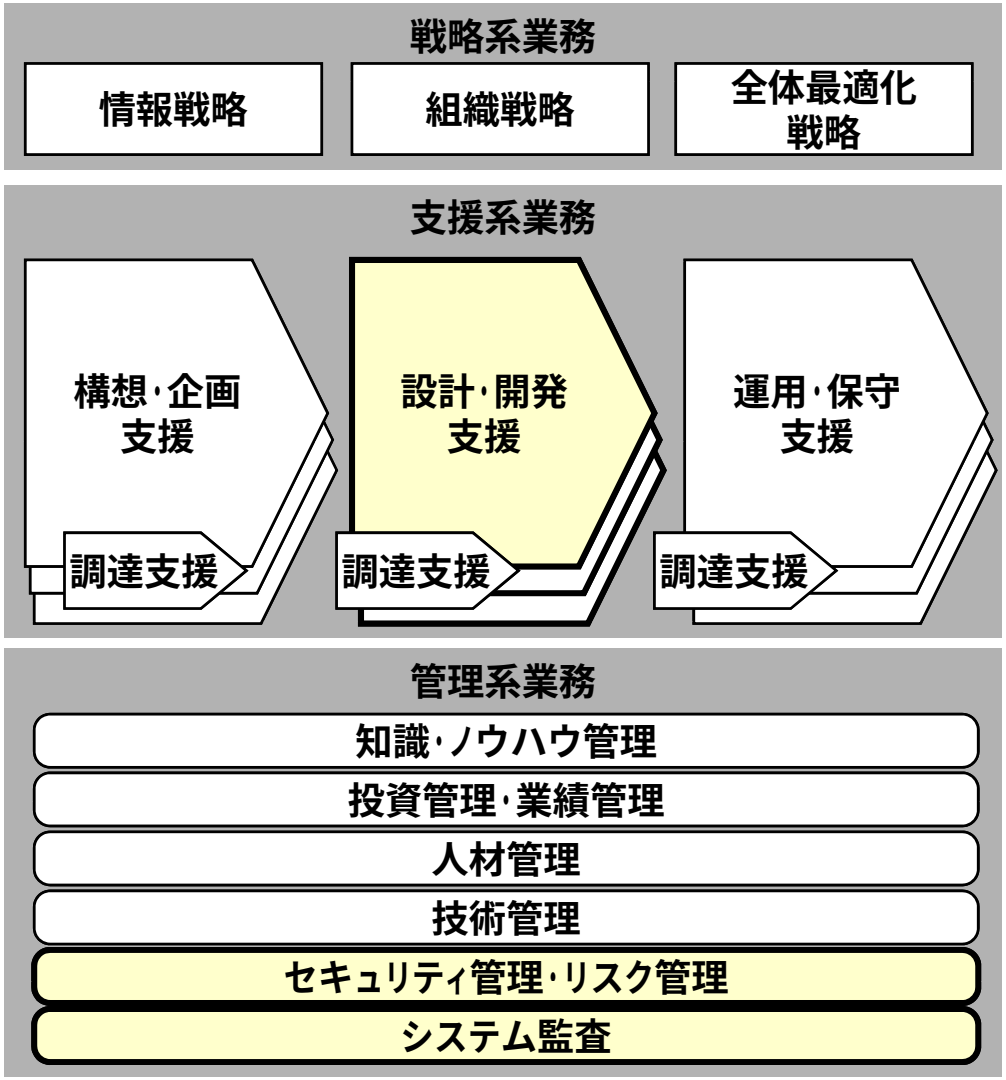
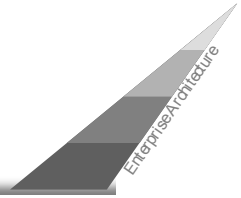
品質管理・信頼性管理・セキュリティ


戦略系業務
情報戦略
組織戦略
全体最適化
戦略
支援系業務
構想･企画
支援
調達支援
設計･開発
支援
調達支援
運用･保守
支援
調達支援
管理系業務
知識･ノウハウ管理
投資管理･業績管理
人材管理
技術管理
セキュリティ管理･リスク管理
システム監査

品質管理・信頼性管理・セキュリティでのCIOの役割

- どんな問題が発生するのか想像し、それに対する適切な指示を行う。
- 予算査定や調達などで、失敗プロジェクトの「入り」をコントロールしたが、走り出したプロジェクトを円滑に送り出し、運用を続けさせる「出」をコントロールするのが、ここでのCIOの役割である。

品質管理･信頼性



品質管理を行うための前提

- **基礎データをそろえる**
  - **システム規模はわかっているか**
    - **何がしかのデータを元に、共通尺度であるFPを算出**
      - **価格→FP、画面数→FP、KLOC→FP**
  - WBS**はかけているか**
    - **進捗と品質をあわせて管理するのであれば、基礎となるWBSがしっかりかけている必要がある**
      - WBS**は網羅性があるのか？WPは10日前後か？**
  - **日常の各種数値を記録しているか**
    - **バグ数、テスト項目数・・・・**
  - **ドキュメンテーションの充実**
    - **最低整備すべきドキュメントの整備**

品質や信頼性を向上させるためのポイント

- システムの品質や信頼性を向上させるには以下の取り組みが重要である。
  - さまざまな障害を想像できる力を身につける
    - これが不足していると設計書を見たときにチェックポイントがつかめない
      - 日経コンピュータの動かないコンピュータなどを見て、さまざまなケースをイメージできるようにトレーニングする
  - データに基づき沈没しつつあるプロジェクトを予見する
    - 沈没するプロジェクトはさまざまな危険信号を発している
  - 楽観的に考えない
    - プロジェクトマネージャは、特に前半において、希望的観測により問題の対策を遅らす場合が多い。ユーザ、ベンダの双方に同じ傾向があるので注意が必要である
  - 品質の悪いシステムを防ぐために
    - 入りを制す
      - 正しい予算や計画にすることでコントロールを行う
        - » 予算査定や調達などで、失敗プロジェクトの「入り」をコントロール
    - 流れを制す
      - 開発工程をモニターすること、傾向把握からコントロールを行う
        - » 走り出したプロジェクトが外れた方向に進んでいないか「流れ」をモニター
      - 内容的にはレビューで潰していく
    - 出を制す
      - 納品の検査で納品前後のコントロールを行う
        - » 運用側で受け入れられるのか「出」をコントロール

認証は信用できるのか

- ISO9000を取得している企業のプロダクト品質は高いのか？
- CMMを持っている企業の品質は高いのか？

- No！！

- 見せ掛けだけの認証取得をしている企業が多い
  - ISO14000を持っているのに真夏にスーツで名刺交換してくる企業とか
  - つまりは、理念が浸透していない

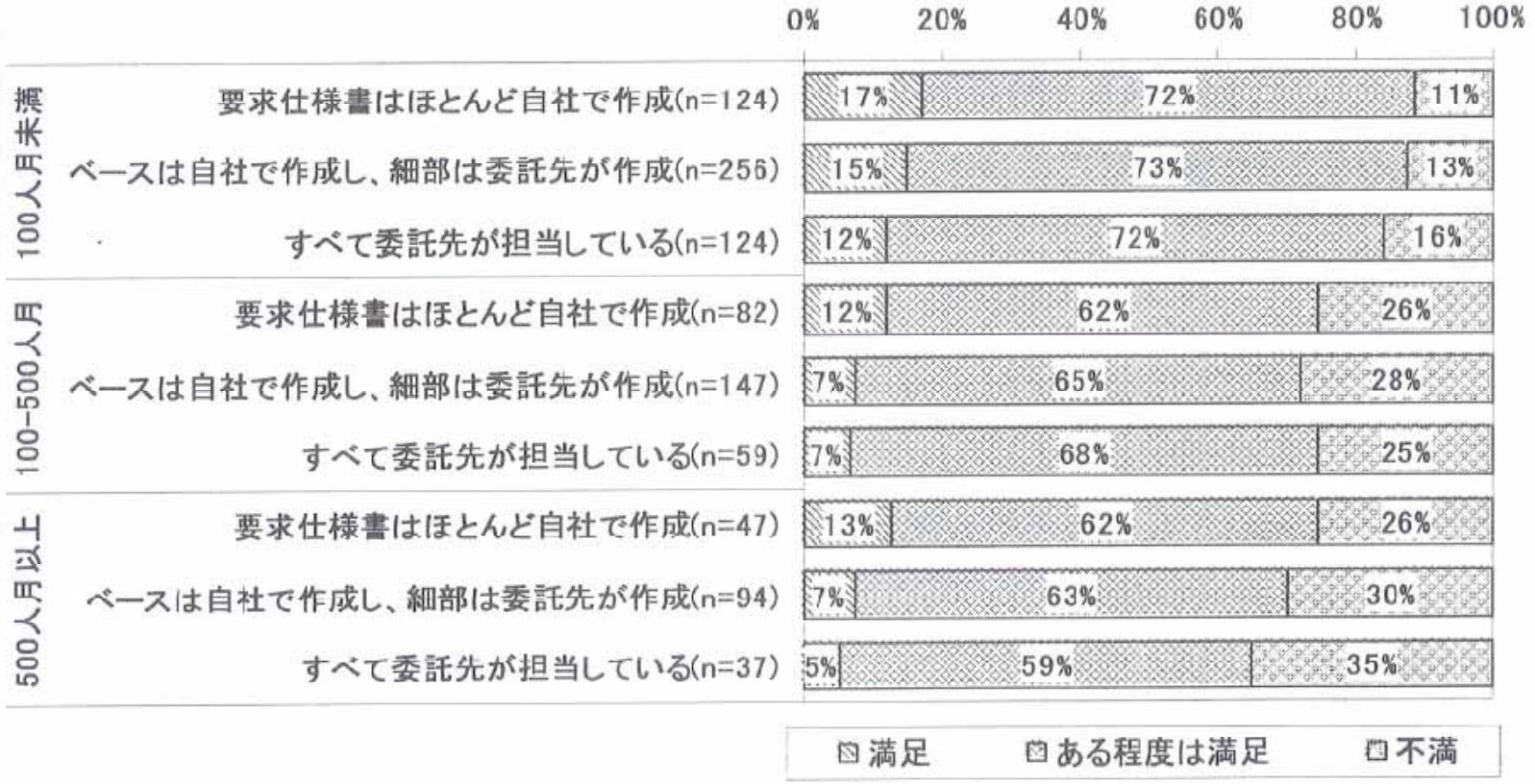
- CMMが出てきたときの企業の反応
  - ISO9000を持っているのになんで必要なんだ？
    - CMMはマイルストンではなくプロセスで品質を確保する
- CMM認証企業への質問
  - あなたは何にかかわりましたか？
  - CMM取得部門からの支援とは具体的になんですか？
- ISO9000取得企業への質問
  - ドキュメント以外での品質管理は何をしていますか

品質は人で担保する

- 品質管理に知見のある人をPMにすることで品質に関する意識を高めることができる。
- チームをきちんと見極める

- 品質担当者は誰ですか
- 品質管理はどのように行いますか
- 品質指標値はありますか

要求仕様書(RFP)作成担当と品質の満足度



品質･信頼性を向上させるためのアプローチ

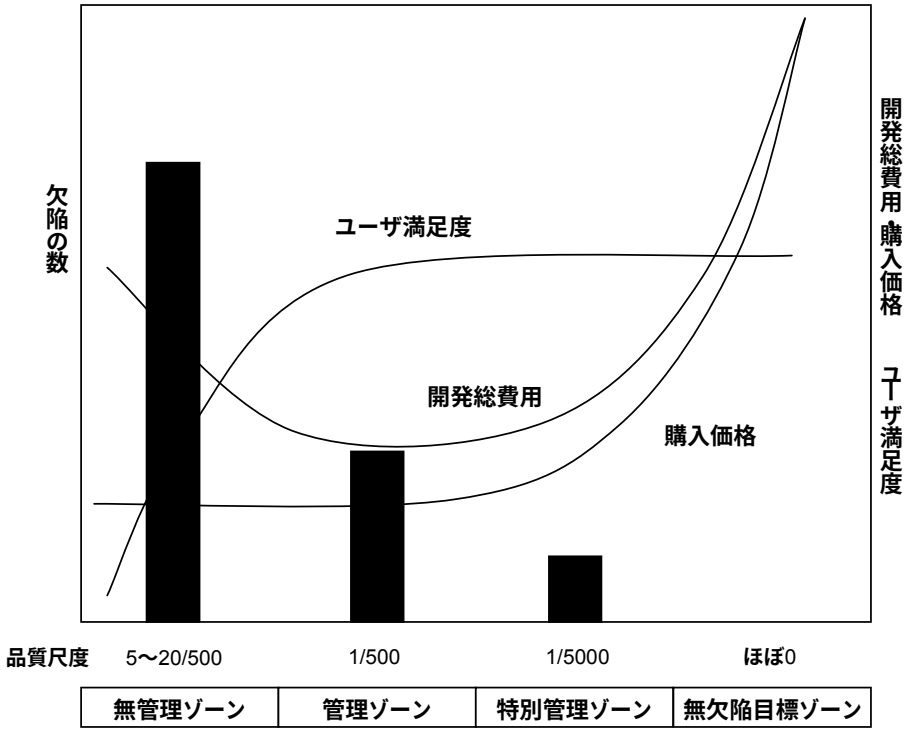
- ドキュメンテーションの充実
- ソフトウェアメトリックスの利用
  - 詳細は、ソフトウェア開発データ活用ガイドブック

- チェックのポイント
  - 入りを制する
    - おかしな計画になっていないか
  - 流れを制する
    - 外れた方向に進んでいないか
  - 出を制する
    - 受け入れてよいのか

- 契約段階でベンダを巻き込んでいくことが重要

品質はむやみに要求すればよいと言う話ではない

- まず始めに、品質と価格のバランスを理解する必要がある。高度な品質を求めると、それに対応して指数関数的に開発費用がかかるようになる。また、逆に品質が低いと対応費用が必要となり、安く開発を行ったとしても結局は高いコストが必要になることも多い。ユーザ満足度を意識して品質レベルを決めていく必要がある。
  - セキュリティについても同様のことが言えるので、コストと信頼性のバランスを熟考の上、セキュリティレベルを考えていく必要がある。



注1 品質尺度: (納入時から安定稼働期迄の欠陥個数)／欠陥費用(万円)
注2 開発総費用と購入価格のギャップはテスト結果の確認、修正結果の確認のために要するユーザ側の付加増加費用をイメージ化したもの

Good Enough Quality
を目指す

出典:「「システム・リファレンス・マニュアル(SRM)」」JUAS 2005

参考：稼働率の目標

	SLA項目	レベル１	レベル２	レベル３	レベル４	レベル５
1	稼働率	98%以下	99%	99.9%	99.99%	99.999%以上
2	バックアップ機	なし	あり （部分的）	あり （2／N＋1台 ）	あり (Hot stand by)	あり (Hot stand by)
3	サービス停止時間 （　）時間／年	172時間	86時間	8.6時間	50分	5分
4	到着時間	1-6時間（昼 ） 12時間（夜間 ）	1-6時間	1-3時間（昼 ） 6時間（夜間 ）	常駐 ケースによっ ては2時 間	常駐
5	修復時間 ・故障修復 ・再立ち上 げ	6時間-12時間 10分-1時間	6時間-12時間 10分-1時間	3時間-6時間 10分-1時間	3時間-6時間 0分-10分	3時間-6時間 即時
6	費用　・構築 費用	1.0倍 1.0倍	1.2～1.8倍 1.1～1.3倍	1.2～3倍 1.3～2.0倍	1.5～4倍 2.0～3倍	4～6倍 3～4倍
7	システム構成（例 ） 必要な機能		NAS	SAN、NAS クラスタリン グ ﾛｰﾄﾞﾊﾞﾗﾝｼﾝｸﾞ	SAN クラスタリン グ ﾛｰﾄﾞﾊﾞﾗﾝｼﾝｸﾞ 二重化	SAN クラスタリング ﾛｰﾄﾞﾊﾞﾗﾝｼﾝｸﾞ 三重化、四重化
8	ペナルティ			対象	対象	対象
9	(参考)アプリ開発費 用	1倍	1倍	1～2倍	2~3倍	3~5倍

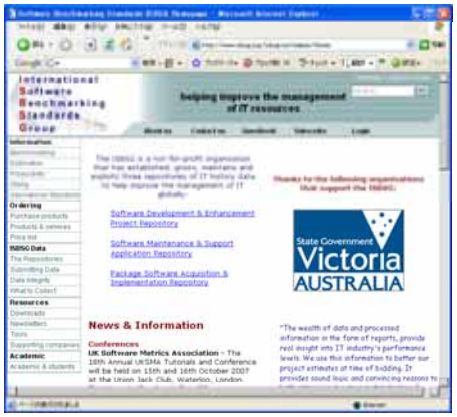
各種ソフトウェアメトリックス調査

- ソフトウェアに関しては多くのデータが公表されている。
  - ユーザ視点での評価
    - JUAS (日本情報システムユーザ協会)
  - ベンダ視点での評価
    - IPA (情報処理推進機構)
    - 日科技連
  - 国際調査
    - ISBSG (International Software Benchmarking Standards Group)

JUAS
http://www.juas.or.jp/

IPA
http://www.ipa.go.jp/

日本科学技術連盟
http://www.juse.or.jp/

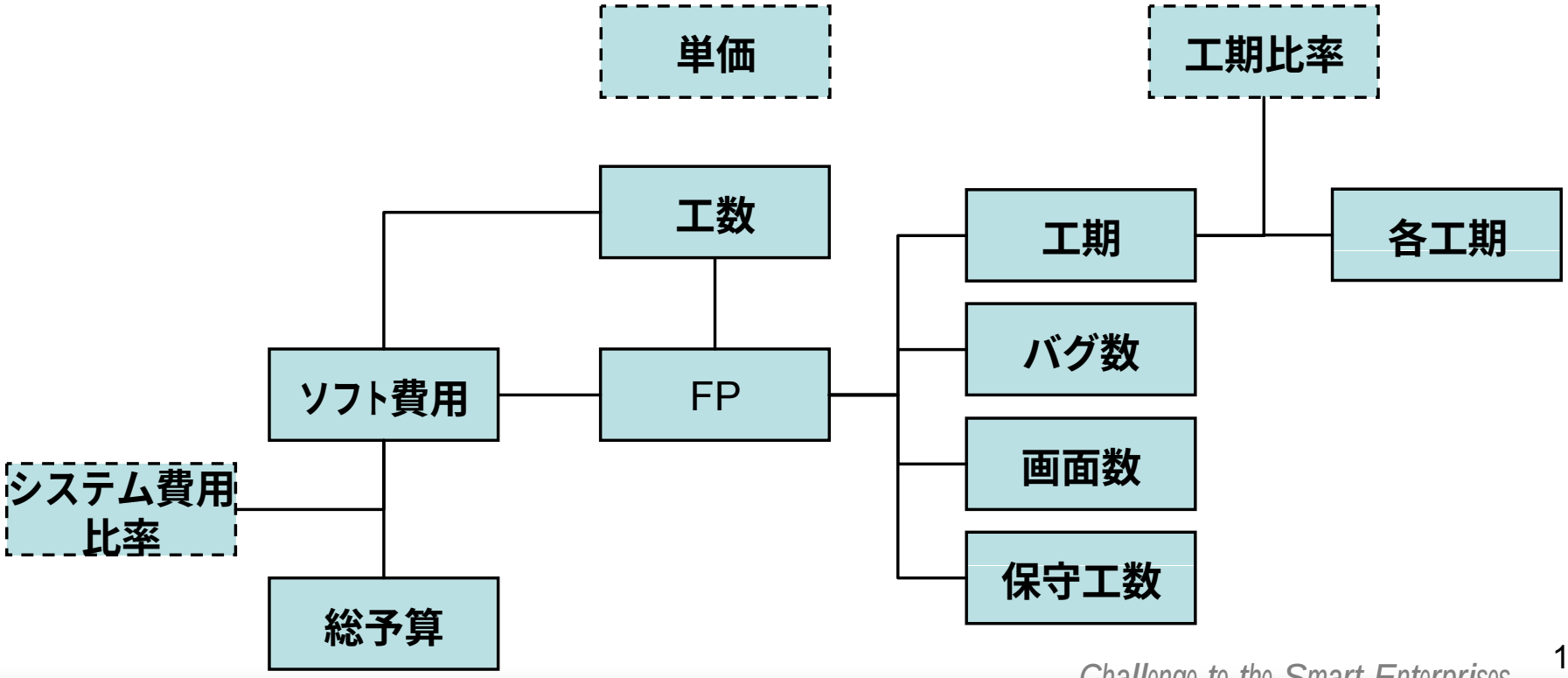


ISBSG
http://www.isbsg.org/

各種学会
・情報処理学会
・ソフトウェア科学会

各種データの関係

- ソフトウェアの開発はこれまで経験に依存するところが多かったが、ベンダ内では生産データを蓄積し、異常プロジェクトの検出に努力している。これらを共通化しようという取り組みも始まっており、情報処理推進機構ではソフトウェア開発白書を刊行している。また、情報システムユーザ協会では経済産業省と連携して、ユーザから見た生産性データの蓄積をソフトウェアメトリックス２００７として作成している。

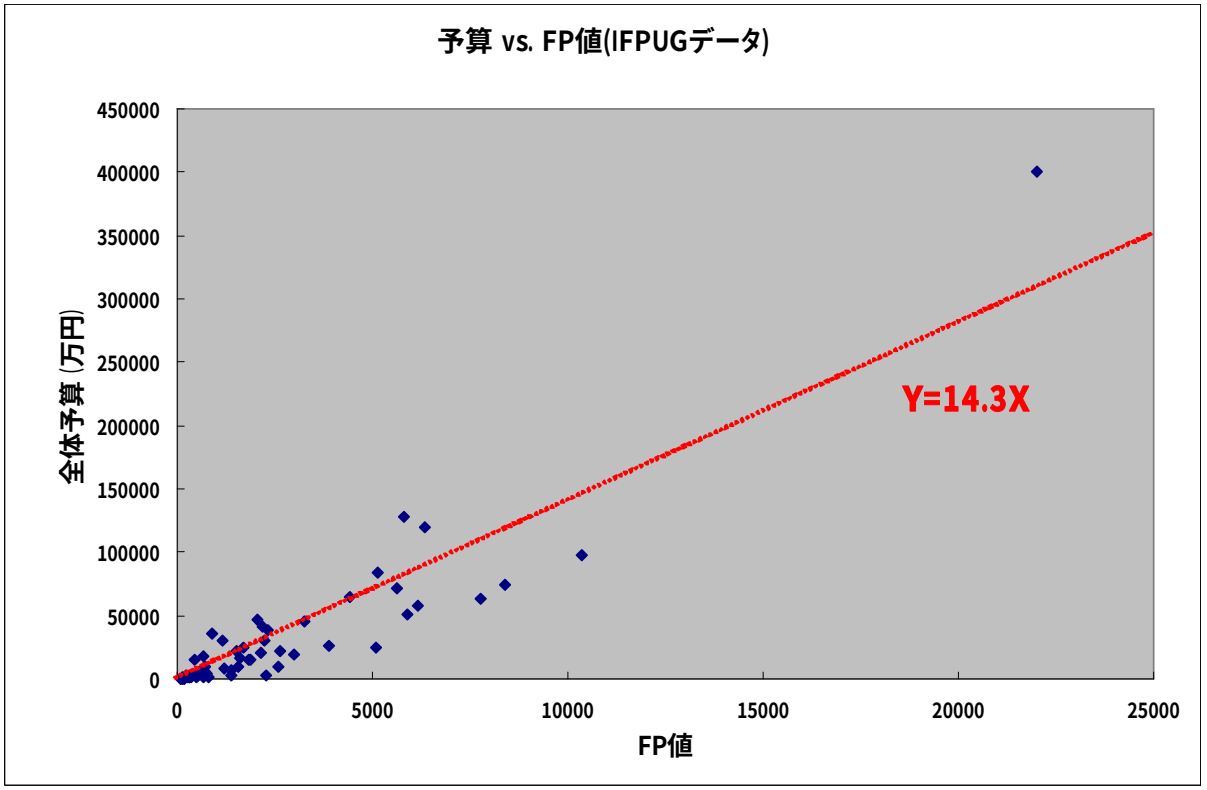


過去のデータを分析する意味

- 今後、ソフトウェアメトリックスを活用していくためには、過去のデータを分析することが重要である。
  - 自組織の弱点を把握
    - システムが適正価格で購入されてきたのか？
    - これまで失敗したプロジェクトの原因はなんだったのか
  - 自組織の傾向を把握
    - 統計とは違った自社、業界特有の傾向を知る
    - 今後の対策用に整理する
  - ソフトウェアメトリックス活用の練習
    - 開発中のシステムにいきなり導入すると、現場の理解も得られないし、実施している本人も混乱する。

