多様化：

- 初心者ユーザも熟達ユーザも等しくアクセス可能

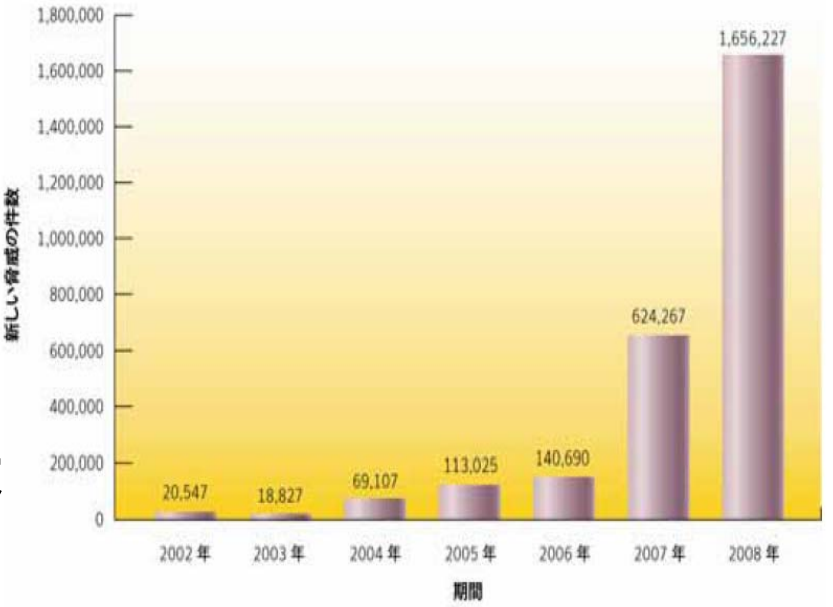
セキュリティ事故の推移

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

WEBサイト脆弱性情報のIPAへの届出件数の推移



ウィルスなどの検出件数

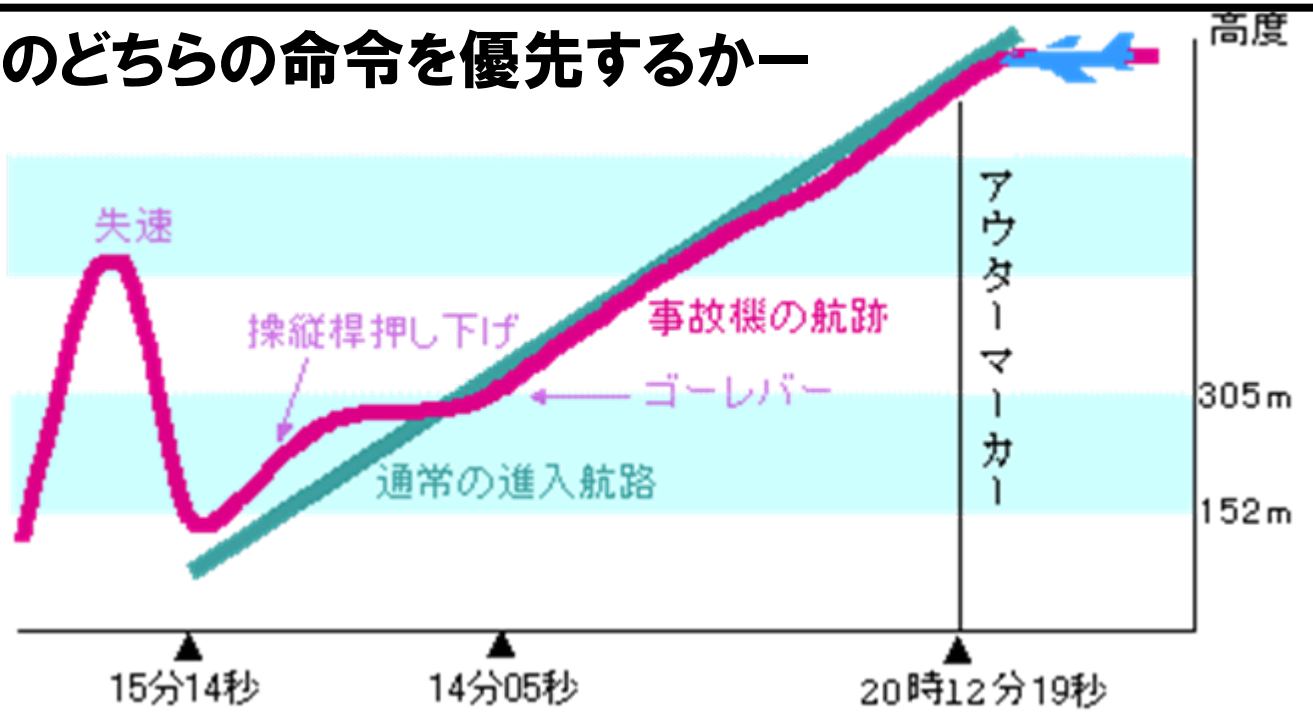


情報社会が直面するリスク

- **システムの大規模化・複雑化：**
  - 機能の複雑さ、規模の増大
  - 独立に設計されたシステムが相互に繋がったネットワークシステム
  - 社会が大きく依存しているITシステムの全容を把握しきれていない
- **情報量の爆発・ボーダーレス化：**
  - 大量に生産され続ける巨大な情報が国境を越えて蓄積、流通
  - 重要な財産である情報が手元に無い、雲のかなたに
- **利用の多様化：**
  - 初心者ユーザも熟達ユーザも等しくアクセス可能
- **組込みソフトウェアの拡大：**
  - 事故の影響の質的な変化
    例：発熱事故、実はソフトのバグ
  - 人間とシステムの予期せぬ相互作用

自動操縦と手動操縦の矛盾

－人間とコンピュータのどちらの命令を優先するか－



名古屋空港で中華航空機が着陸に失敗炎上（1994年4月）

着陸態勢に入った中華航空140便（エアバスA300-600R）はアウターマーカー通過後、