基礎データとしてのFPの算出

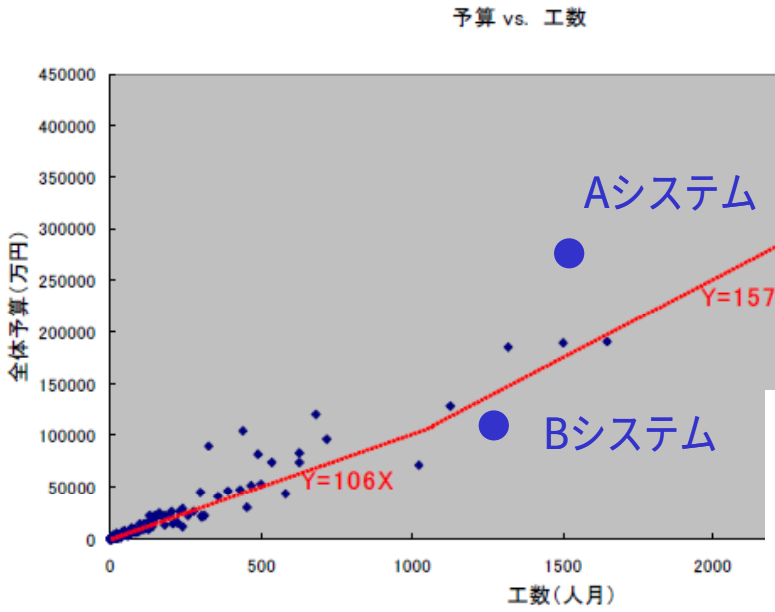
- FPでないシステムは、予算やステップ数を元に概算のファンクションポイントを推定する。ファンクションポイントを基軸にすることで、開発中の各種情報を横串の比較、推定することが可能になる。



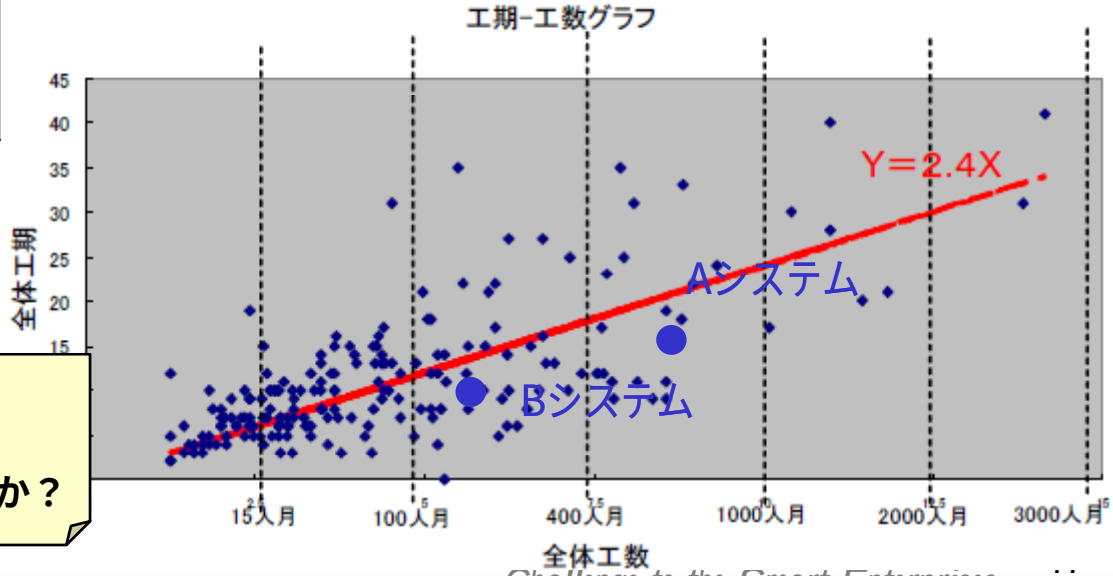
「ソフトウェアメトリックス２００８」JUAS

過去データ分析の例

- 高い買い物か安い買い物かわかる！
- 適正工期かわかる！



契約によって契約単価がバラバラ
・難易度が違ったのか？
・既存か新規か？
・ぼったくられたか？
・買い叩いたか？
・品質との関係はどうなっているのだろうか

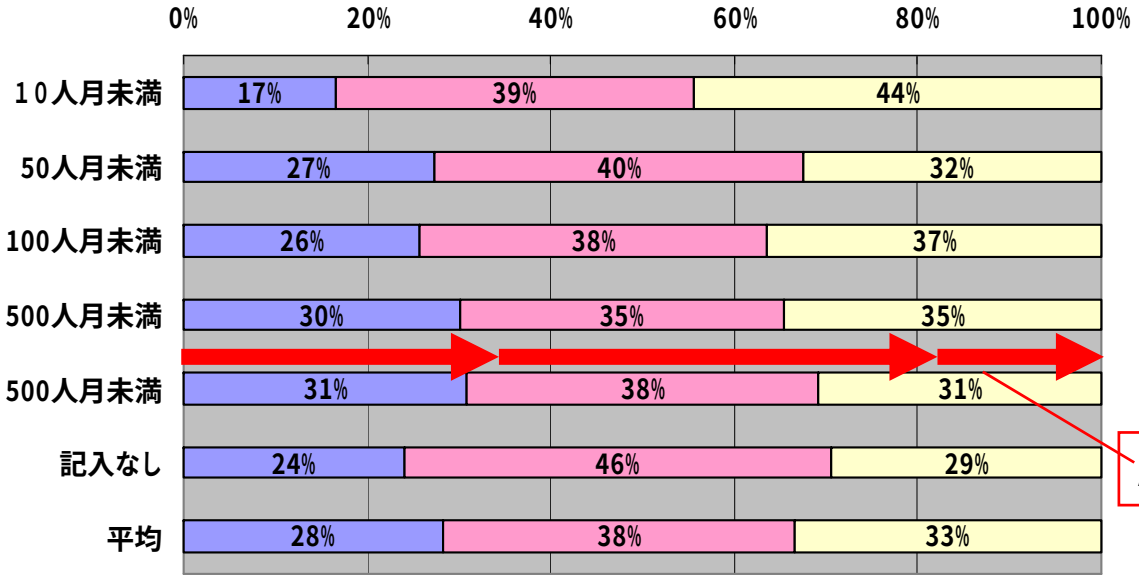


全般的に短工期開発
・品質は大丈夫か？
・人を突っ込みすぎているのではないか？

「ソフトウェアメトリックス２００７」JUAS

過去データ分析の例

◆ 工期配分が適正かわかる！



実装工期が長くテスト工期が短い

テスト工程が短すぎる
・品質は大丈夫か？
・計画時のスケジュールはどうなっていたのか？
・何か遅延の原因があったのか？

「ソフトウェアメトリックス2007」JUAS

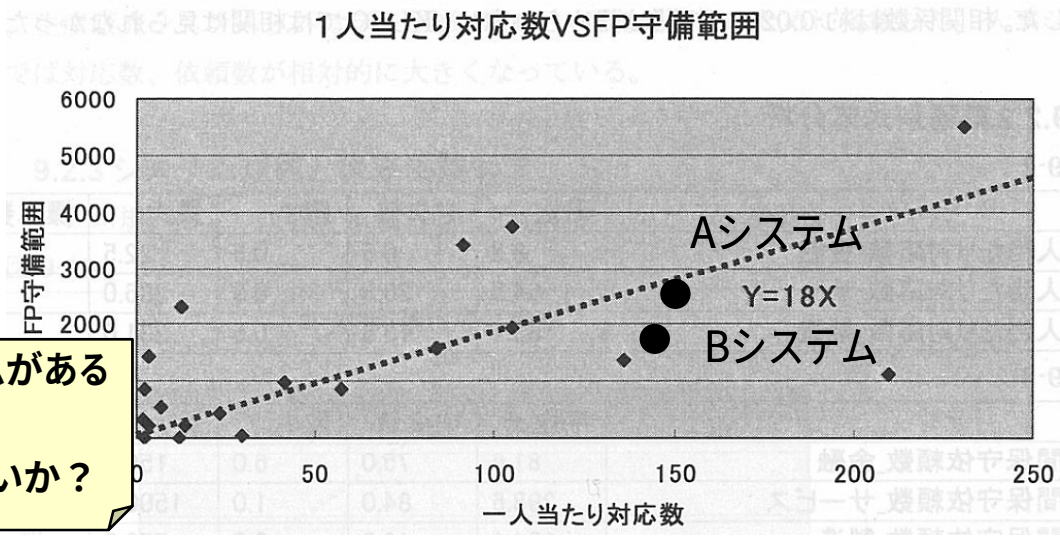
過去データ分析の例

- 適正な品質レベルかわかる！
- 保守工数が適正かわかる！

	Aランク	Bランク	Cランク	Dランク	Eランク	Fランク
欠陥率	0	0.25未満	0.5未満	1未満	3未満	3以上
割合	9.7%	33.1%	18.8%	17.5%	14.9%	5.8%
件数	15	51	29	27	23	9

Aシステム　　Bシステム

導入時の品質が悪いものがある
・目標品質は？
・難易度は？
・欠陥レベルは？



FP保守範囲が狭いのに対応数が多いシステムがある
・納品品質が低かったのではないか？
・運用マニュアルが整備されていないのではないか？

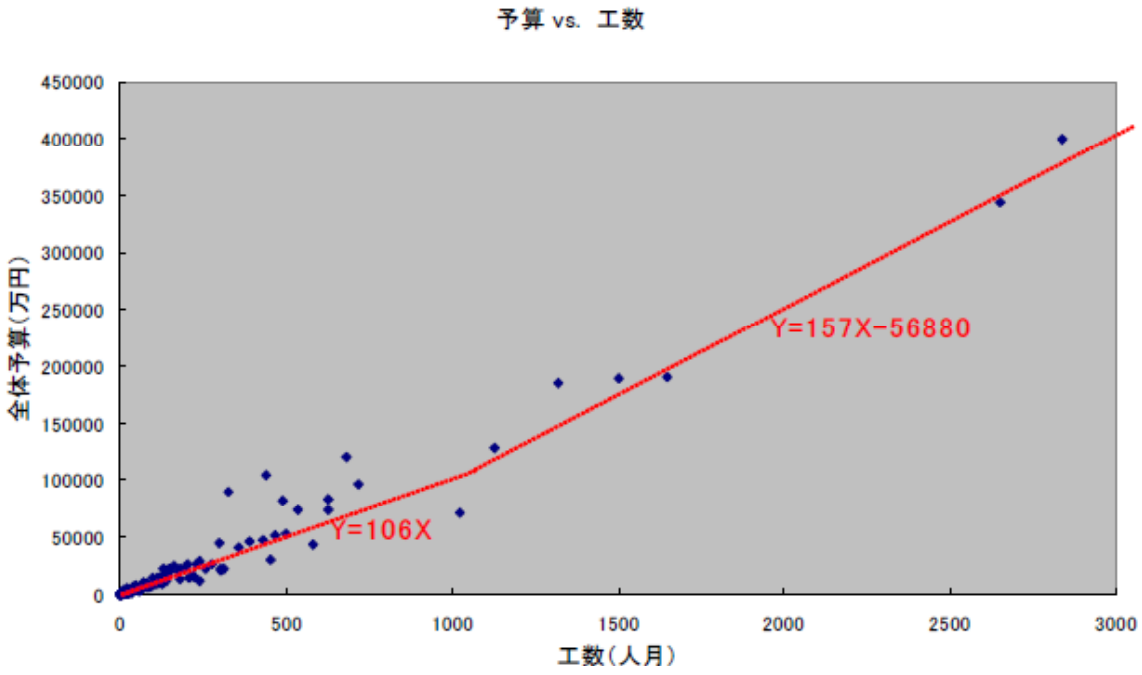
「ソフトウェアメトリックス２００７」JUAS

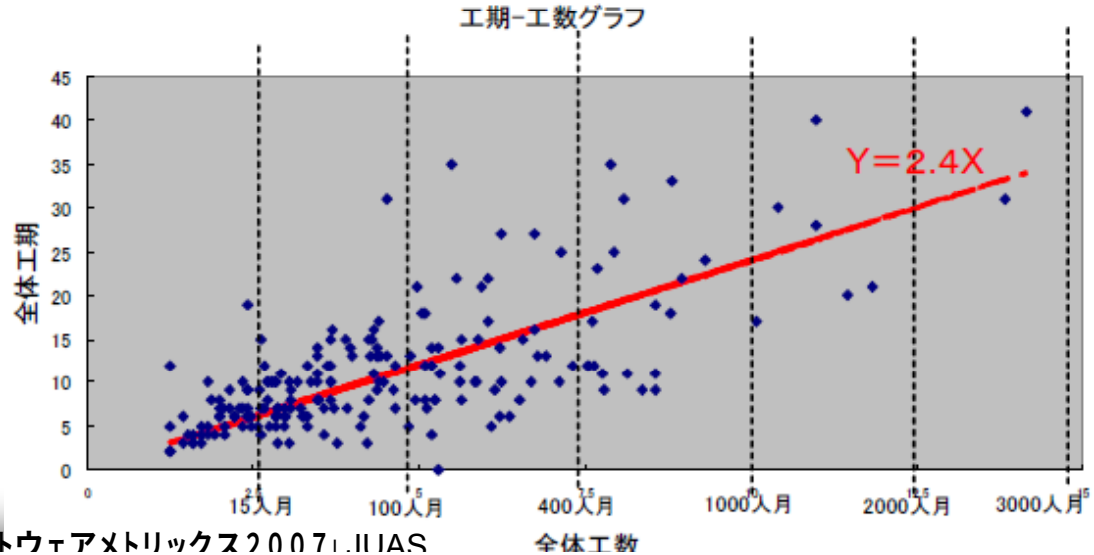
演習

- システム概要
  - 3部門で持っていた販売システムを統合し、顧客情報を一元管理するシステムであり、先進の技術を導入しリアルタイムレスポンスを実現するシステムである。
- データ
  - 予算　25000万円 (設計から導入まで)
  - 工数　350人月
  - 計画工期　10ヶ月 (設計:実装:テスト＝2ヶ月:4ヶ月:4ヶ月)
  - 実績工期　10ヶ月 (設計:実装:テスト＝3ヶ月:5ヶ月:2ヶ月)
  - 欠陥率 (フォロー不具合数:総合テスト2での不具合数)　0.63 (62:32)
  - FP保守範囲　750FP/人
  - 一人当たり対応数　82件/人
- 利用部門の感想
  - 時々止まるが、前のシステムもこんなものだったのであきらめている。
- システム部門の感想
  - 他のシステムに較べて、手間がかかる気がする

- 問題
  - 各データをグラフ上にプロットしてください
  - そのシステムの現状に関する評価をしてください
  - どこでプロジェクトマネージャは手を打つべきだったか考えてください

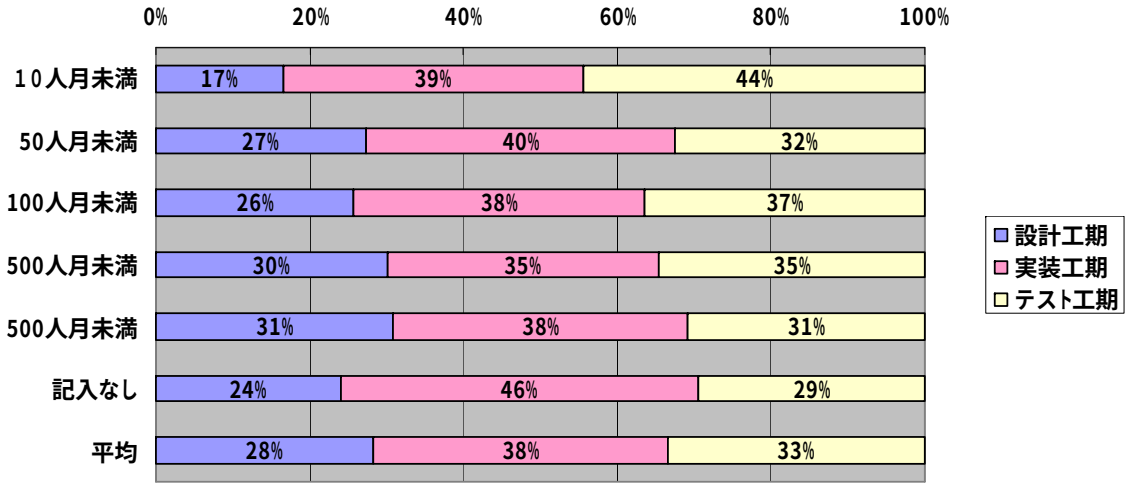
演習



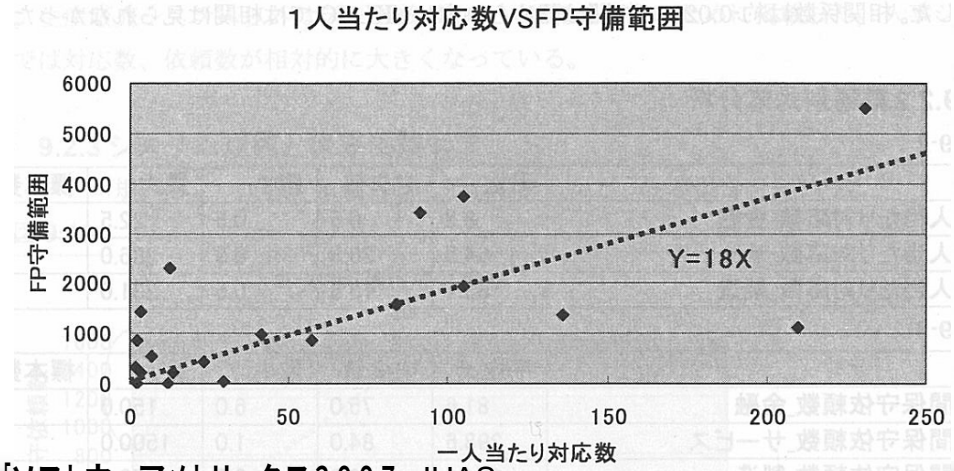


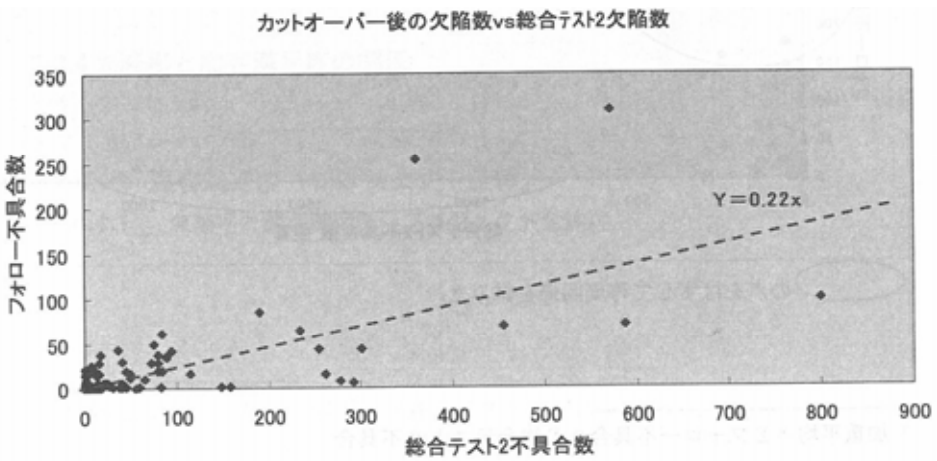
「ソフトウェアメトリックス2007」JUAS

演習



	Aランク	Bランク	Cランク	Dランク	Eランク	Fランク
欠陥率	0	0.25未満	0.5未満	1未満	3未満	3以上
割合	9.7%	33.1%	18.8%	17.5%	14.9%	5.8%
件数	15	51	29	27	23	9





「ソフトウェアメトリックス２００７」JUAS

メトリックス調査に補正は必要か？

- ソフトウェアのメトリックスデータは統計的に処理されているので、もっと詳細なデータが知りたいことがある。
- たとえば、言語により生産性が大きく異なることが知られている。

言語	レベル	LOC／FP
アセンブラ	1.00	320.00
C	2.50	128.00
C＋＋	5.00	64.00
COBOL	3.00	106.67
FORTRAN	3.00	106.67
PowerBuilder	20.00	16.00
SQL	25.00	12.80
VB	10.00	32.00
平均	7.63	92.35

「ソフトウェア見積もりの全て」共立出版
平均は、adaなどの言語を含む15言語の平均
レベルは、アセンブラを1としたときの生産性

Source Code Language	Conversion Ratio (LOC Per Function Point)
Basic Assembly	575
JCL	400
Macro Assembly	400
C	225
Cobol74(Cobol I)	220
FORTRAN	210
Cobol85(Cobol II)	175
Pascal	160
PL/1	126
RPG I	120
RPG II/III	110
Natural	100
C++	80
Java	80
dBaseIII	60
Focus	60
Clipper	60
oracle	60
Sybase	60
dBaseIV	55
Perl	50
JavaScript	50
VBScript	50
Shell Script	50
SAS	50
APL	50

資料
DCG

KLOC/コンバージョンレシオ＝FP　でコンバージョン可能

様々な補正が可能

- アーキテクチャー別の補正値
- 新規開発、再開発別の補正値
- 言語別の品質

SLOCあたりの不具合数

言語	平均	標準偏差
COBOL	0.075	0.160
C	0.051	0.100
VB	0.103	0.206
Java	0.122	0.330

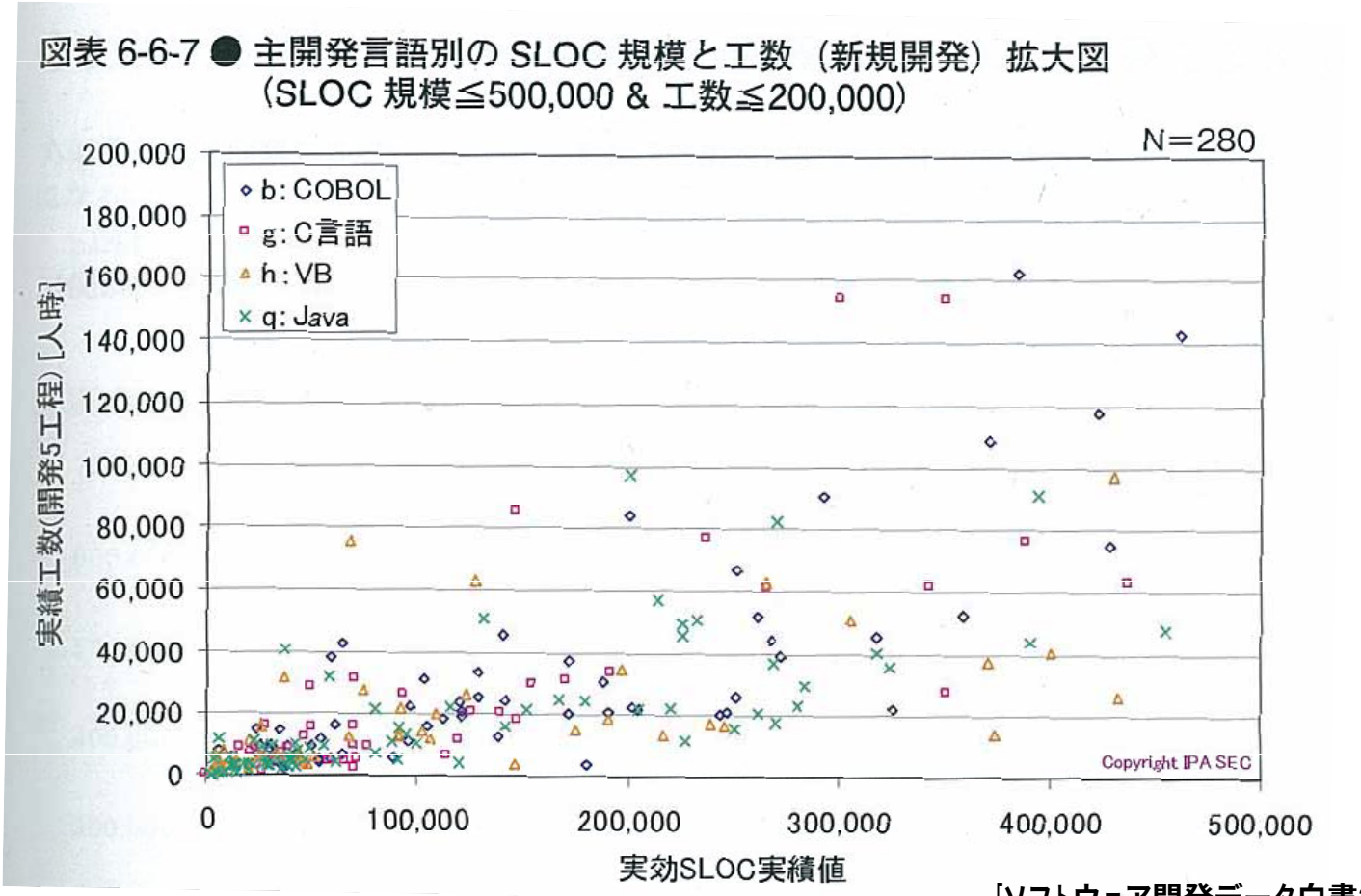
SLOCあたりの不具合数

言語	平均	標準偏差
新規開発	0.119	0.375
改修･保守	0.175	0.694
再開発	0.181	0.417
拡張	0.077	0.192

「ソフトウェア開発データ白書2007」IPA

BUT! 現段階では補正をしなくてもよい

- エンジニアリングはまだ夜が明けたばかり。



「ソフトウェア開発データ白書2007」IPA

欠陥除去には何が有効か

- 全ての工程できちんと検査をしていくことが重要なのはもちろん、前工程で問題点を潰していくことが重要。
- 後工程でバグが見つかると、上流工程で発見されたバグの数倍の労力（コスト、時間）がかかる。
  - 要求段階での修正コストを1とすると、アーキテクチャ設計時で3、システムテスト段階で10、出荷後で10～100倍と言われている

		欠陥除去率の最悪の場合の結果（%）		
		最小	中央値	最大値
設計インスペクション	×	30	40	50
コードインスペクション	×			
品質保証	×			
正規のテスト	×			

		欠陥除去率の最悪の場合の結果（%）		
		最小	中央値	最大値
設計インスペクション	×	32	45	55
コードインスペクション	×			
品質保証	○			
正規のテスト	×			
設計インスペクション	×	37	53	60
コードインスペクション	×			
品質保証	×			
正規のテスト	○			
設計インスペクション	×	43	57	66
コードインスペクション	○			
品質保証	×			
正規のテスト	×			
設計インスペクション	○	45	60	68
コードインスペクション	×			
品質保証	×			
正規のテスト	×			

		欠陥除去率の最悪の場合の結果（%）		
		最小	中央値	最大値
設計インスペクション	×	50	65	75
コードインスペクション	×			
品質保証	○			
正規のテスト	○			
設計インスペクション	×	53	68	78
コードインスペクション	○			
品質保証	○			
正規のテスト	×			
設計インスペクション	×	55	70	80
コードインスペクション	○			
品質保証	×			
正規のテスト	○			
設計インスペクション	○	60	75	85
コードインスペクション	×			
品質保証	○			
正規のテスト	×			
設計インスペクション	○	65	80	87
コードインスペクション	×			
品質保証	×			
正規のテスト	○			
設計インスペクション	○	70	85	90
コードインスペクション	○			
品質保証	×			
正規のテスト	×			

		欠陥除去率の最悪の場合の結果（%）		
		最小	中央値	最大値
設計インスペクション	×	75	87	93
コードインスペクション	○			
品質保証	○			
正規のテスト	○			
設計インスペクション	○	77	90	95
コードインスペクション	×			
品質保証	○			
正規のテスト	○			
設計インスペクション	○	83	95	97
コードインスペクション	○			
品質保証	○			
正規のテスト	×			
設計インスペクション	○	85	97	99
コードインスペクション	○			
品質保証	×			
正規のテスト	○			

		欠陥除去率の最悪の場合の結果（%）		
		最小	中央値	最大値
設計インスペクション	○	95	99	99.99
コードインスペクション	○			
品質保証	○			
正規のテスト	○			

ソフトウェア開発の定量化手法第二版、共立出版

情報システム予算の内訳

- **予算におけるハードウェア、ソフトウェア、パッケージソフト、通信、運用などサービス、省内担当職員人件費、その他は、平成17年度情報処理実態調査から、企業においては以下の比率になる。**
  - **この数値をもとに、要求されている予算の妥当性を検証していく、但し、あくまでも企業における平均値であり、システムの特性に寄って費用比率は変わってくるのであくまで上記の数値は参考値である。**

分類	比率	備考
ハードウェア	１７．４％	買い取り、減価償却、レンタル、リース等
ソフトウェア	３１．５％	買い取り、減価償却、レンタル、リース、開発・カスタマイズ関連費用等
通信関連	３．８％	
運用などサービス	２５．２％	データ入力、運用･保守委託料、教育、外部派遣要員等
職員人件費	１４．２％	
その他	７．５％	

工期と満足度の関係

- 標準的な工期に対して短工期、長工期の分布が25%になるようにして分析を行うと以下の結果がある。標準的な工期に対して短工期で行う場合にはリスクが高まることから、短工期開発にする理由を確認するとともに、短工期開発で行う場合のリスク回避対策（優秀なPMを配置する等）を提示させる必要がある。

工期乖離度		顧客満足度(プロジェクト全体)					
		満足	やや不満	不満	未回答	総計(割合)	
長工期	件数	34	13	0	3	50	24.8%
	割合	68.0%	26.0%	0.0%	6.0%	100.0%	
適正工期	件数	68	28	4	4		51.5%
	割合	65.4%	26.9%	3.8%	3.8%	100.0%	
短工期	件数	29	15	2	2	48	23.8%
	割合	60.4%	31.3%	4.2%	4.2%	100.0%	
総計	件数	131	56	6	9	202	100.0%
	割合	64.9%	27.7%	3.0%	4.5%	100.0%	

工期短縮に対する対策

◆ 工期短縮に対しては適切な対応を行う必要がある。

尺　度	25%工期延長	標　　準	25%工期短縮	25%以上工期短縮
式	3.4×(人月)$^{1/3}$	2.7×(人月)$^{1/3}$	2.0×(人月)$^{1/3}$	1.4×(人月)$^{1/3}$
工期設定の考え方	金融等欠陥の発生を無くしたい品質重視	提案モデル	・顧客の要望 ・流通業	外的対応／工期制約 対コンペ戦略、新商品の販売、株式の上場、企業の統合など
工程管理上の対応策	充分なシステムテスト期間の確保	中日程計画の充実 （役割分担別WBS管理）	中日程計画の充実 （週間別管理）	小日程計画の充実 （日別管理）
その他の対応策	・品質重視のテスト計画書、テストケースの緻密化 ・安定稼動のための分割立上 等	・WBSによる全体計画と局面化開発の徹底 ・レビューの徹底 ・テストケース充実 ・コンバージョンデータのフル活用 ・確実な変更管理	同　左 ＋ （下　記） ・標準化の徹底と実力のある一括外注の採用。 ・システム範囲、対象の部分稼動 ・RAD+DOA ・性能事前検証 ・変更管理の強化	同　左 ＋ （下　記） ・ベテランプロマネの投入と会社あげての協力及び監視 ・パート図での計画 ・ベストメンバー選出 ・クリーンルーム手法 ・二交代制の配置 ・開発促進ツールの採用 ・顧客主体のテストチーム設置 ・パッケージの活用 ・部品の再利用 ・オープンな進捗情報管理

注　WBS：Work Breakdown Structure
RAD: Rapid Application Development
DOA: Data Oriented Approach

図表 5-1-15　工期評価尺度と対策

出典：「「システム・リファレンス・マニュアル（SRM）」JUAS,2005

仕様の明確度と満足度の関係

- 情報システムの調達において、仕様が曖昧なことがよく指摘されるが、仕様は明確に定義することが重要である。下表のように、仕様が曖昧であると工期遅延を発生することが多くなり、満足度も低くなることが多い。このような傾向を念頭に置き、仕様作成時には仕様の明確さを、プロジェクトメンバ、リーダ、プログラムマネジメントオフィスの各段階で十分に確認する必要がある。

仕様明確度		工期遅延度							遅延度20%以上の割合
		予定より早い	予定通り	10%未満	20%未満	50%未満	それ以上	総計	
非常に明確	件数		15	2	1	2		20	10.0%
	割合		75.0%	10.0%	5.0%	10.0%		100.0%	
	平均工期遅延率		0.0%	6.7%	11.1%	42.9%		5.1%	
かなり明確	件数	7	75	8	7	6	1	104	6.7%
	割合	6.7%	72.1%	7.7%	6.7%	5.8%	1.0%	100.0%	
	平均工期遅延率	-29.68%	0.00%	6.18%	13.96%	26.13%	66.67%	1.57%	
やや曖昧	件数	7	38	8	5	10	4	72	19.4%
	割合	9.7%	52.8%	11.1%	6.9%	13.9%	5.6%	100.0%	
	平均工期遅延率	-31.3%	0.00%	6.65%	14.22%	27.27%	54.17%	5.48%	
非常に曖昧	件数		3			3	1	7	57.1%
	割合	0	42.9%	0.0%	0.0%	42.9%	14.3%	100.0%	
	平均工期遅延率		0.00%			41.61%	50%	24.98%	
合計	件数	14	131	18	13	21	6	203	13.3%
	割合	6.9%	64.5%	8.9%	6.4%	10.3%	3.0%	100.0%	
	平均工期遅延率	-30.47%	0.00%	6.47%	13.84%	29.63%	55.56%	4.07%	

仕様明確度		顧客満足度(プロジェクト全体)				
		満足	やや不満	不満	未回答	総計
非常に明確	件数	18	1		1	20
	割合	90.0%	5.0%	0.0%	5.0%	100.0%
かなり明確	件数	85	27	2	6	120
	割合	70.8%	22.5%	1.7%	5.0%	100.0%
やや曖昧	件数	40	30	5	2	77
	割合	51.9%	39.0%	6.5%	2.6%	100.0%
非常に曖昧	件数	3	3		1	7
	割合	42.9%	42.9%	0.0%	14.3%	100.0%
合計	件数	146	61	7	10	224
	割合	65.2%	27.2%	3.1%	4.5%	100.0%

「ソフトウェアメトリックス２００７」JUAS

仕様の明確度とプロジェクト遅延

- 更に仕様変更であるが、仕様変更が大きなものは遅延も生じやすく満足度にも問題が生じがちであるので、このような傾向を念頭に置き、仕様作成時に仕様内容を、プロジェクトメンバ、リーダ、プログラムマネジメントオフィスの各段階で十分に確認する必要がある。

仕様変更発生度		遅延度							遅延度20%以上の割合
		予定より早い	予定通り	10%未満	20%未満	50%未満	それ以上	総計	
変更なし	件数	1	6			2		9	22.20%
	割合	11.11%	66.7%	0.0%	0.0%	22.2%	0.0%	100.0%	
	平均工期遅延率	-71.4%	0.0%			32.2%		-0.8%	
軽微な変更が発生	件数	9	102	9	8	12	3	143	10.50%
	割合	6.29%	71.3%	6.3%	5.6%	8.4%	2.1%	100.0%	
	平均工期遅延率	-30.31%	0.00%	7.00%	12.93%	29.41%	55.56%	2.89%	
大きな変更が発生	件数	4	22	9	4	6	3	48	18.80%
	割合	8.33%	45.8%	18.8%	8.3%	12.5%	6.3%	100.0%	
	平均工期遅延率	-20.6%	0.00%	5.94%	16.35%	26.11%	55.56%	7.50%	
重大な変更が発生	件数					2		2	100.00%
	割合	0.00%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	
	平均工期遅延率					36.03%		36.03%	
合計	件数	14	130	18	12	22	6	202	13.90%
	割合	8.2%	63.9%	7.4%	5.7%	11.5%	3.3%	100.0%	
	平均工期遅延率	-30.47%	0.00%	6.47%	14.07%	29.37%	55.56%	4.15%	

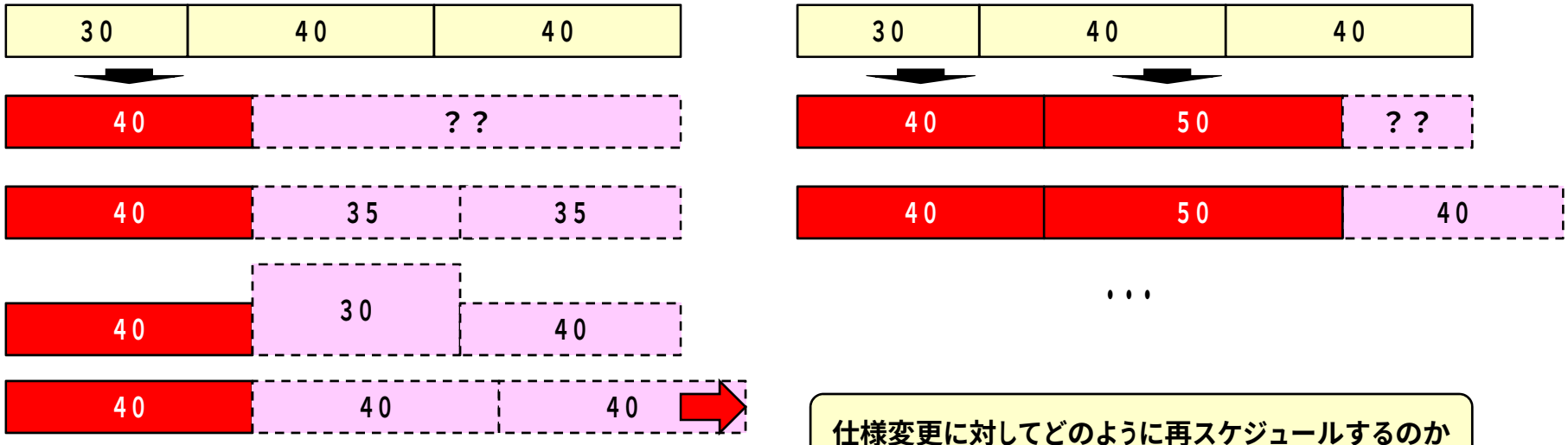
仕様変更発生度		ユーザ満足度（プロジェクト全体）				
		満足	やや不満	不満	未回答	総計
変更なし	件数	8	3			11
	割合	72.7%	27.3%	0.0%	0.0%	100.0%
軽微な変更が発生	件数	109	38	2	8	157
	割合	69.4%	24.2%	1.3%	5.1%	100.0%
大きな変更が発生	件数	28	17	5	3	53
	割合	52.8%	32.1%	9.4%	5.7%	100.0%
重大な変更が発生	件数	1	1			2
	割合	50.0%	50.0%	0.0%	0.0%	100.0%
合計	件数	146	59	7	11	223
	割合	65.5%	26.5%	3.1%	4.9%	100.0%

「ソフトウェアメトリックス２００７」JUAS

仕様変更と欠陥率

- 当然のことながら仕様変更が発生したものは変更しないものに対して作業にゆがみが生じることから欠陥率も大きくなる傾向がある。

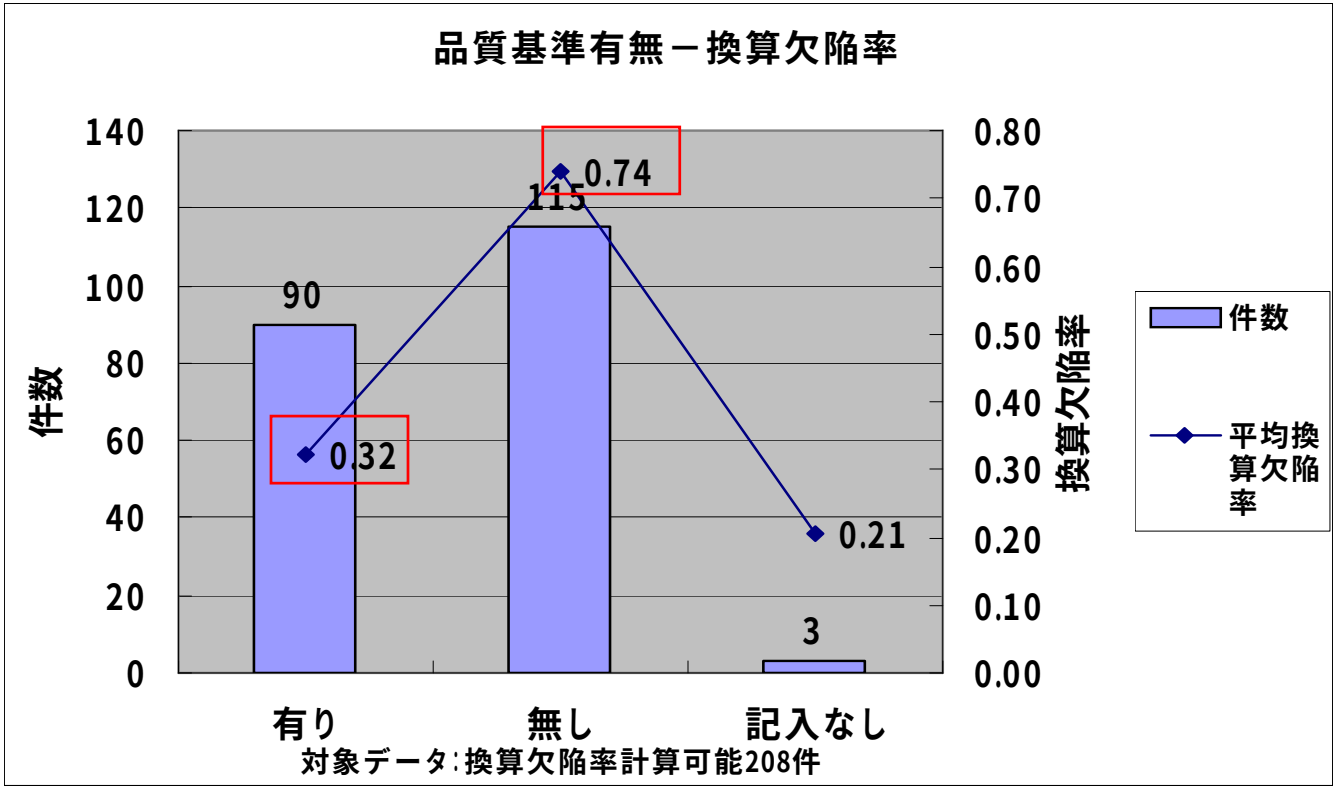
仕様変更発生度	件数	平均換算欠陥率	最大欠陥率
変更なし	8	0.36	0.73
軽微な変更が発生	140	0.57	11.89
大きな変更が発生	56	0.54	4.38
重大な変更が発生	1	0.03	0.03
合計	205	0.81	11.89