自動操縦装置が誤ってゴー・アラウンド・モード（着陸やり直し）になる。

パイロットはそれに気付かず、操縦輪で機首下げの操作を行った。
オートマチック・フライト・システム（AFS）の作動が相反し、パイロットが懸命に機首を下げようとする意図に反してコンピュータに制御された水平尾翼が機首上げの方向に反発し続けたため失速、墜落炎上した。

講演の内容

1．情報化社会が直面するリスク

2．安心・安全な情報化社会へ向けて

3．DEOSへの期待

情報セキュリティ基本計画

内閣官房 情報セキュリティセンタ（NISC）

「事故前提社会」への対応力強化

- 事故の可能性を完全に排除する対策の実現は容易ではないという点に関する理解を社会全体で増進
- 万一問題が顕在化しても、過敏な反応を起こさず、冷静に受け止めて適切な対応を迅速に行う

- 諦めて予防の対策を行わない、被害は仕方ないと諦めるということでは決して無い

第1次基本計画（2006年度～2008年度） 第2次基本計画（2009年度～2011年度）

社会的共通認識の形成と高度情報化社会の協働建設

SEC Software Engineering for Mo·No·Zu·Ku·Ri

経済産業省（METI）

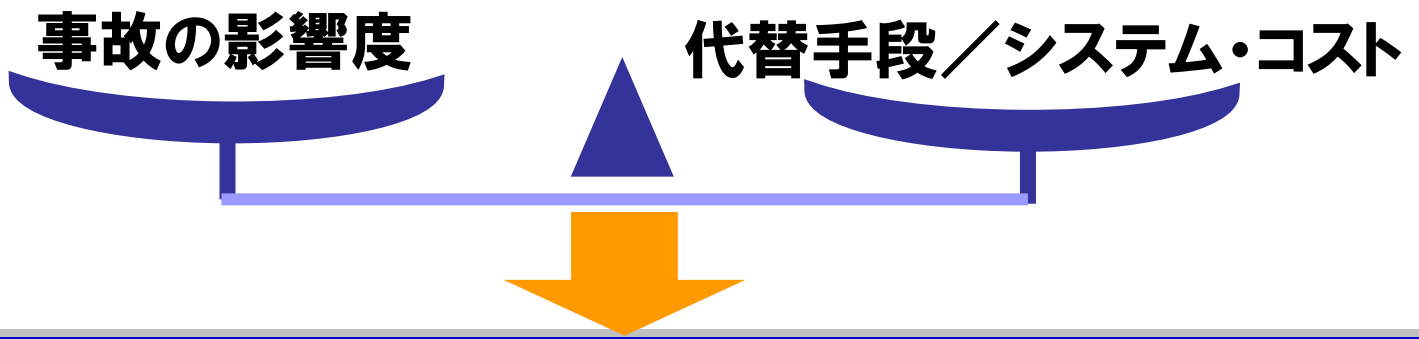
情報システムの信頼性・セキュリティを「見える化・測る化」し、サービス内容とリスク、コストバランスがとれた信頼性・セキュリティの水準を共通認識とし、この目標を協働して実現