「ソフトウェアメトリックス2008」JUAS

品質基準の有無と満足度

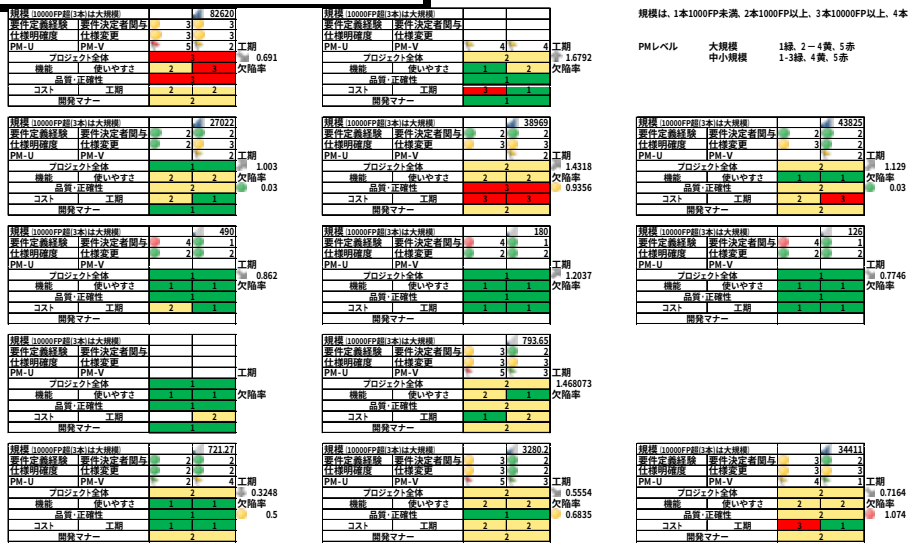
◆ 品質基準があるものは、やはり品質がよい。



参考：満足度を意識する

- 品質と正確性はまだ満足度向上の余地があり、それが、プロジェクト全体の満足度向上にも貢献できるものと考えられる。
- 意識して取り組んでいく必要がある

プロジェクト全体		63%	
機能	使いやすさ	75%	69%
品質・正確性		57%	
コスト	工期	51%	64%
開発マナー		62%	



ベンダPMの重要性

- また、PMスキルが高いほど欠陥率が低いとの傾向がある。特に未経験者がPMの場合には、途中の確認を行うなど注意が必要である。

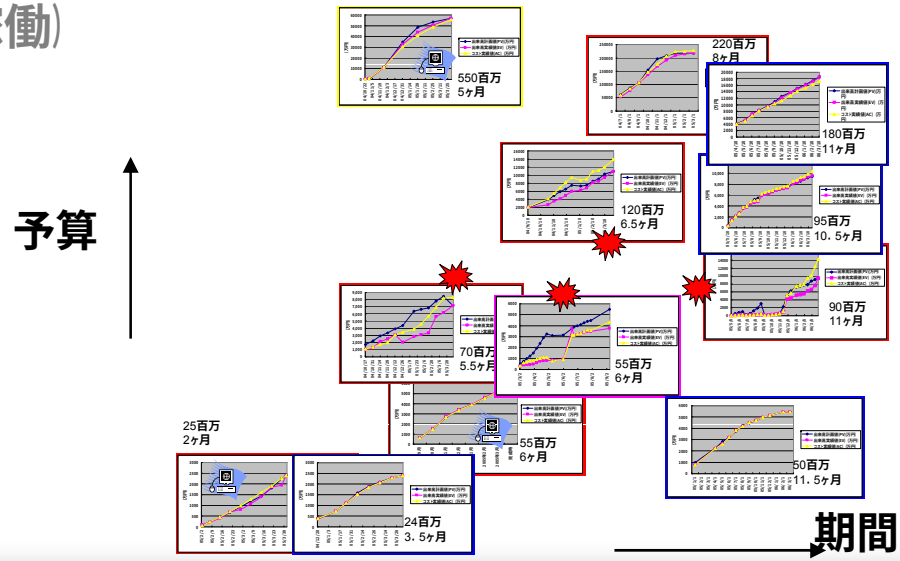
		PMスキル(ベンダ)						
		1	2	3	4	5	記入なし	計
換算欠陥率	件数	54	35	54	30	5	30	208
	平均	0.27	0.49	0.7	0.42	1.54	0.82	0.55
	最大	1.69	2.95	9.06	1.83	4.38	11.89	11.89
	最小	0.00	0.00	0.00	0.00	0.02	0.00	0.00

PMスキル
1.多数の中･大規模プロジェクトの管理を経験
2.少数の中･大規模プロジェクトの管理を経験
3.多数の小･中規模プロジェクトの管理を経験
4.少数の小･中規模プロジェクトの管理を経験
5.プロジェクト管理の経験なし

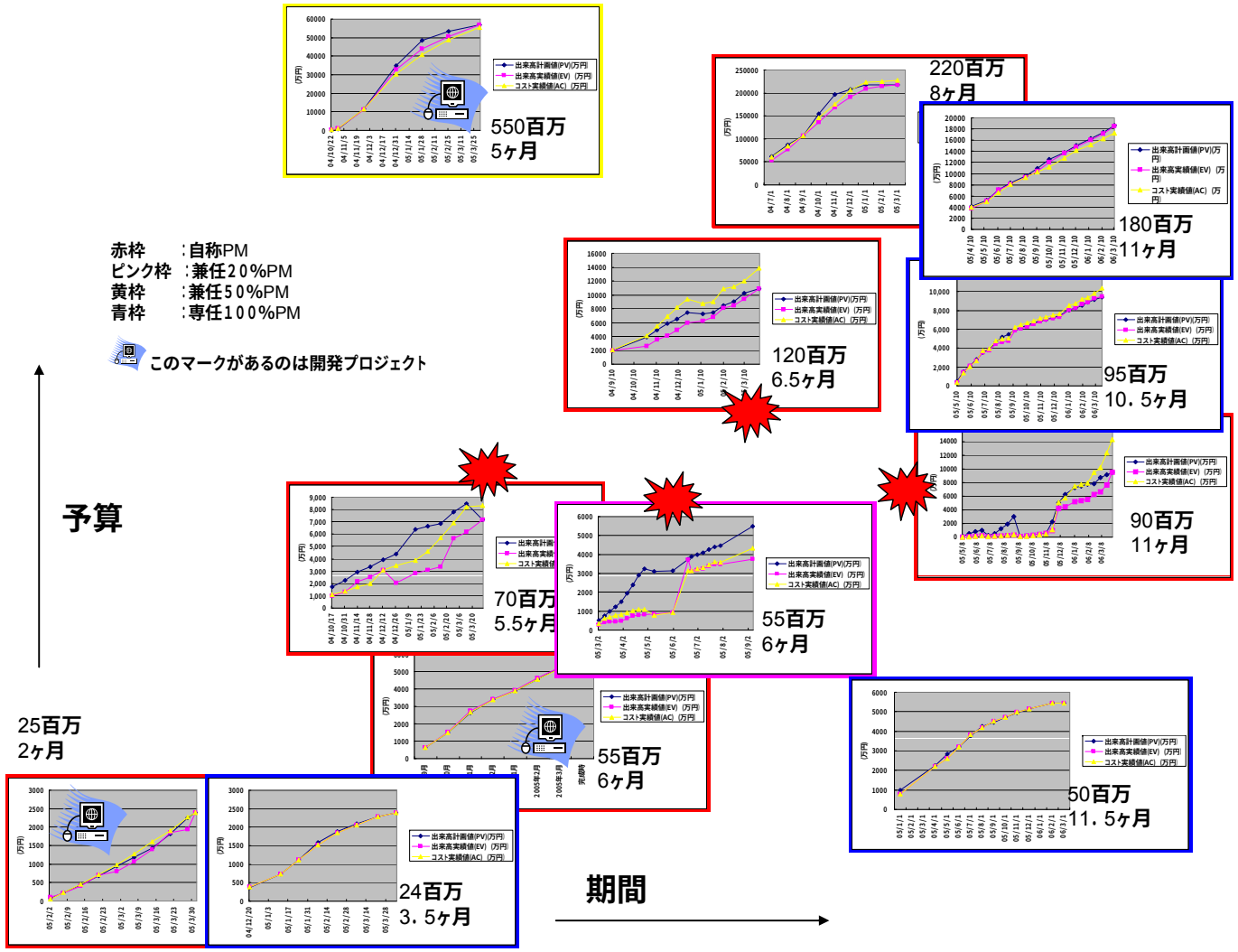
「ソフトウェアメトリックス２００８」JUAS

PMの資格をどう考えたらよいか

- 情報処理技術者やＰＭＰ（Project Management Professional）などのプロジェクト・マネジメント専門技術者の有無とプロジェクト管理手法であるＥＶＭの結果を比較することにより、認定された技術者を導入した場合と導入しない場合の品質の差異などを分析したものがある。下図は、右に行くほど実施期間が長く、上に行くほど開発金額が大きくなるようにEVMのデータをプロットしたものである。（正常なプロジェクトは右肩上がりにグラフが上がっていき、計画と実績があまりずれない）このグラフには業務分析プロジェクトと開発プロジェクトが含まれるが、開発プロジェクトはグラフ中にコンピュータマークを付けているが失敗が少ない傾向がある。それに対し赤で囲まれたプロジェクトは、プロジェクトマネージャの資格は持っていないが自称プロジェクトマネージャが実施しているプロジェクトであるが、多くのプロジェクトが失敗に終わっている。逆に資格者が配置されている青いプロジェクトでは失敗プロジェクトは存在していない。（黄色は資格を持ったPMが50%以上稼働、ピンク色は資格を持ったPMが20%以上稼働）



参考：拡大


赤枠　：自称PM
ピンク枠　：兼任２０%PM
黄枠　：兼任５０%PM
青枠　：専任１００%PM
このマークがあるのは開発プロジェクト
予算
期間
550百万
5ヶ月
220百万
8ヶ月
180百万
11ヶ月
120百万
6.5ヶ月
95百万
10．5ヶ月
90百万
11ヶ月
70百万
5.5ヶ月
55百万
6ヶ月
55百万
6ヶ月
50百万
11．5ヶ月
25百万
2ヶ月
24百万
3．5ヶ月

そのほかの技術者も必要

◆ 政府調達では以下のように定義している。

情報システムに係る工程等	資　　格	ITスキル標準[36]における職種
1．情報システム化計画の策定	・システムアナリスト ・技術士（情報工学部門）	・コンサルタント ・IT アーキテクト
2．要件定義等	・システムアナリスト ・アプリケーションエンジニア ・技術士（情報工学部門）	・コンサルタント ・IT アーキテクト
3．設計・開発 4．結合・総合テスト等	・アプリケーションエンジニア ・ソフトウェア開発技術者 ・テクニカルエンジニア（ネットワーク、データベース又は情報セキュリティ） ・基本情報技術者 ・システム監査技術者 ・技術士（情報工学部門）	・IT スペシャリスト ・アプリケーションスペシャリスト ・カスタマサービス
5．受入テスト 6．移行	・アプリケーションエンジニア ・ソフトウェア開発技術者 ・テクニカルエンジニア（ネットワーク、データベース又は情報セキュリティ） ・基本情報技術者	・IT スペシャリスト ・アプリケーションスペシャリスト ・カスタマサービス

欠陥とは何か

- バグ管理のツールであるBugzillaの定義では以下の通りである。
  - Blocker
    - 開発やテストができ名くるくらい重大なもの
  - Critical
    - クラッシュやデータ損出につながるもの
  - Major
    - 機能が動かないもの
  - Minor
    - 機能は動くが、手続きが必要など完全ではないもの
  - Trivial
    - 機能に影響を与えない記述ミスなど
  - Enhancement
    - 機能強化要望

欠陥はどこまで許されるのか

- システムの品質において、欠陥の含まれる割合は重要であるが、一般的に以下のように分布しており、通常のシステムとしてはBランク以上の品質が望まれ、少なくともCランクは満たしていることが望まれる
- ここで欠陥率とは、「ユーザが発見した欠陥数（顧客側総合テスト以降、フォローまで出発見された欠陥）」÷プロジェクト全体工数である。つまり欠陥率0．25とは4人月あたり1個のバグということである。

	Aランク	Bランク	Cランク	Dランク	Eランク	Fランク
欠陥率	0	0.25未満	0.5未満	1未満	3未満	3以上
割合	9.7%	33.1%	18.8%	17.5%	14.9%	5.8%
件数	15	51	29	27	23	9

欠陥の影響はどのくらいか

- 欠陥はMTTFとも関連があり、それを基に要求品質を設定していく必要がある。

KLOCあたりの欠陥レベル	平均故障間隔（MTTF）	品質レベル（参考）
30以上	2分以下	
20～30	4～15分	
10～20	5～60分	
5～10	1～4時間	
2～5	4～24時間	E～Fクラス相当
1～2	24～160時間	Eクラス相当
1以下	ほとんど故障しない	A～Dクラス相当

プロトタイプレベル

商用ソフトレベル
10-2000件/1000時間

高信頼性ソフトレベル
0.1-10件/1000時間

ソフトウェア開発の定量化手法第二版、共立出版

電球が切れる確率1000時間に一回
バイクに乗って怪我をする確率1000時間に0.143回

「ソフトウェアテスト手法」高橋、湯本

欠陥率と満足度の関係

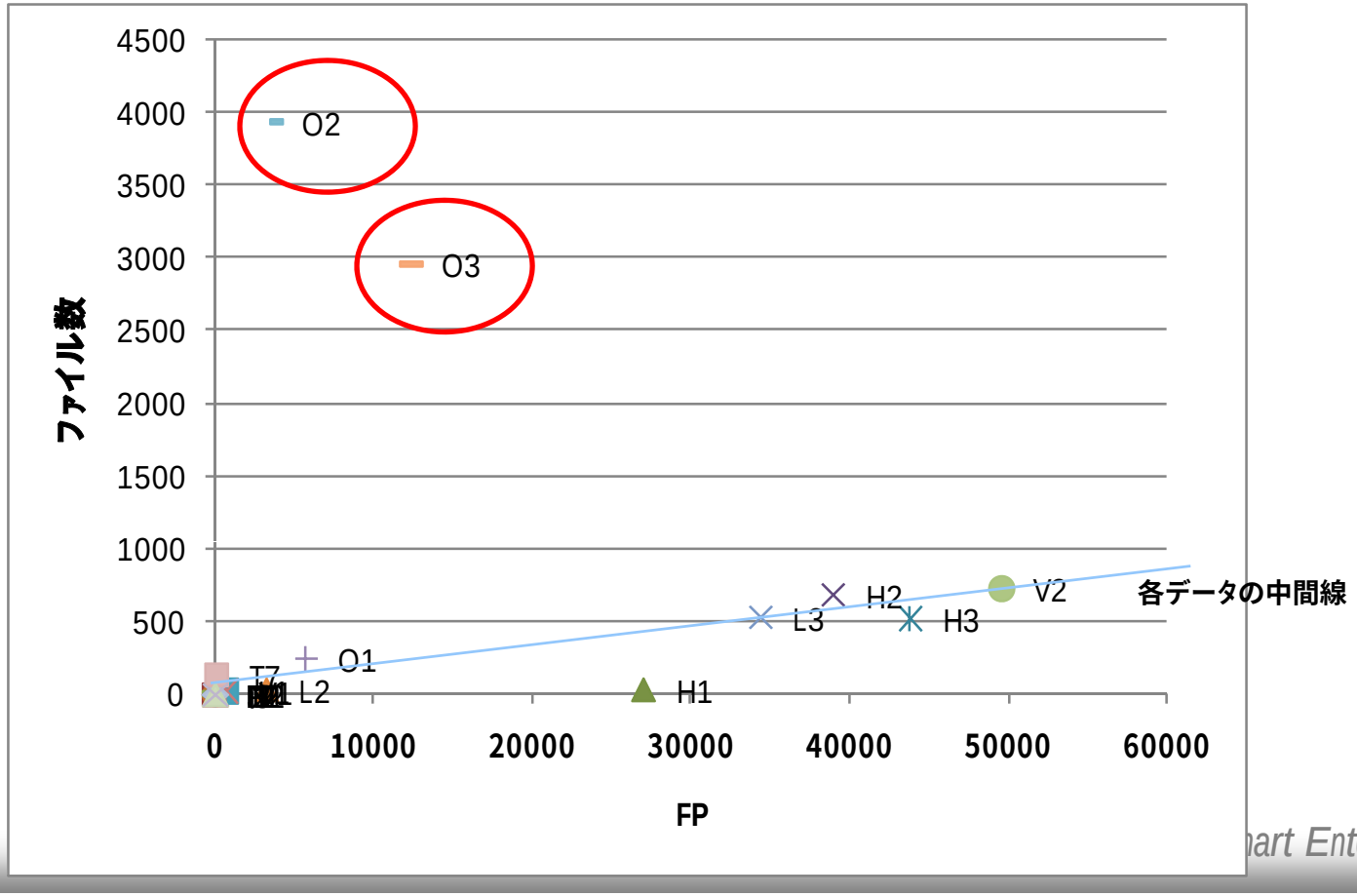
- 欠陥率のみでユーザの満足度が決まるわけではないが、やはり、Cランク以上のシステムが満足という顧客評価を得られる割合が高くなっている。

欠陥率		顧客満足度(品質)					
		満足	やや不満	不満	未回答	計	満足率
0	件数	10	4	1		15	66.7%
	平均	0.00	0	0		0	
0.25未満	件数	36	13	1		50	72.0%
	平均	0.10	0.13	0.05	0.15	0.11	
0.5未満	件数	21	6	1		28	75.0%
	平均	0.36	0.36	0.48	0.33	0.36	
1未満	件数	14	11	1		26	53.8%
	平均	0.68	0.68	0.54	0.65	0.68	
3未満	件数	13	7	2		22	59.1%
	平均	1.79	1.80	1.53	2.08	1.79	
3以上	件数	8	1			9	88.9%
	平均	5.63	4.35			5.49	
計	件数	102	42	6		150	68.0%
	平均	0.87	0.67	0.69	0.8	0.81	

「ソフトウェアメトリックス２００７」JUAS

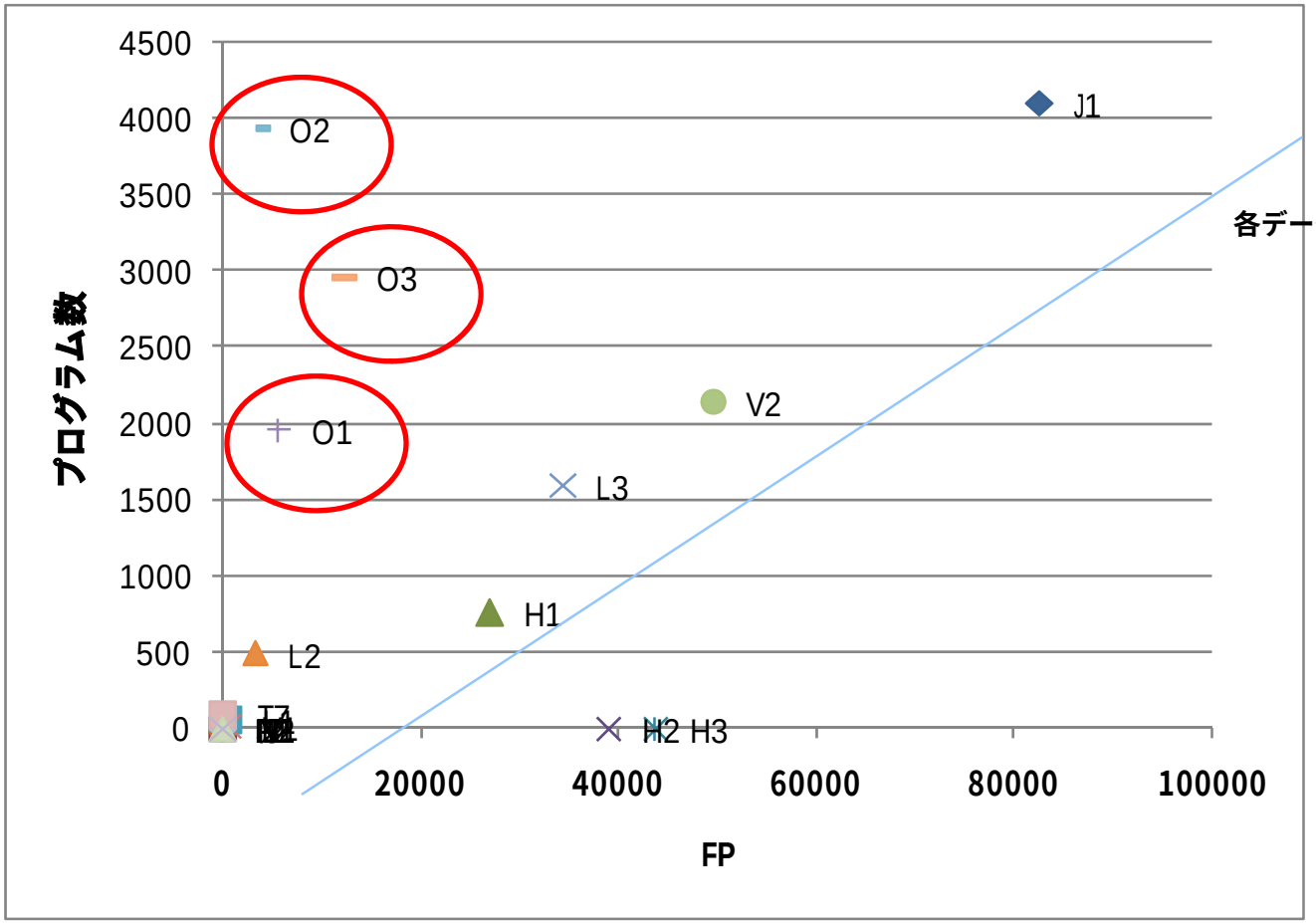
システムの構造は大丈夫か？

- 設計が終わると見える規模あたりのファイル数
- ファイル数が異常に多いシステムが発見できる。
- このようなＰＪは、作りに問題があるのではないか。
  - 保守性への影響も想定される。



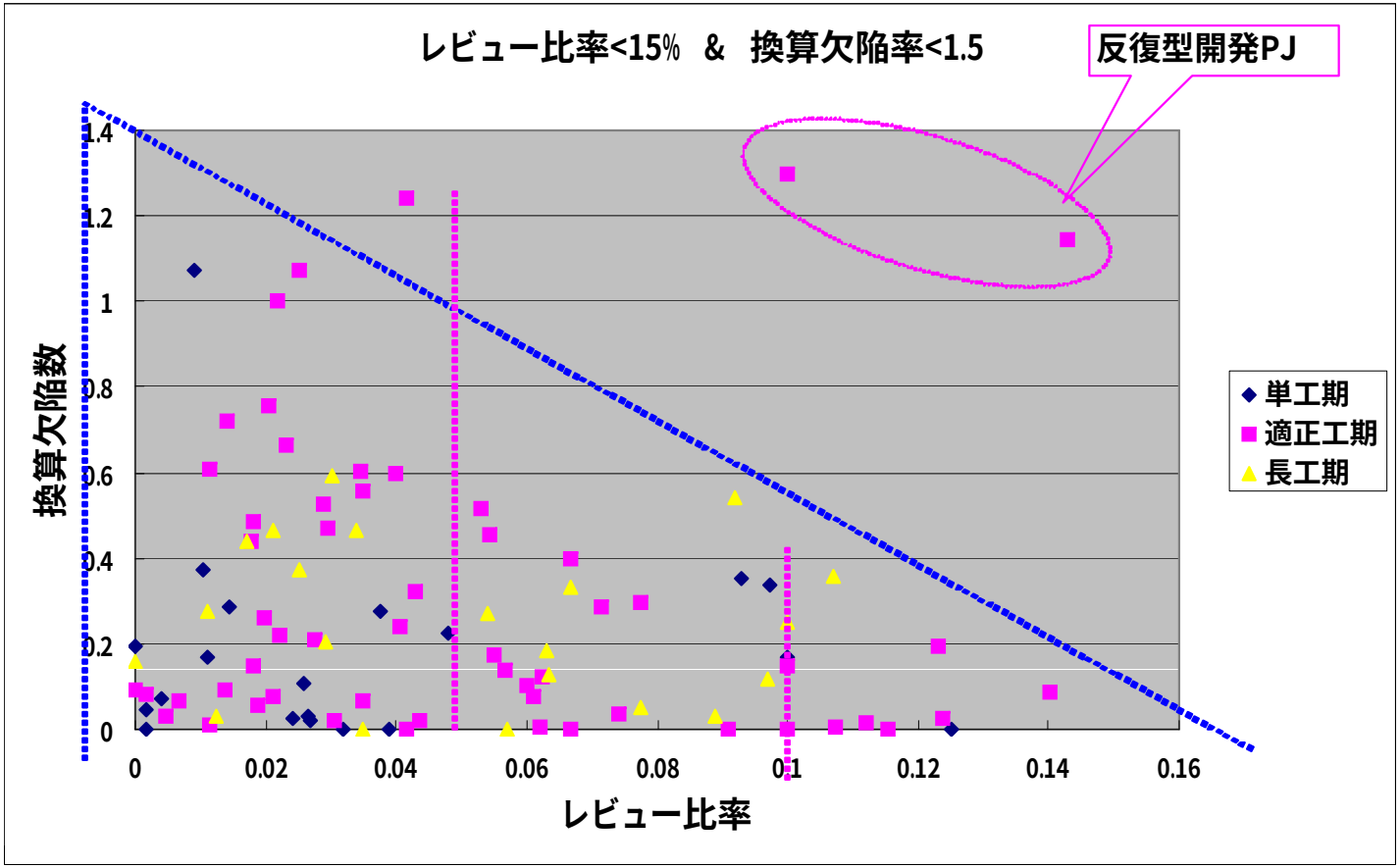
システムの構造は大丈夫か？

- ファイル同様に、プログラムの分割が細かいレベルで行われているものも発見できる。



レビュー

◆ 当然のことながら、レビューをすれば欠陥は少なくなる。



正しいレビューの実施

- レビューは役割分担やチェックリストなどを用意した上で計画的に実施することが重要である。ただし、現場に過剰な負荷をかけてはいけない。
  - アドホックレビュー
    - 作業途中で必要に応じて実施するレビュー
  - パスアラウンドレビュー
    - メール配布などによるレビュー依頼
  - ペアレビュー
    - 作成者とレビューアが成果を確認
  - ウォークスルー
    - 業務の流れに沿って作成者が説明を行うレビュー
  - チームレビュー
    - 計画された成果物レビューなど
  - インスペクション
    - モデレータ、記録者など役割を指定して行う査察的、厳格なレビュー