- ✔ 経営者が活用できる信頼性・セキュリティ水準の判断・評価基準の策定
- ✔ ユーザ・ベンダの現場担当者が活用できる要求項目・水準の設定
- ✔ 定量的評価・管理指標の策定
- ✔ 相互運用性やセキュリティなどに関する評価体制の整備

高度情報化社会における情報システム・ソフトウェアの 信頼性及びセキュリティに関する研究会 中間報告より（経済産業省）

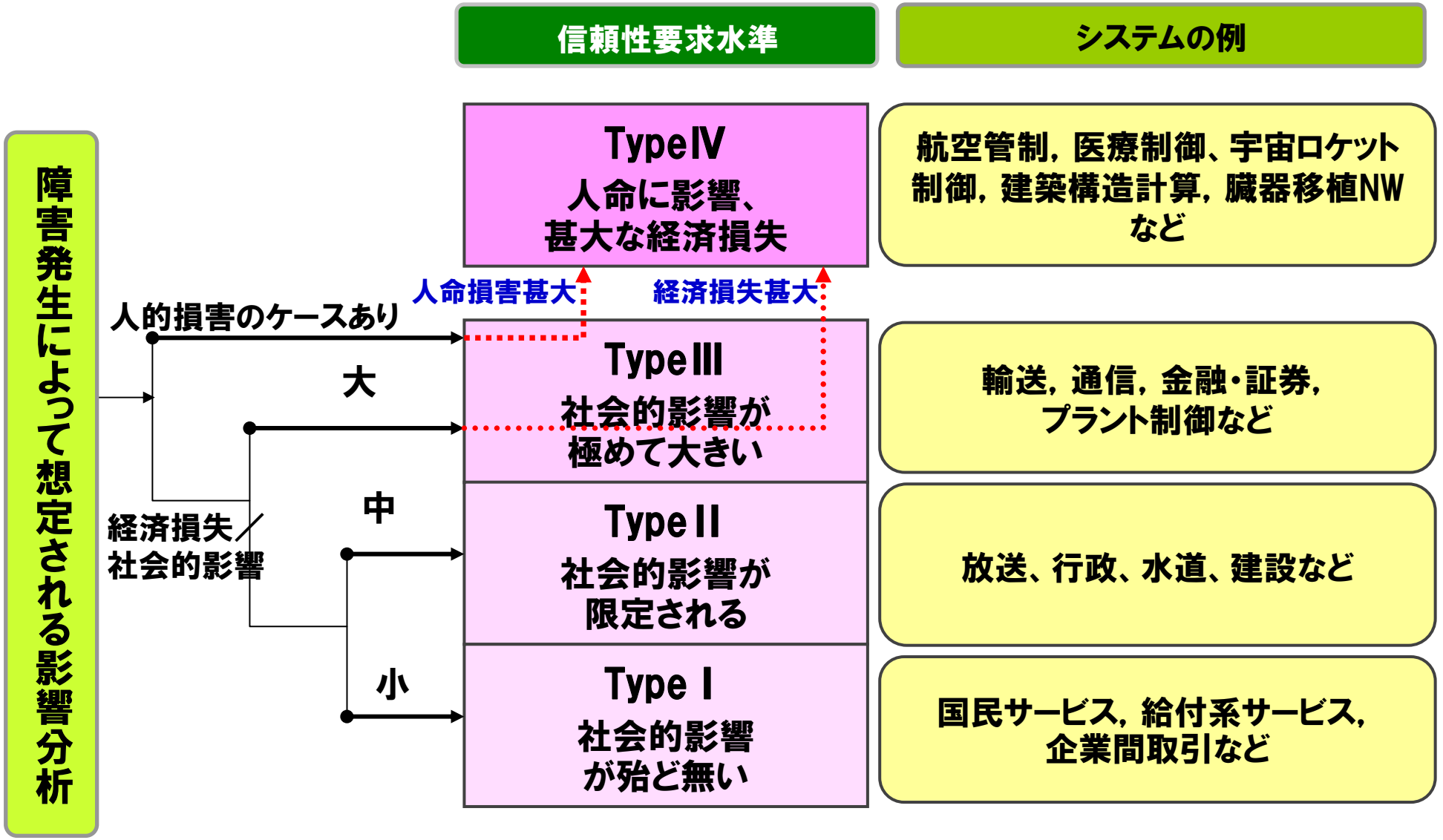
信頼性要求水準設定の考え方

- サービス中断によるエンドユーザの影響度評価
  - ✔ 想定される事故の重大性 ： 人的・経済的・時間的損失と影響範囲
  - ✔ 想定される事故の発生頻度 ： 発生の頻度
- 代替手段の有無・事業継続計画（BCP）
- システムで対処する場合のコスト