重点レビュー項目

- ISO9126にも規定されるソフトウェア品質特性に基づきレビューを行っていく
  - 機能性
    - 合目的性、正確性、接続性、整合性、セキュリティ
  - 信頼性
    - 成熟性、障害許容性、回復性
  - 使用性
    - 理解性、習得性、操作性
  - 効率性
    - 実行効率性、資源効率性
  - 保守性
    - 解析性、変更作業性、安定性、試験性
  - 移植性
    - 環境適応性、移植作業性、規格準拠性、置換性

基本設計のレビュー密度

- ソフトウェア開発データ白書によると以下の通りである。
  - 1000FPあたり
    - 平均77.1件
  - KLOCあたり
    - 平均3.9件
  - ページあたり
    - 平均0.393件

レビューにおけるカバレッジ

- CMMレベル5を持つInfosysでは以下の基準でレビューを行っている。

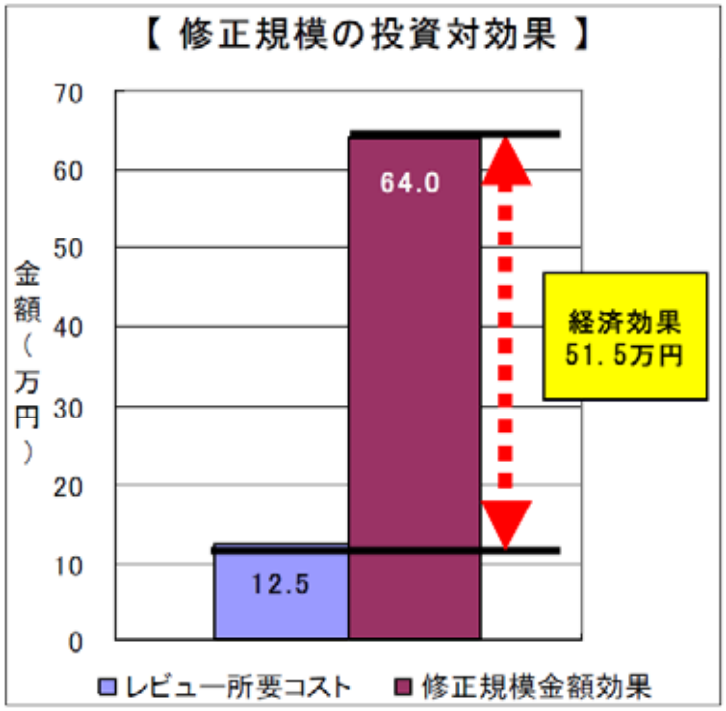
Infosysのレビュー能力ベースライン

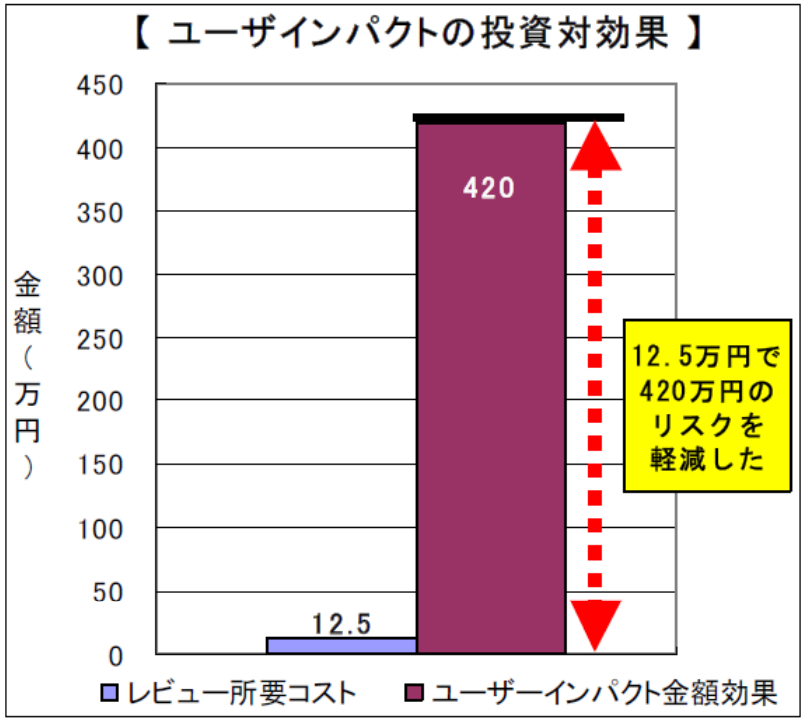
レビュー項目	準備のカバレッジ率	グループレビューのカバレッジ率	表面的/軽微な欠陥の欠陥密度	致命的/重大な欠陥密度
要件		5-7page/時	0.5-1.5欠陥/page	0.1-0.3欠陥/page
ハイレベル設計		4-5page/時（または200-250仕様文/時）	0.5-1.5欠陥/page	0.1-0.3欠陥/page
詳細設計		3-4page/時（または70-100仕様文/時）	0.5-1.5欠陥/page	0.2-0.6欠陥/page
コード	160-200LOC/時	4-6page/時	0.01-0.06欠陥／LOC	0.01-0.06欠陥/page
統合テスト計画		5-7page/時	0.5-1.5欠陥/page	0.1-0.3欠陥/page
統合テストケース		3-4page/時		
システムテスト計画		5-7page/時	0.5-1.5欠陥/page	0.1-0.3欠陥/page
システムテストケース		3-4page/時		
プロジェクト管理計画および構成管理計画	4-6page/時	2-4page/時	0.6-1.8欠陥/page	0.1-0.3欠陥/page

ソフトウェア開発のためのプロジェクトマネジメント入門、ソフトバンクパブリッシング

レビューの効果

- レビューを行いバグを防ぐことで、未然に無駄な支出を防止することが可能である。





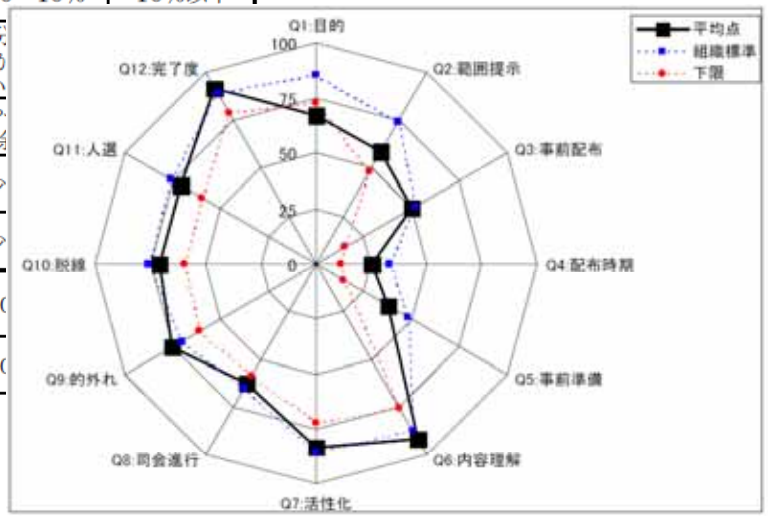
日本科学技術連盟第23年度(2007年度) 分科会成果報告 第1分科会「ソフトウェアプロセス評価･改善」 グループA (論文)レビューの質と価値の定量化の提案

レビューの質を高める取り組み

- レビューが正しく行われていたかは、以下のようにチェックリストで管理できる

表 2. レビューの質チェック票

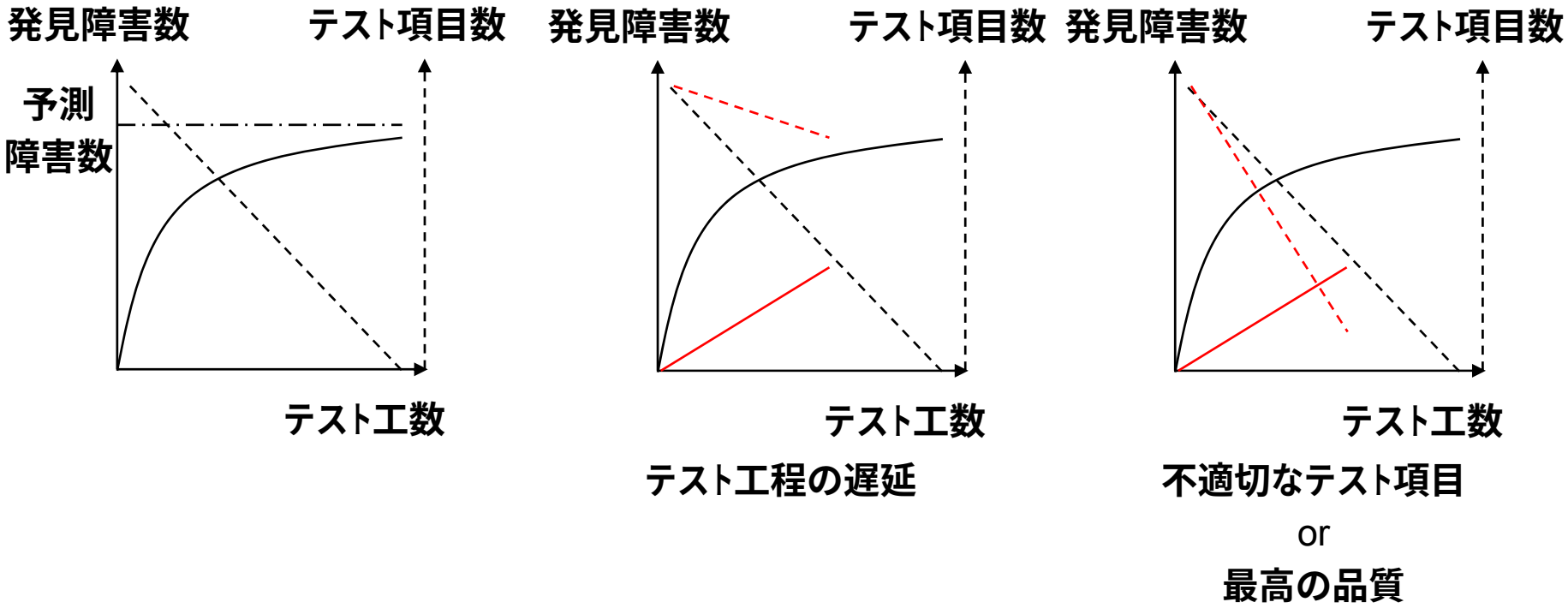
1	レビューの目的／狙いが明確だったか	良い	まあ良い	やや改善の余地有り	改善の余地有り
2	レビュー物件の対象範囲は適切に示されていたか（前回との差分などが明確だったか）	良い	まあ良い	やや改善の余地有り	改善の余地有り
3	必要な資料は全て事前配布／提示されていたか	80%以上	80～50%	50～10%	10%以下
4	資料はいつ頃事前配布されていたか（資料に目を通す期間は確保されていたか）	十分余裕が有った	期間が有った	期間が十分では無かった	事前配布無しまたは直前
5	レビュー会議前に資料に目を通したか	80%以上	80～50%	50～10%	10%以下
6	レビュー物件の内容は十分伝わったか	80%以上	80～50%	50～10%	10%以下
7	全員が満遍なく発言していたか	全員発言していた	半分以上は発言していた	半分	
8	レビューリーダのファシリテートは適切だったか	良い	まあ良い	や	
9	的外れな指摘/コメントが少なかったか	全く無し	殆ど無し	少	
10	脱線することが少なかったか	全く無し	殆ど無し	少	
11	参加者で問題領域をカバーできていたか（レビュー参加者の人選は適切だったか）	100%	80%以上	80	
12	予定のレビュー対象範囲が完了したか	100%	80%以上	80	



日本科学技術連盟第23年度(2007年度) 分科会成果報告 第1分科会「ソフトウェアプロセス評価･改善」 グループA (論文)レビューの質と価値の定量化の提案

テストには科学的データがあるが活用されていないことが多い

- テストではバグがあることは証明できるが、バグが無いことを証明することはできない。
  - ただし、コストとバランスの取れたレベルを実現することはできる
- FPあたりのテスト項目数



テスト項目数と項目密度

- 規模あたりのテスト項目数と項目比率は以下のとおりである。
- 特に、項目は多ければよいというものではなく、比率よく配分しなければならない。

	検査項目総数
制御系	30-50件/KLOC
業務系	20-25件/KLOC

	項目比率
基本/正常	50-60%
異常/障害	10-20%
限界/境界	5-10%
周囲条件/インタフェース	20%

第48回SEA-SPIN Meeting　ソフトウェアテストと品質

工程	ケース数
単体テスト	7.5/FP (5-15/FP)
結合テスト	2.5/FP (0.5-8/FP)
総合テスト(1)	1.2/FP (0.6-10/FP)
総合テスト(2)	ー

2007JUAS　QCDワーキング

参考10/KLOC=1/FP

FPあたりのケース密度

- まだ、データ数が少なく、今後のデータの充実が望まれる。

		～10人月	～50人月	～100人月	～500人月	500人月超	記入なし	総計
件数		2	5	5	6	3	3	24
ベンダ内テスト	平均（CASE/FP）	0.7	2.8	3.0	12.7	1.8	0.8	4.8
	最大（CASE/FP）	1.2	5.9	9.9	31.4	3.2	1.4	31.4
	最小（CASE/FP）	0.3	0.1	0.1	2.8	0.5	0.1	0.1
顧客側テスト	平均（CASE/FP）	0.7	0.1	0.6	2.2	0.2	0.2	0.8
	最大（CASE/FP）	1.3	0.3	1.3	6.9	0.4	0.4	6.9
	最小（CASE/FP）	0.0	0.0	0.0	0.0	0.1	0.0	0.0

言語別、開発種別ごとのケース密度

- 結合テスト、総合テストなどのケース密度が、言語別、開発種別々に提供されている

ソフトウェア開発データ白書２００８　P240

主要開発言語別SLOCあたりの総合テストケース数の基本統計量（改良開発）

（ケース数/Kstep）	N	最小	中央	最大	平均
全体	124	0.019	10.188	796.881	45.464
COBOL	43	0.019	6.392	82.875	13.589
C言語	32	1.250	17.114	796.881	84.459
VB	19	0.723	18.784	215.700	47.879
Java	30	0.115	7.896	604.000	48.027

上記総合テストには、ユーザ受け入れ試験を含む

主要開発言語別SLOCあたりの総合テスト欠陥数の基本統計量（改良開発）

ケース数/Kstep	N	最小	中央	最大	平均
全体	122	0.000	0.219	13.333	1.068
COBOL	46	0.000	0.097	12.222	0.905
C言語	30	0.000	0.338	13.333	1.339
VB	18	0.036	0.576	4.917	1.160
Java	28	0.000	0.123	10.000	0.988

テスト計画書のメトリックスチェックの例

結合テスト計画書の確認項目

開発区別	新規	改良
言語	Java	
規模（Kstep）	1200	

目標設定値	下限	上限	実績	IPA参考
テストケース密度（ケース数/Kstep）			3.36	38.965
テストケース数			4032	
不具合検出率（不具合数／Kstep）			0.051	1.401
不具合数			61.2	

テストケース項目比率

	比率	参考
基本/正常		50-60%
異常/障害		10-20%
限界/境界		5-10%
周囲条件/インタフェース		20%

第48回SEA-SPINソフトウェアテストと品質

評価

移植であるためテストケース密度は低い。この時点で品質は非常によいが、ケースが少ないため不具合の検証に漏れがある可能性がある。今後もモニターしていく必要がある。

改良
java
中央値

ソフトウェア開発データ白書2008　P240

主要開発言語別SLOCあたりの結合テストケース数の基本統計量（改良開発）

（ケース数/Kstep）	N	最小	中央	最大	平均
全体	104	0.321	24.889	1964.000	84.837
COBOL	32	0.321	15.285	82.438	22.695
C言語	25	3.632	43.894	674.000	111.403
VB	15	1.445	32.061	631.064	115.358
Java	32	3.774	38.956	1964.000	111.955

主要開発言語別SLOCあたりの結合テスト欠陥数の基本統計量（改良開発）

ケース数/Kstep	N	最小	中央	最大	平均
全体	102	0.000	1.126	42.357	2.770
COBOL	34	0.000	0.457	5.226	1.181
C言語	25	0.067	1.333	42.358	5.230
VB	15	0.181	0.896	10.946	2.599
Java	28	0.000	1.401	24.000	2.596

システムテスト計画書の確認項目

開発区別	新規	改良
言語	Java	
規模（Kstep）	1200	

目標設定値	下限	上限	実績	IPA参考	JUAS参考
テストケース密度（ケース数/Kstep）	2			7.896	22.7374
テストケース数	2400				
不具合検出率（不具合数／Kstep）	0.02	0.08		0.123	
不具合数	24	96			

評価

移植であるためテストケース密度は低いが、品質目標は適正レベルといえる。

改良	500人以上
java	平均
中央値	言語コンバージョン5/7.63

ソフトウェア開発データ白書2008　P240

主要開発言語別SLOCあたりの総合テストケース数の基本統計量（改良開発）

（ケース数/Kstep）	N	最小	中央	最大	平均
全体	124	0.019	10.188	796.881	45.464
COBOL	43	0.019	6.392	82.875	13.589
C言語	32	1.250	17.114	796.881	84.459
VB	19	0.723	18.784	215.700	47.879
Java	30	0.115	7.896	604.000	48.027

上記総合テストには、ユーザ受け入れ試験を含む

主要開発言語別SLOCあたりの総合テスト欠陥数の基本統計量（改良開発）

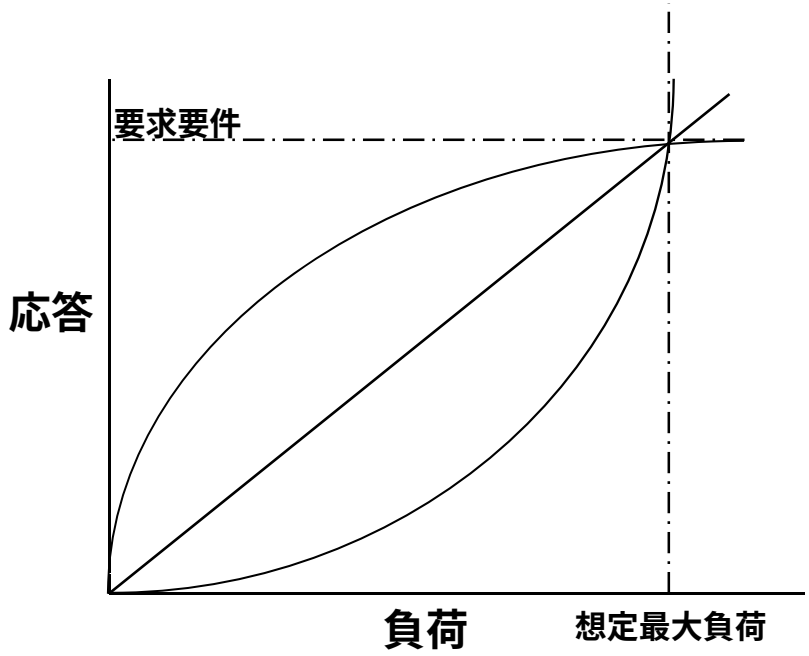
ケース数/Kstep	N	最小	中央	最大	平均
全体	122	0.000	0.219	13.333	1.068
COBOL	46	0.000	0.097	12.222	0.905
C言語	30	0.000	0.338	13.333	1.339
VB	18	0.036	0.576	4.917	1.160
Java	28	0.000	0.123	10.000	0.988

ソフトウェアメトリックス調査2008　P142

		～10人月	～50人月	～100人月	～500人月	500人月超	記入なし	総計
件数	件数	2	37	15	26	5	4	89
ベンダ内テスト（ケース数/Kstep）	平均	38.8	92.2	23.5	88	14.9	120.5	75.1
	最大	41.1	963.4	125.1	916	30.5	456.1	963.4
	最小	36.5	0	0.1	0	3.7	0.3	0
顧客側テスト（ケース数/Kstep）	平均	5.2	37.9	26.9	13.6	1.4	2.6	24.5
	最大	10.4	588.5	347.4	80	4.1	5.2	588.5
	最小	0.0	0.0	0.1	0.0	0.1	0.1	0.0