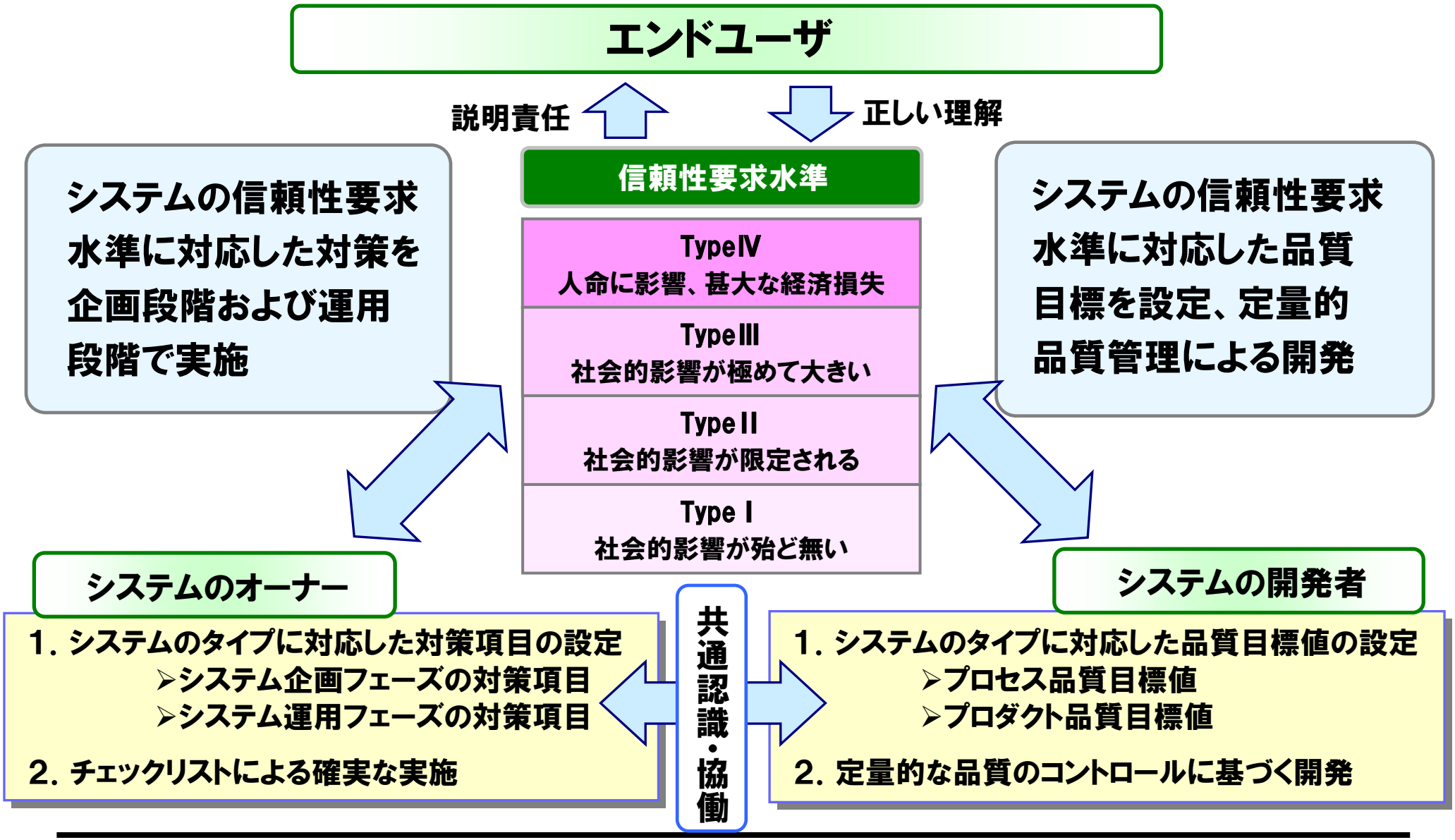


システム信頼性要求水準の設定

システムの信頼性要求水準



信頼性要求水準の共通認識と協働



定量的品質目標の設定と品質管理

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

ソフトウェアの定量的品質管理項目　例			信頼性要求水準			
			Type I	Type II	Type III	Type IV
プロダクト品質評価指標 成果物の出来栄えを評価する指標	ドキュメント品質	ボリューム品質	10	15	20	40
		バランス品質	・・	・・	・・	・・
	ソースコード品質	コード特性品質				
		テスト密度				
プロセス品質評価指標 ソフトウェアの開発過程の作業の品質を評価する指標	レビュー品質	レビュー作業充当率				
		レビュー作業実施率				
	テスト品質	テスト作業充当率				
		テスト作業実施率				
		欠陥摘出密度	・・	・・	・・	・・

テスト十分性品質評価指標 – テスト密度

ＩＤ	PD40	略称	DOTI
名称	テスト密度	名称（英語表記）	Dencity Of Test Items

参考値	Ⅰ	Ⅱ	Ⅲ	Ⅳ	補正ベース値
	25.00	50.00	75.00	100.00	25.00
参考値の範囲	0.00 ～ 50.00	25.00 ～ 75.00	50.00 ～ 100.00	75.00 ～ 125.00	
計測単位	項目				
許容誤差	有効数字上位2桁まで				
指標値の意味	● ソース規模あたりのテスト実施 十分性、ソースに対するテスト				
計測方法	テスト項目数 / ソースコード全行				

信頼性要求水準がTypeⅡの場合

**テスト項目は
1KLOCあたり50項目程度
を目安とする**

プロセス品質評価指標の例

設計レビュー作業充当率

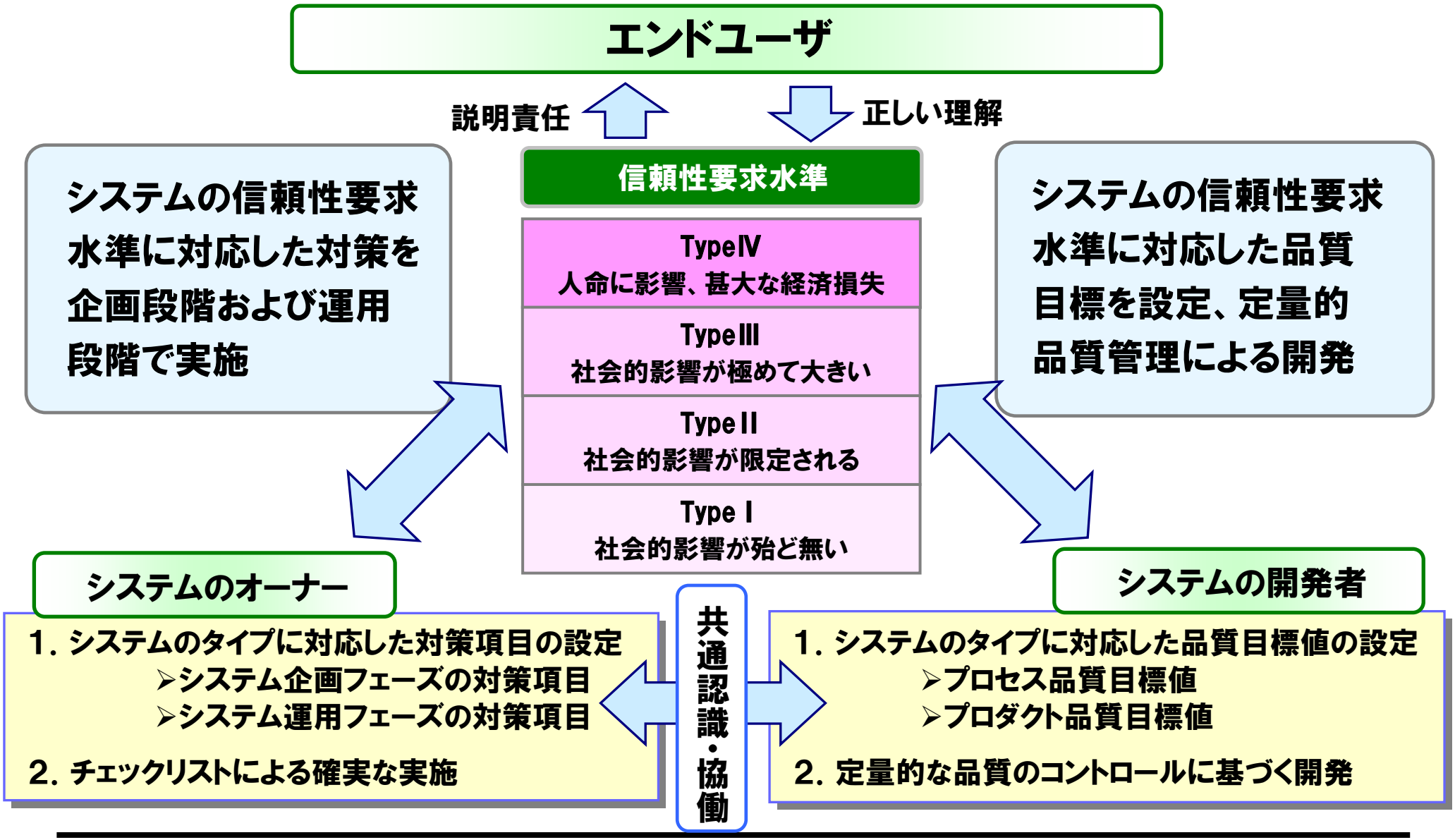
ID	PR11	略称	RDRE
名称	設計レビュー作業充当率	名称(英語表記)	Ratio of the Design Review Effort