パフォーマンステスト

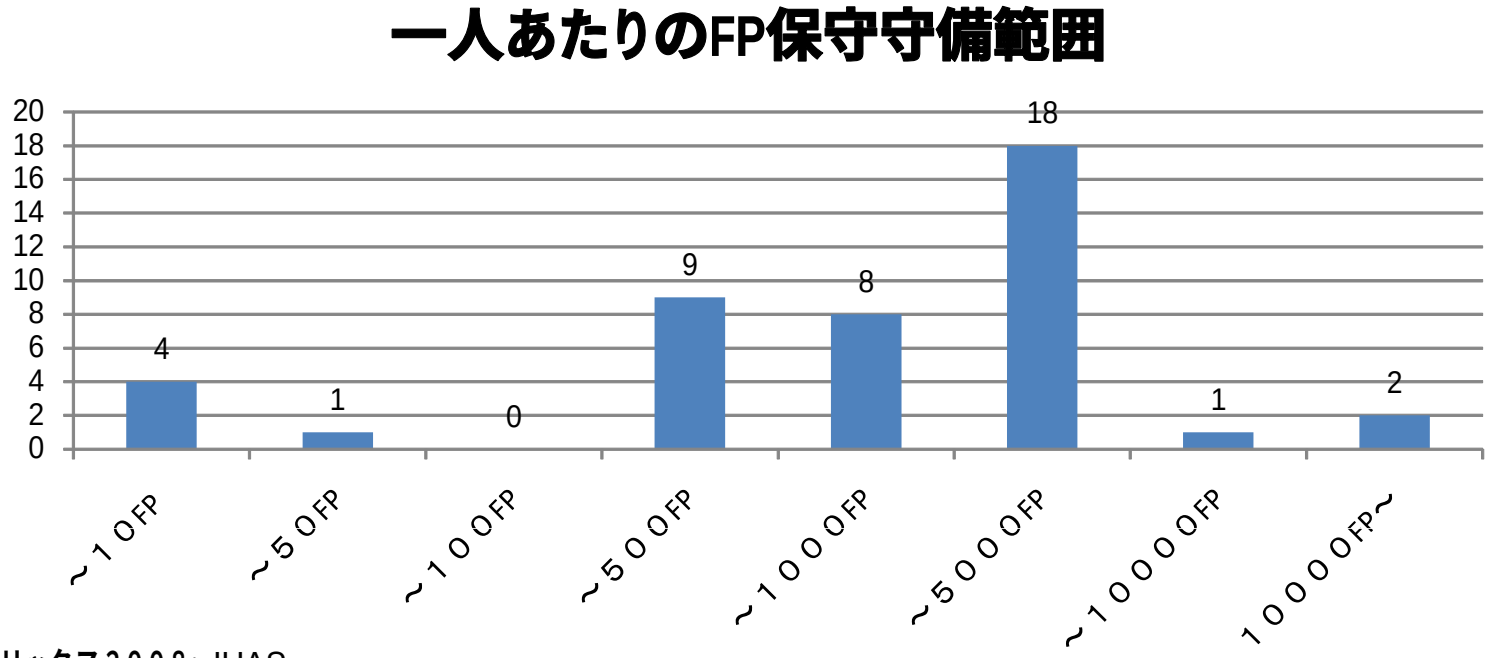
- システムにおける性能は非常に重要な要素である。
  - 最大負荷時に要件を満たすのか
  - 想定負荷までの応答の状況はどうなっているのかなど検証していく必要がある
  - また、同じシステム上で並行稼動しているプロセスがある場合は、その条件もテスト時に確認されているか検証する必要がある。



立ち上げの段階で適正な保守計画を行う

- データは少ないが数百から数千FPに一人の割合で要員を配置している。

平均　　３６５２．４FP



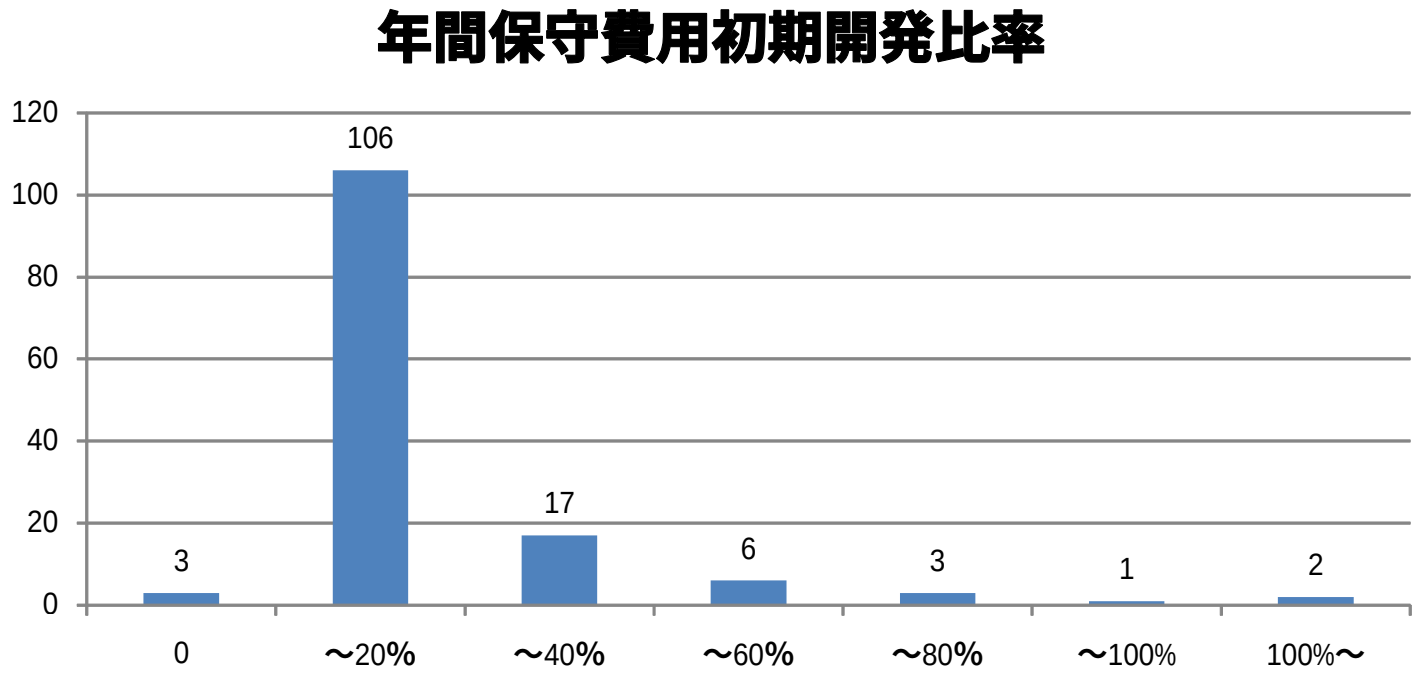
「ソフトウェアメトリックス２００８」JUAS

ライフサイクルコストとして適正か検証をしていく

初期費用に対する保守費用

◆ 年間保守費用は、初期開発費用の１６．６%程度である。

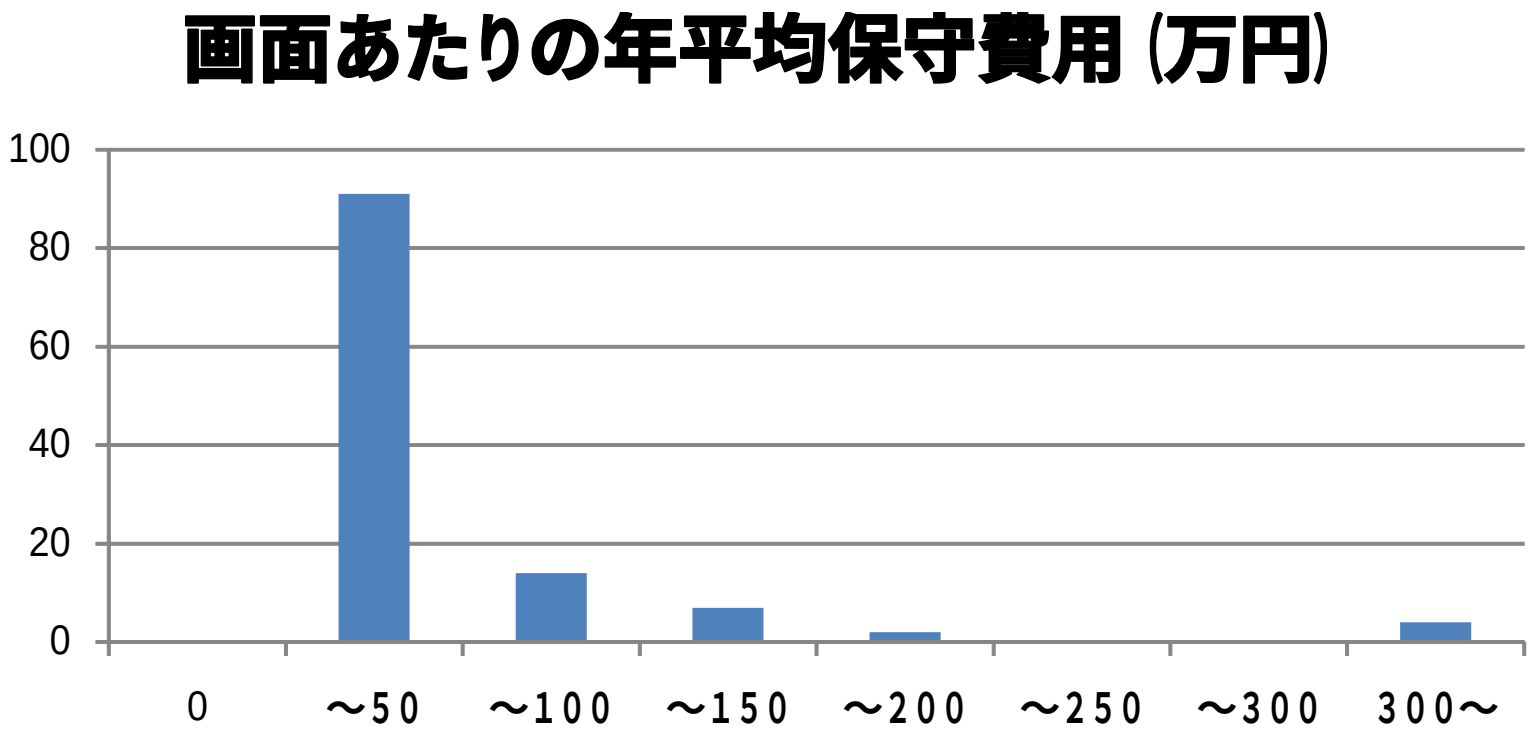
平均　　16.60%



「ソフトウェアメトリックス２００８」JUAS

画面あたりの保守費用

平均　102.1万円



これくらいの予算は確保しておくと良い

「ソフトウェアメトリックス２００８」JUAS

カットオーバー時の品質の保守への影響

- 保守作業との関係はあるが、欠陥率のデータはまだ十分ではない。

平均保守作業期間

	半日以下	1日以内	3日以内	1週間以内	1月以内	1月以上
CO時の品質はよい	31.7%	19.4%	20.0%	12.1%	11.1%	5.7%
CO時の品質はよくない	29.9%	17.8%	14.3%	14.1%	14.3%	9.5%

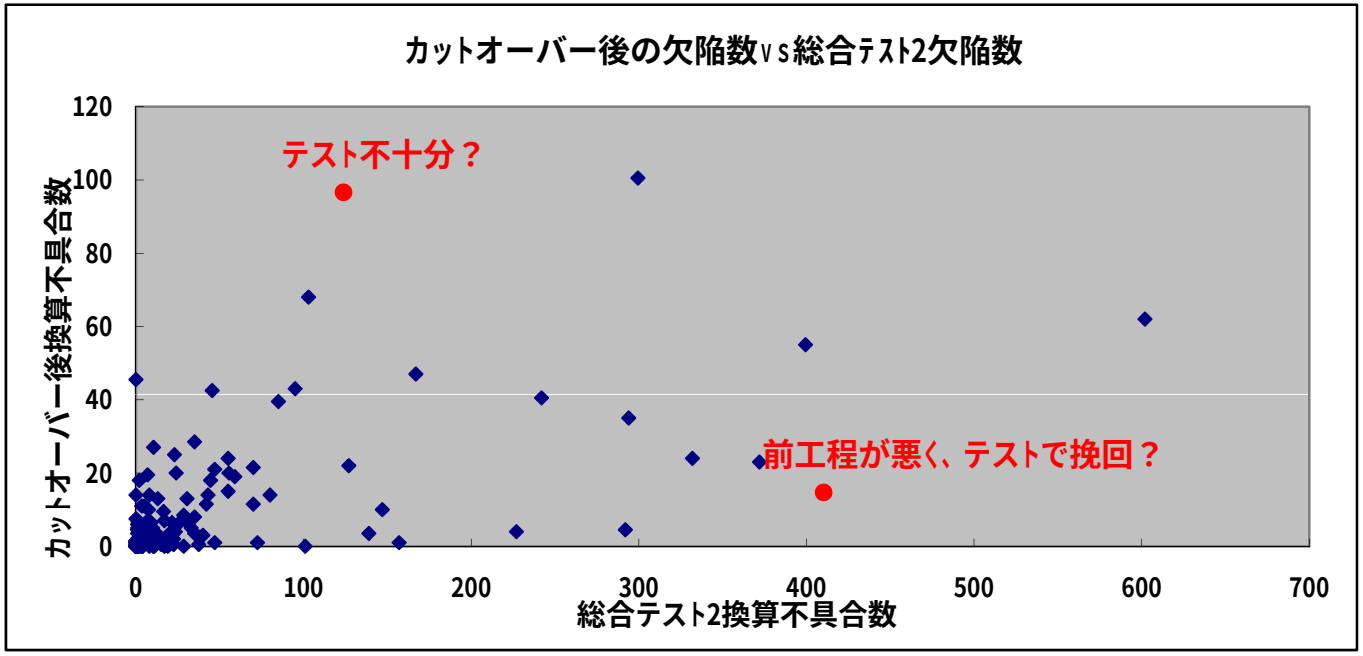
CO時品質	初年度保守欠陥率	2年目以降保守欠陥率	受入確認即時合格率
非常によい	8.5%	6.3%	50.3%
よい	20.1%	11.7%	64.2%
普通	19.7%	13.4%	61.2%
やや悪い	22.1%	7.1%	81.2%
非常に悪い	9.0%	5.0%	95.0%

保守欠陥率＝欠陥発生件数／保守作業実施件数

「ソフトウェアメトリックス２００８」JUAS

サービス開始後の品質は適正レベルか

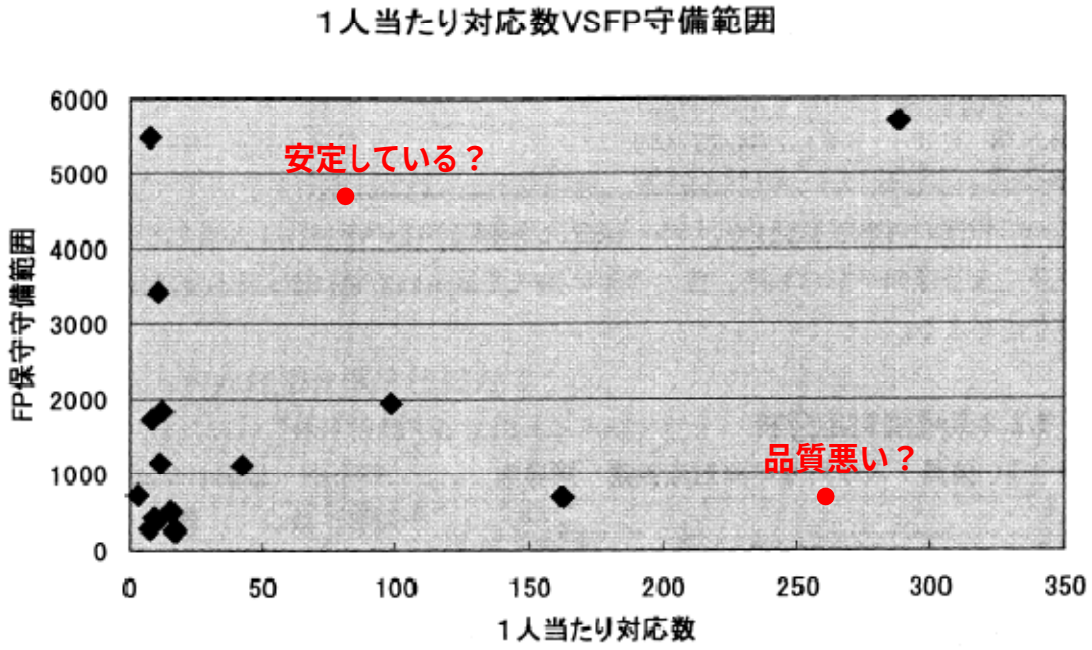
- またテスト中の欠陥数から稼働後の欠陥数を推定することができ、これと比較することにより品質の安定度を判断することが可能になる。テスト時の発生不具合数から、導入初期の不具合数を想定し、それを念頭に置いた運用を行い、異常な不具合の多さなどが見られるときは、テストデータの確認など必要な措置を講じる必要がある。



「ソフトウェアメトリックス２００８」JUAS

保守による品質の検証

- 規模あたりの欠陥数、フォロー時の欠陥の多さで品質の概要はつかめるが、長期にわたっても検証をすることが重要である。保守要員がどのくらいの範囲を見ているのか、その中でどのくらいの対応をしているかという点からも対策が考えられる。
  - FP守備範囲が広く、一人当たり対応数が多いものは、保守要員の人で不足による悪循環などが想定される。



「ソフトウェアメトリックス２００７」JUAS

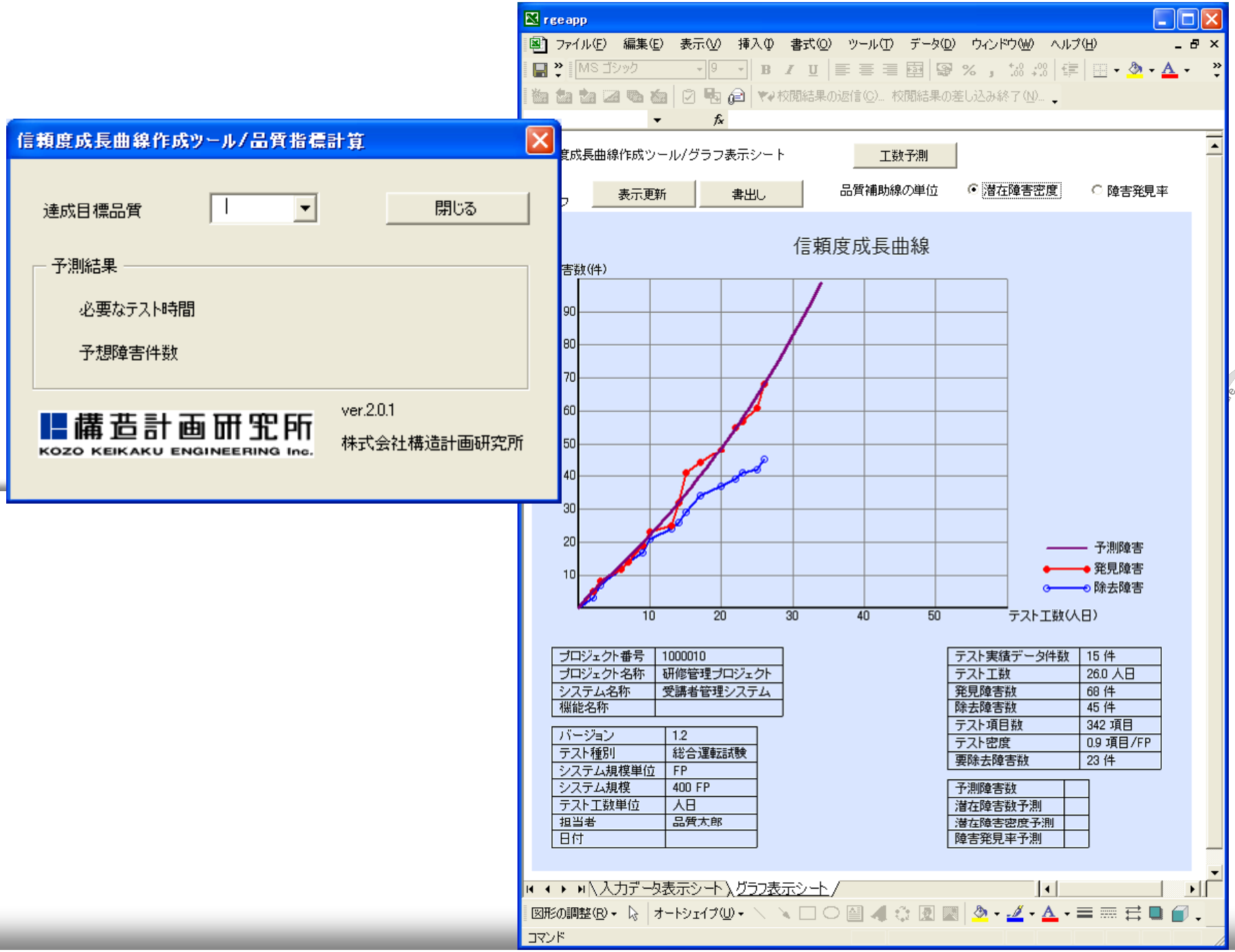
稼働率の目標

	レベル1	レベル2	レベル3	レベル4	レベル5
稼働率	98%以下	99%	99.9%	99.99%	99.999%以上
バックアップ機	なし	あり (部分的)	あり (2／N＋1台)	あり (Hot stand by)	あり (Hot stand by)
サービス停止時間 (　)時間／年	172時間	86時間	8.6時間	50分	5分
到着時間	1-6時間 (昼) 12時間 (夜間)	1-6時間	1-3時間 (昼) 6時間 (夜間)	常駐 ケースによっては2時間	常駐
修復時間 ・故障修復 ・再立ち上げ	6時間-12時間 10分-1時間	6時間-12時間 10分-1時間	3時間-6時間 10分-1時間	3時間-6時間 0分-10分	3時間-6時間 即時
費用 ・構築費用 ・運用費用	1.0倍 1.0倍	1.2～1.8倍 1.1～1.3倍 (マニュアル)	1.2～3倍 1.3～2.0倍	1.5～4倍 2.0～3倍 (保守も)	4～6倍 3～4倍
システム構成 (例) 必要な機能		NAS	SAN NAS クラスタリング ﾛｰﾄﾞﾊﾞﾗﾝｼﾝｸﾞ	SAN クラスタリング ﾛｰﾄﾞﾊﾞﾗﾝｼﾝｸﾞ 三重化	SAN クラスタリング ﾛｰﾄﾞﾊﾞﾗﾝｼﾝｸﾞ 三重化、四重化
ペナルティ			対象	対象	対象