参考値	I	II	III	IV	補正ベース値
	2.00	6.00	10.00	14.00	4.00
参考値の範囲	0.00 ～ 6.00	2.00 ～ 10.00	6.00 ～ 14.00	10.00 ～ 18.00	
計測単位	%				
許容誤差	有効数字上位2桁までのパーセ				
指標値の意味	● 設計のレビューにどれだけ時 (設計プロセスの工数)とのバ ● 安全性、信頼性を要求される りますが、設計作業そのもの ほど良いというものではなく ります。				
計算方法	設計レビュー工数/設計作成工数 RDRE = REDE/PEDE				

信頼性要求水準がTypeIIの場合

設計レビューは
設計作業工数のうちの6%程度
を目安とする

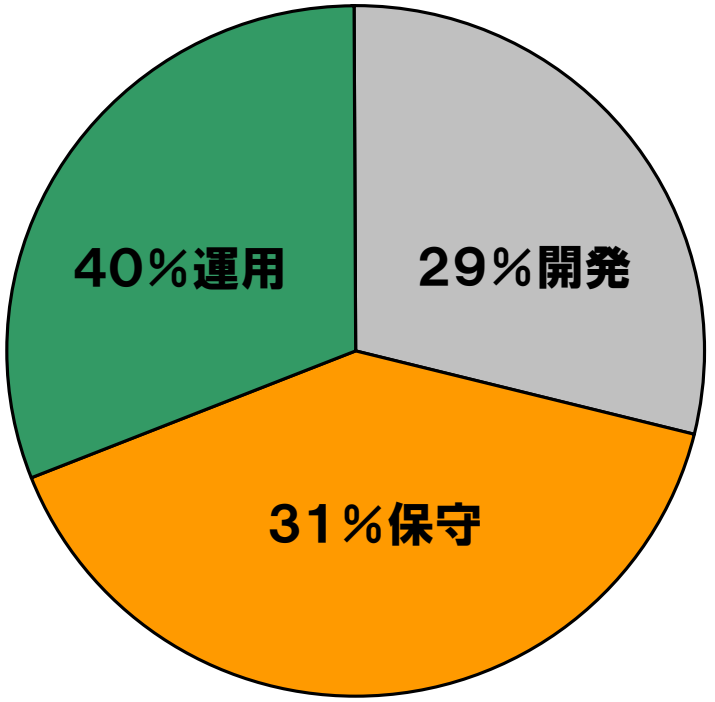
信頼性要求水準の共通認識と協働



障害事例分析の結果1

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

■ 85障害事例（06年12月～08年10月）の分析例
（SEC重要インフラ研究会）



・開発に係わる原因
　による障害:　29%
・運用／保守に係わる原因
　による障害:　71%

（
運用段階での障害　40%
保守段階での障害　31%
）

運用・保守に起因する障害

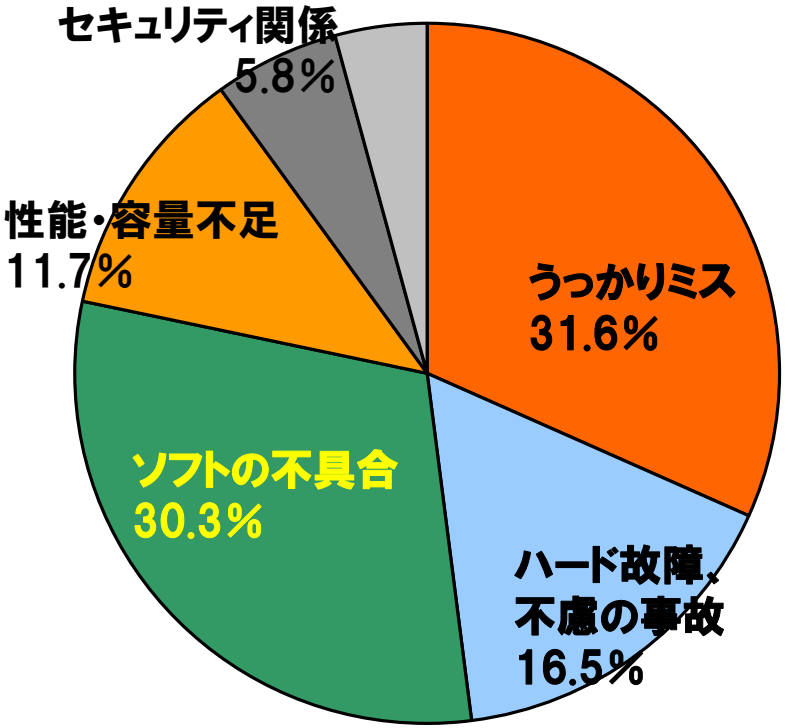
SEC
Software Engineering
for Mo・No・Zu・Ku・Ri

原因の工程		事例
運用		・ハード障害の切り替えミス ・データ容量上限値の設定誤り ・待機系の訓練で本番データ利用
保守	是正保守	・確認テストで使用した環境の復元誤り
	予防保守	
	適応保守	・ハード増設時のソフト組込みミス ・性能改善のためのファイル再配置ミス
	完全化保守	・業務プログラム更新時の設定ミス

障害事例分析の結果2

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

■ 291障害事例（00年1月～09年8月）の分析例
（出典：日経コンピュータ09年8月19日号）



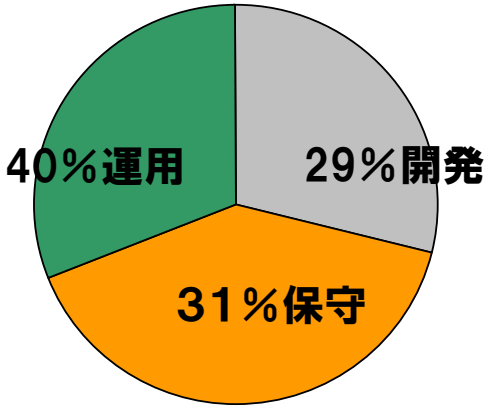
うっかりミス	31.6%
ハード故障、不慮の事故	16.5%
ソフトの不具合	30.3%
性能・容量不足	11.7%
セキュリティ関連の不具合	5.8%
その他	4.1%

障害事例分析の結果

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

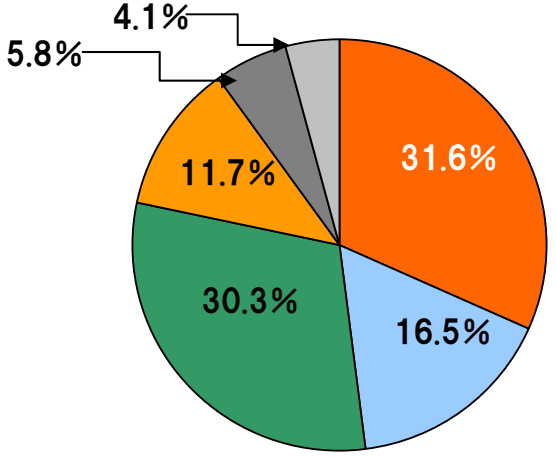
開発に起因する故障よりも、運用・保守段階に起因する故障が多い

- 85障害事例 （06年12月～08年10月）の分析例（SEC重要インフラ研究会）



開発に係わる原因による障害	29%
運用・保守に係わる原因による障害	71%
（運用段階での障害	40%）
（保守段階での障害	31%）

- 291障害事例 （00年1月～09年8月）の分析例（日経コンピュータ09年8月19日号）



うっかりミス	31.6%
ハード故障、不慮の事故	16.5%
ソフトの不具合	30.3%
性能・容量不足	11.7%
セキュリティ関連の不具合	5.8%
その他	4.1%