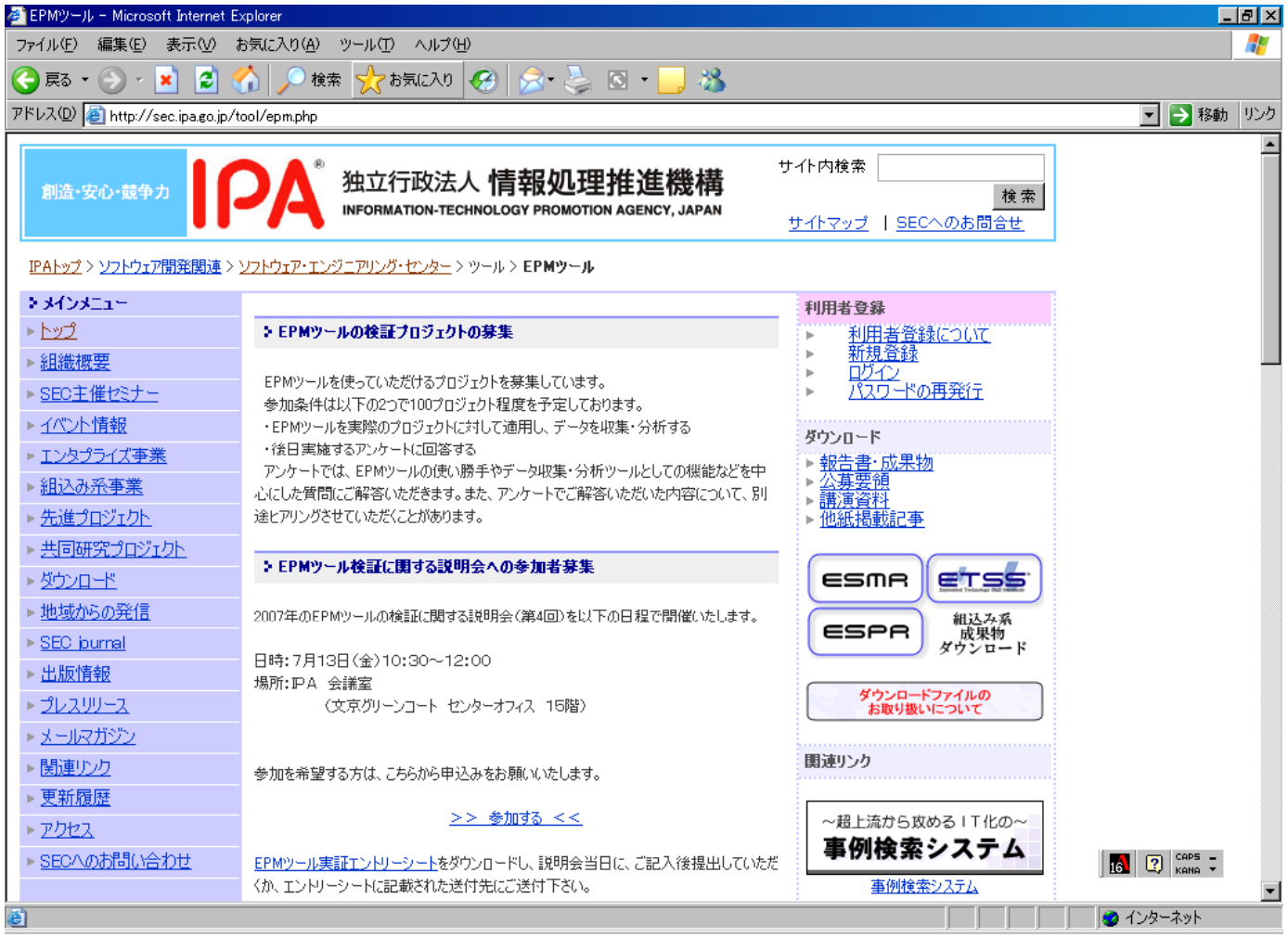
ツール例

	項目名	入力値	単位	必須	状態	条件
(a)	プロジェクト番号	1000010		-	ok	
(b)	プロジェクト名称	研修管理プロジェクト		-	ok	
(c)	システム名称	受講者管理システム		-	ok	
(d)	機能名称			-	ne	
(e)	バージョン	1.2		-	ok	
(f)	テスト種別	総合運転試験		-	ok	
(g)	システム規模単位	FP		○	ok	◇””
(h)	システム規模	400 FP		○	ok	>0
(i)	テスト工数単位	人日		○	ok	◇””
(j)	担当者	品質太郎		-	ok	
(k)	日付			-	ne	

(l)	(m)	(n)	(o)	(p)	(q)	(r)	(s)	(t)	(u)	(v)
No.	識別コメント	テスト工数 >0 (人日)	累計	発見障害数 >=0 (件)	累計	除去障害数 >=0 (件)	累計	テスト項目数 >=0 (件)	累計	状態
1		2.0	2.0	5	5	3	3	34	34	ok
2		1.0	3.0	3	8	4	7	35	69	ok
3		3.0	6.0	4	12	5	12	42	111	ok
4		1.0	7.0	2	14	2	14	33	144	ok
5		2.0	9.0	5	19	3	17	12	156	ok
6		1.0	10.0	4	23	4	21	15	171	ok
7		3.0	13.0	2	25	3	24	25	196	ok
8		1.0	14.0	7	32	2	26	31	227	ok
9		1.0	15.0	9	41	3	29	21	248	ok
10		2.0	17.0	3	44	5	34	12	260	ok
11	障害除去に注力が必要	3.0	20.0	4	48	3	37	34	294	ok
12		2.0	22.0	7	55	2	39	12	306	ok
13		1.0	23.0	2	57	2	41	12	318	ok
14		2.0	25.0	4	61	1	42	9	327	ok
15		1.0	26.0	7	68	3	45	15	342	ok


信頼度成長曲線作成ツール/品質指標計算
達成目標品質
閉じる
予測結果
必要なテスト時間
予想障害件数
構造計画研究所
KOZO KEIKAKU ENGINEERING Inc.
ver.2.0.1
株式会社構造計画研究所
rgeapp
ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) ヘルプ(H)
工数予測
表示更新
書出し
品質補助線の単位
潜在障害密度
障害発見率
信頼度成長曲線
害数(件)
テスト工数(人日)
予測障害
発見障害
除去障害
プロジェクト番号 1000010
プロジェクト名称 研修管理プロジェクト
システム名称 受講者管理システム
機能名称
バージョン 1.2
テスト種別 総合運転試験
システム規模単位 FP
システム規模 400 FP
テスト工数単位 人日
担当者 品質太郎
日付
テスト実績データ件数 15 件
テスト工数 26.0 人日
発見障害数 68 件
除去障害数 45 件
テスト項目数 342 項目
テスト密度 0.9 項目/FP
要除去障害数 23 件
予測障害数
潜在障害数予測
潜在障害密度予測
障害発見率予測
入力データ表示シート
グラフ表示シート
図形の調整(R)
オートシェイプ(U)
コマンド

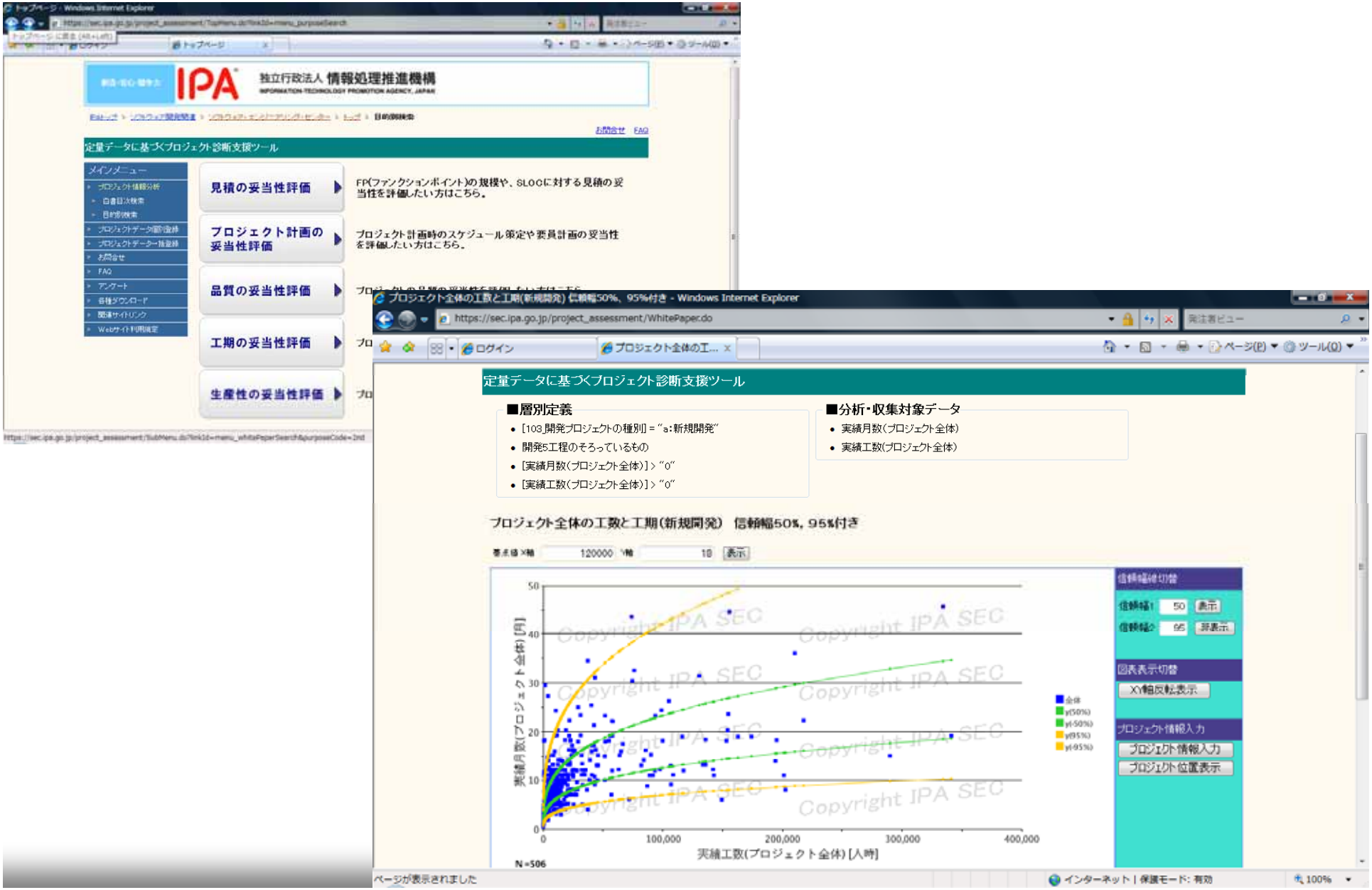
IPAが提供する無料ツールEPM



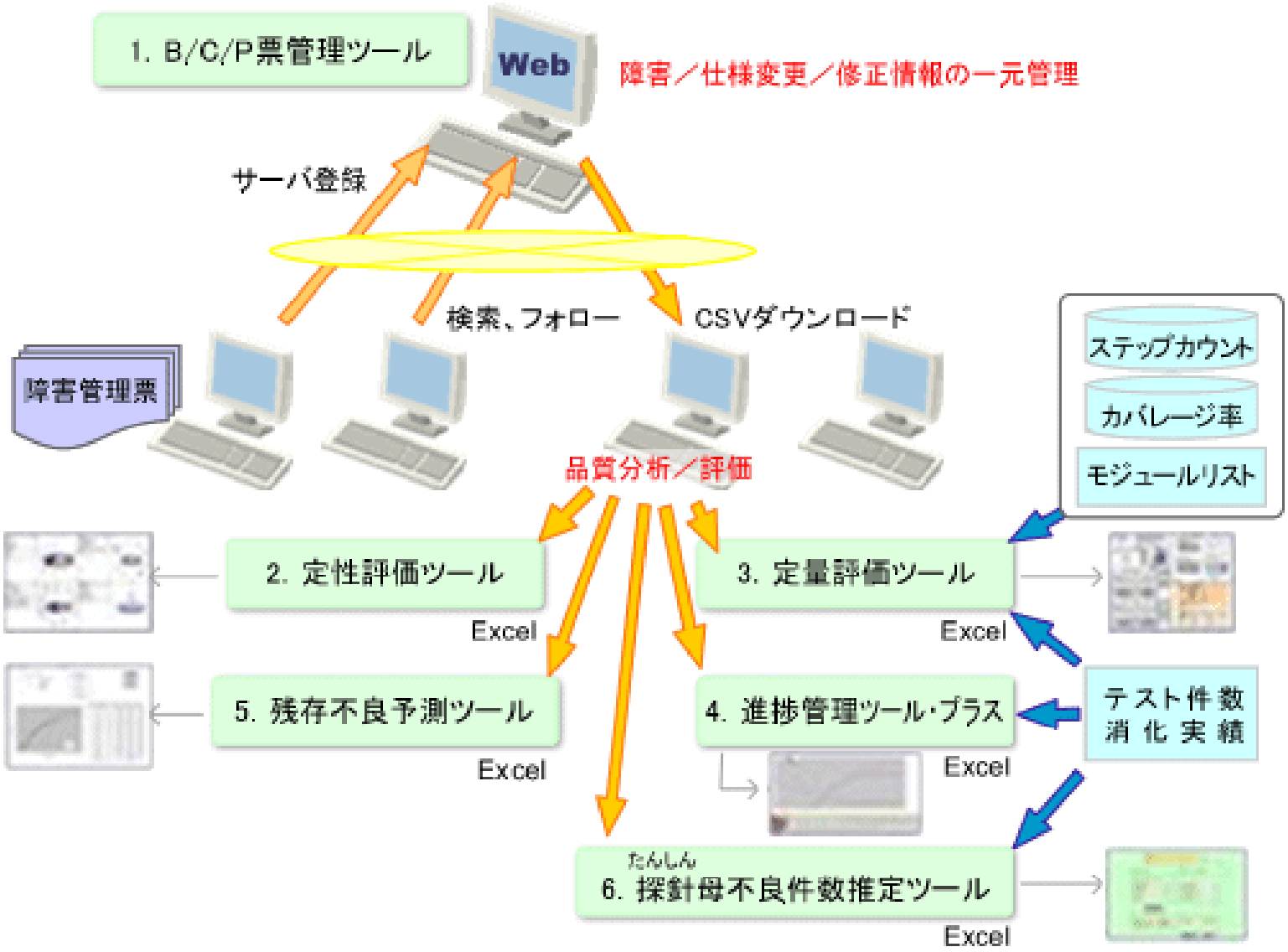
IPAが提供する無料ツールEPM

IPAが提供する無料ツールEPM

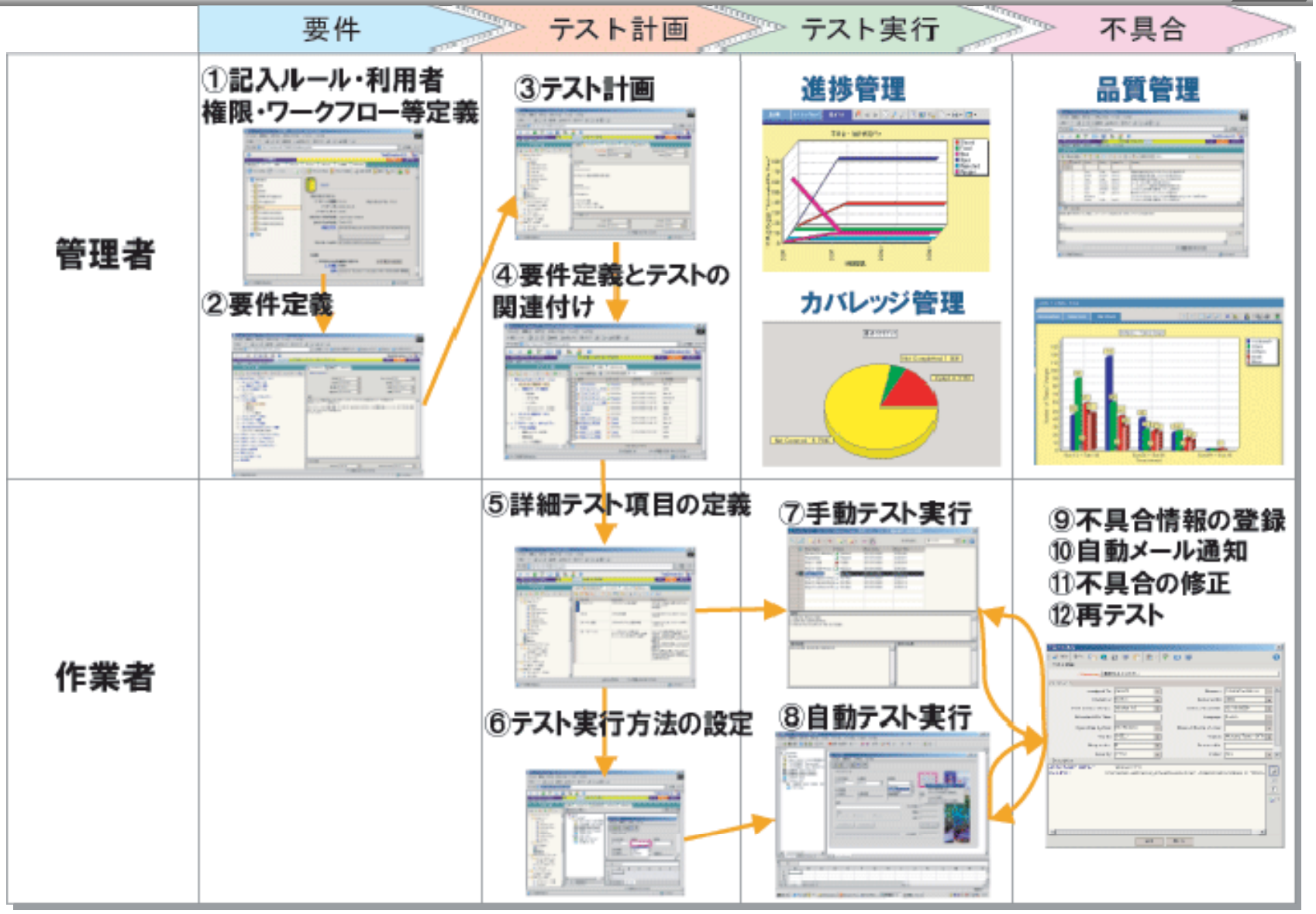
プロジェクト診断支援ツール



日立製作所ＱＥーＥＸＰＥＲＴ



日立システムの総合品質管理



カットオーバー時の品質の保守への影響

- 保守作業との関係はあるが、欠陥率のデータはまだ十分ではない。

平均保守作業期間

	半日以下	1日以内	3日以内	1週間以内	1月以内	1月以上
CO時の品質はよい	31.7%	19.4%	20.0%	12.1%	11.1%	5.7%
CO時の品質はよくない	29.9%	17.8%	14.3%	14.1%	14.3%	9.5%

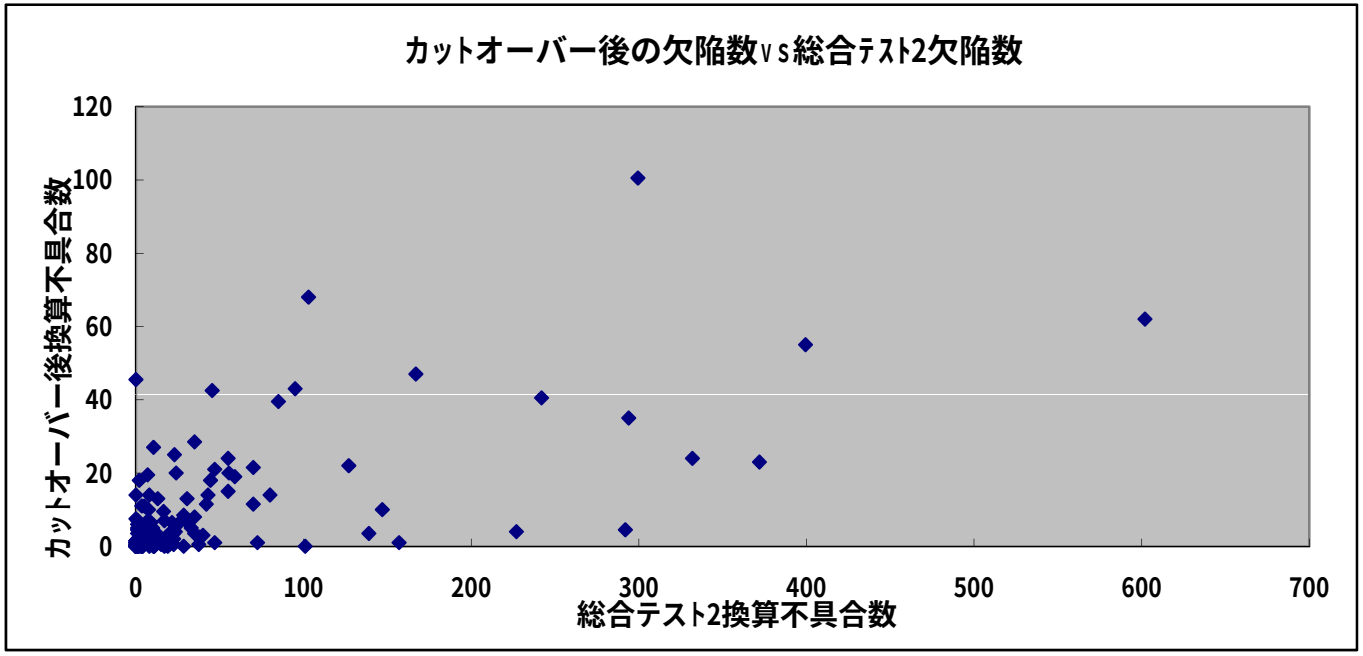
CO時品質	初年度保守欠陥率	2年目以降保守欠陥率	受入確認即時合格率
非常によい	8.5%	6.3%	50.3%
よい	20.1%	11.7%	64.2%
普通	19.7%	13.4%	61.2%
やや悪い	22.1%	7.1%	81.2%
非常に悪い	9.0%	5.0%	95.0%

保守欠陥率＝欠陥発生件数／保守作業実施件数

「ソフトウェアメトリックス２００８」JUAS

サービス開始後の品質は適正レベルか

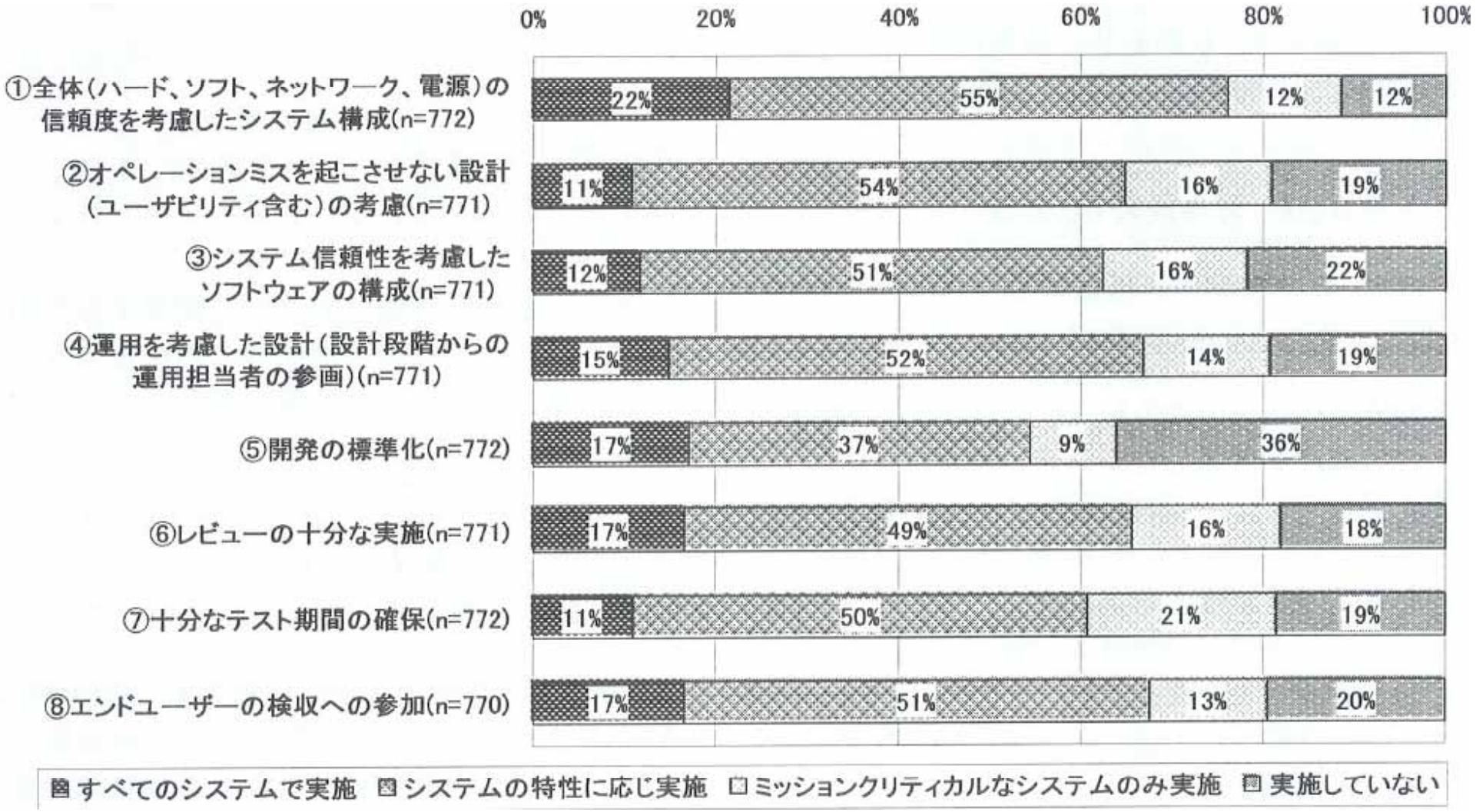
- またテスト中の欠陥数から稼働後の欠陥数を推定することができ、これと比較することにより品質の安定度を判断することが可能になる。テスト時の発生不具合数から、導入初期の不具合数を想定し、それを念頭に置いた運用を行い、異常な不具合の多さなどが見られるときは、テストデータの確認など必要な措置を講じる必要がある。