信頼性要求水準の共通認識と協働

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

エンドユーザ

説明責任 ⇧　⇩ 正しい理解

システムの信頼性要求水準に対応した対策を企画段階および運用段階で実施

信頼性要求水準

TypeⅣ 人命に影響、甚大な経済損失
TypeⅢ 社会的影響が極めて大きい
TypeⅡ 社会的影響が限定される
TypeⅠ 社会的影響が殆ど無い

システムの信頼性要求水準に対応した品質目標を設定、定量的品質管理による開発

システムのオーナー

1. システムのタイプに対応した対策項目の設定
   - ➢システム企画フェーズの対策項目
   - ➢システム運用フェーズの対策項目
2. チェックリストによる確実な実施

共通認識・協働

システムの開発者

1. システムのタイプに対応した品質目標値の設定
   - ➢プロセス品質目標値
   - ➢プロダクト品質目標値
2. 定量的な品質のコントロールに基づく開発

情報システムの信頼性向上に関するガイドライン

5大分類　14中分類　65対策項目

2009年(第2版)経済産業省

Ⅱ. 信頼性・安全性向上に向けての全般的配慮事項
Ⅲ. 企画・要件定義・開発及び保守・運用全体における事項
　1. 企画・要件定義段階における留意事項
　2. 開発段階における留意事項
　3. 保守・運用段階における留意事項
　4. 障害対応に関する留意事項
　5. システムライフサイクルプロセス全体における横断的な留意事項
Ⅳ. 技術に関する事項
Ⅴ. 人・組織に関する事項
Ⅵ. 商慣行・契約・法的要素に関する事項

Ⅱ．信頼性･安全性向上に向けての全般的配慮事項

1．関係者の責務
- （１）情報システム利用者の責務
- （２）情報システム供給者の責務
- （３）共同作業であることの認識

2．経営層の責務
- （１）情報システム障害が経営リスクの問題であることの認識
- （２）経営資源の投入
- （３）CIO（情報統括役員）の登用と活用
- （４）説明責任の認識
- （５）保守･運用の重要性の認識
- （６）事業継続計画の策定と訓練の実施

3．未然防止と事後対策の両側面からの対策の実施

4．信頼性･安全性向上に向けた多面的取組の必要性

5．情報システム障害に対する動作の基本

講演の内容

１．情報化社会が直面するリスク

２．安心・安全な情報化社会へ向けて

３．DEOSへの期待

ディペンダビリティ（Dependability）

- 一般の市民が安心して利用できる頼りになるシステム
- ユーザ視点

- 様々なアクシデント（例えば、故障、バグ、設計ミス、システムへの悪意ある妨害など）があったとしても、システムの提供するサービスをユーザが許容できるレベルで維持できるようにすること。

ディペンダビリティの定義と定量的な指標

要件・指標の実現を合理的に説明できる開発・運用の方法論

ITシステムは“生きたシステム”

- ハードの故障は常態化、もはや異常ではない
  - 平均故障間隔(MTBF)が3年のサーバを1000台使うと、毎日1台は故障する
- ソフトには、一定のバグが残る
  - 確率的にバグが顕在化し故障を起こすことは避けられない
- 業務内容の変更、サービスの改善のためにソフトウェアは常に入替えが行われる
- ハードウェアの機種更改による入替えは常時行われる
- システム同士の接続が運用しながら行われる
- 人間とシステムの予期せぬ相互作用が起きる

外部条件があらかじめ定まった静的なシステム

↓

外部条件が変化し、事前に全ては定まらない
“生きたシステム”

IPA® Software Engineering Center
Information-technology Promotion Agency, Japan

システム障害とディペンダビリティ

2009年11月20日

独立行政法人 情報処理推進機構
ソフトウェア・エンジニアリング・センター
所長　松田　晃一