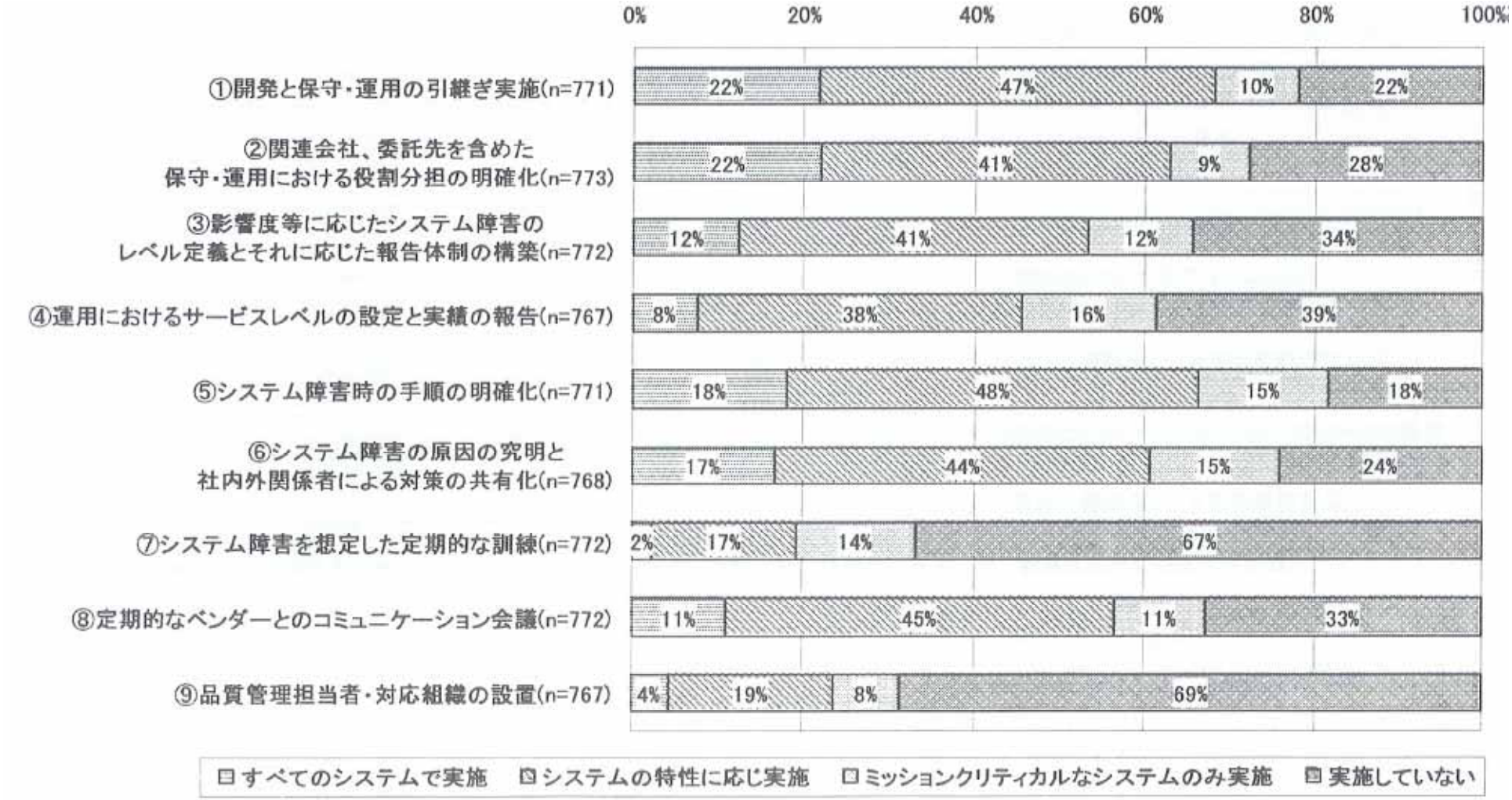
「ソフトウェアメトリックス２００８」JUAS

信頼性向上

信頼性向上のためのシステム設計・開発段階での施策



信頼性向上のためのシステム保守・運用段階での施策

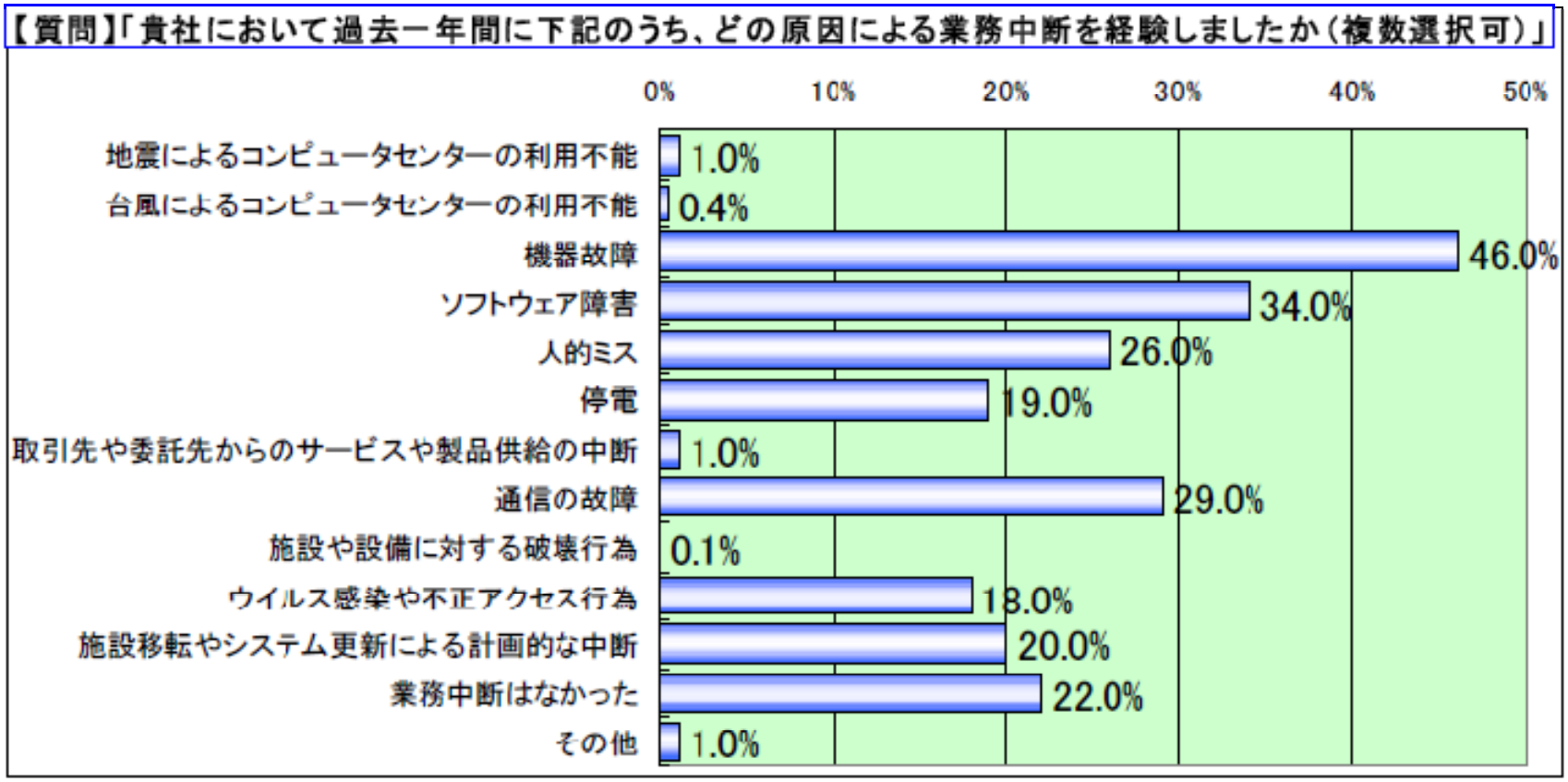


信頼性にかかわる事象

＜.結果事象の分類定義＞ （東京証券取引所「危機管理への取り組み」より）

①局所被害	テロ（予告、破壊行為）等により当取引所は被害をうけているものの、外部関係機関には特段の影響がない場合
②広域災害	大規模地震、風水害等により、当取引所及び外部機関がともに被害を受けている場合
③システム障害	システムのハード障害、アプリケーション障害、通信回線障害等により、当取引所の情報システムが利用不能となった場合

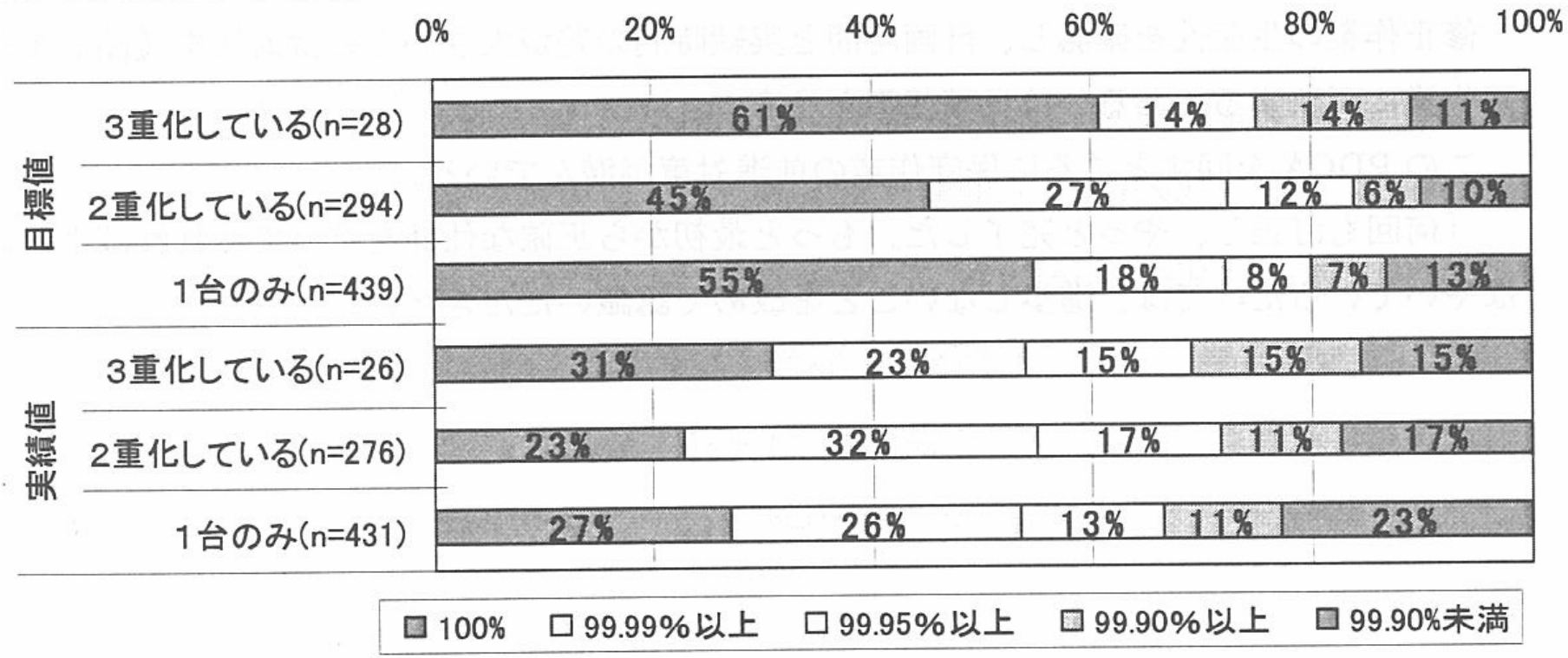
信頼性にかかわる経験



事業継続計画策定ガイドライン

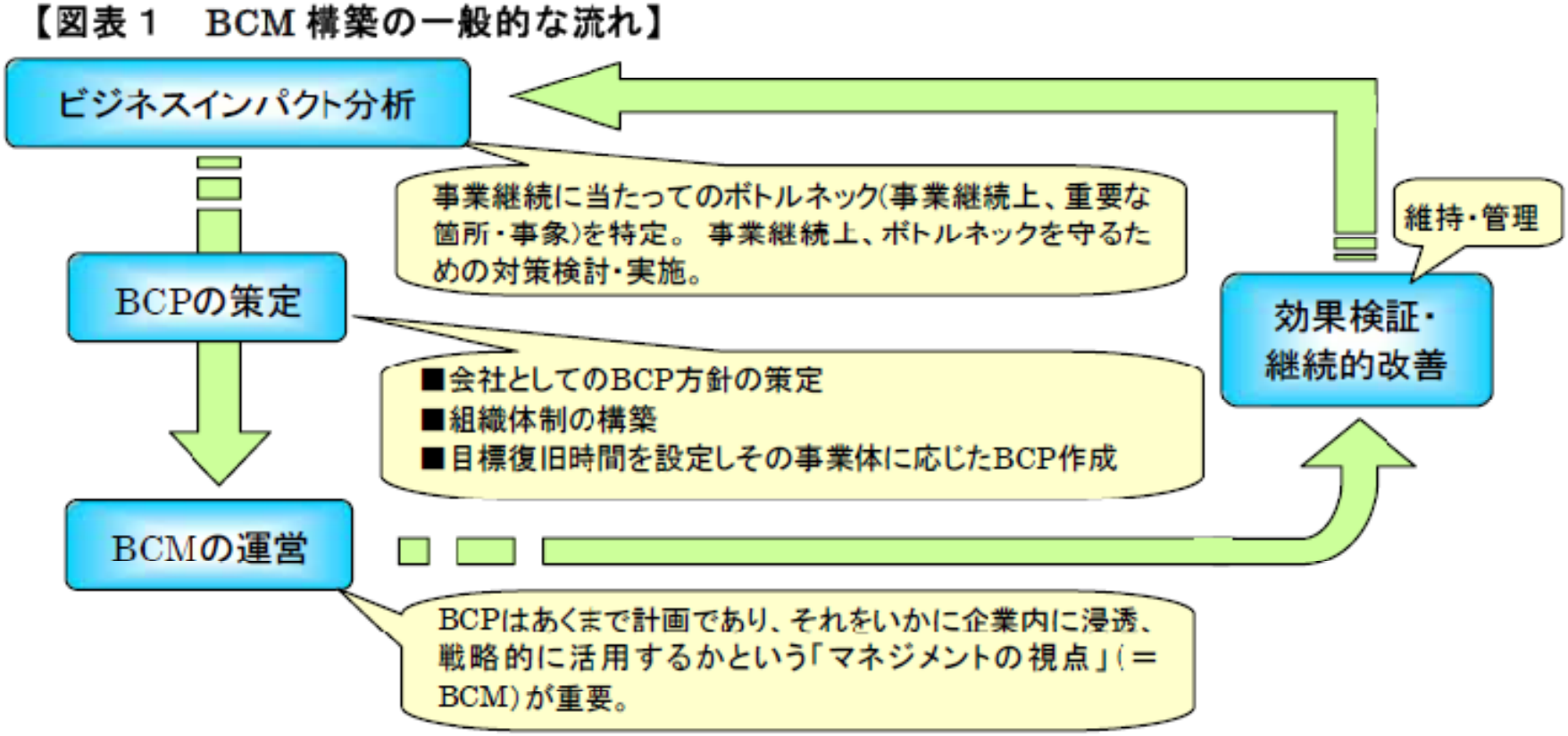
ハード的にどこまで信頼性を確保できるのか

◆ 実は目的ほどに効果を発揮していない。



IT動向調査2004,JUAS

管理プロセス



事業継続計画策定ガイドライン

大学における感染症対策・事業継続計画BCP (Business Continuity Plan) 例

脅威：感染症 (特に心配されるのが鳥インフルエンザのヒトーヒト感染)

米国の想定では18ヶ月の流行期間 (日本の想定は8週間)

罹患率25%　死亡率0．5－2%と予想

(SARSやエボラ出血熱のような突然の感染症も予想される)

	想定される状況	必要な対応
第一フェーズ (立ち上がり) 国内感染報告のニュース 外出自粛の世論	学生・職員への不安の蔓延	対応の周知
第二フェーズ (蔓延) 国内感染報告の増加 政府から社会活動制限依頼	学内に患者が発生 登校拒否学生の増加 職員も欠席	授業の中止 事務局の縮小 試験実施は困難 (学生、職員とも登校困難) ネット活動のみ実施
第三フェーズ (最流行) **企業活動が縮小** **公共交通機関の運用も縮小** **医療機関がパンク** **インターネットが大混雑**	**学生の10%が罹患、死者も発生** **事務職員も同様** **事実上、機能停止**	**全くの機能停止** **ネット活動のみ実施**
第四フェーズ (終息) 患者数が減少し 社会活動の制限解除	解除通知を受けて徐々に回復	解除通知 後事務処理 ネット活動から実施 (e-Learning授業、テレビ会議)

時期や期間によっては、大学の運営、経営に深刻な影響

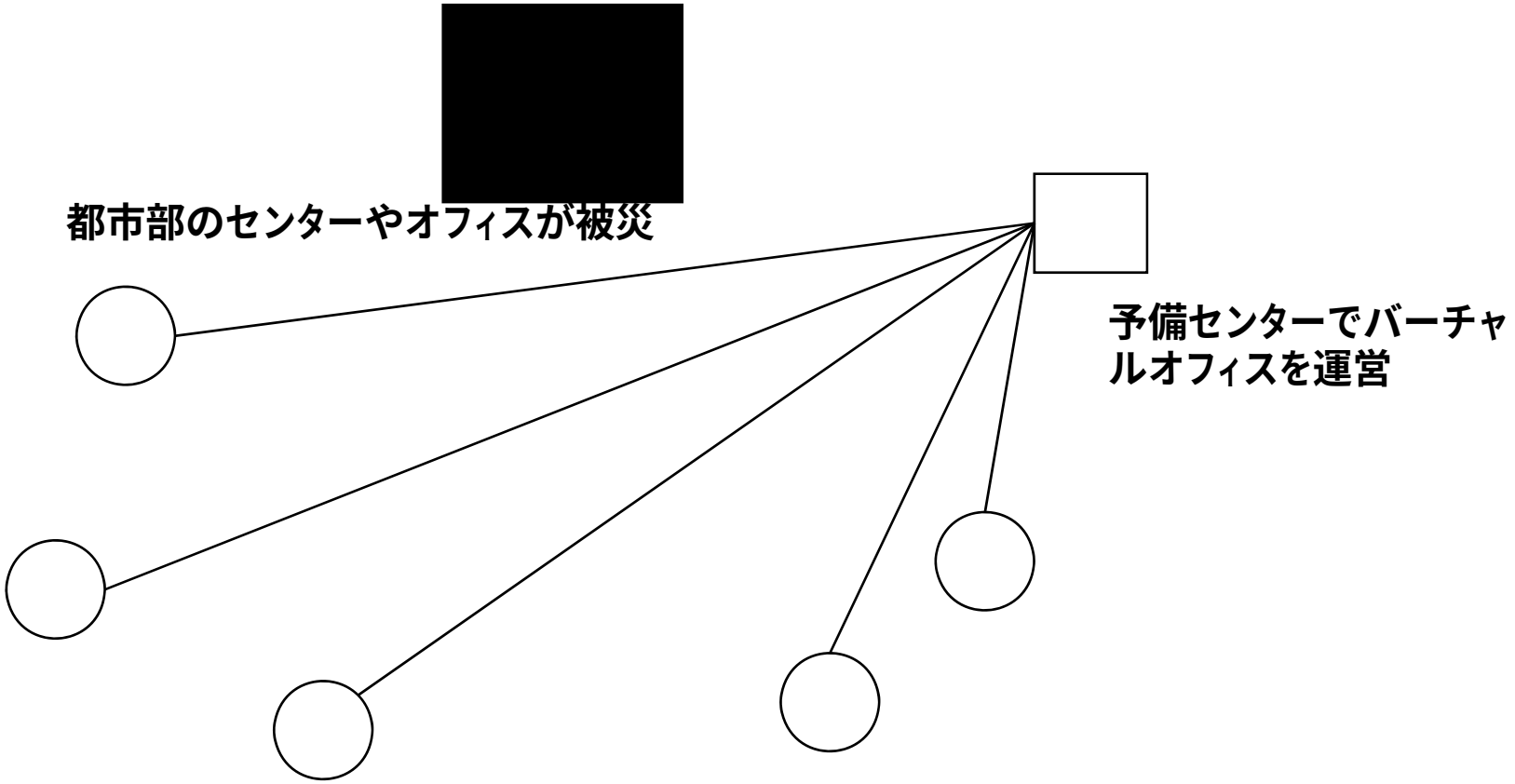
新入生の減少
卒業生の滞留　等

日頃から準備が必要
緊急用の容量も必要
回線確保ができない場合の想定も必要

ブロードキャストによる説明会
テレビ会議、テレワークによる運営
E-Learninngによる授業継続

BCP対策としてのテレワーク

- 米国では、ニューオリンズの災害以降、テレワークの推進が加速



情報システムの信頼性向上に関するガイドライン

◆ METI

表　2　信頼性・安全性の水準に応じた必須・推奨事項及び関連規格等

実施項目	必須／推奨区分			実施区分			活用できる規格・標準・基準・指針・資格等
	システムA	システムB	システムC	情報システム利用者が実施	情報システム供給者が実施	情報システム利用者・供給者両者が合意	
Ⅲ．企画・開発及び保守・運用全体における事項							
１．企画段階における留意事項							
（１）信頼性・安全性水準の利用者・供給者間での合意	◎	◎	◎			○	(4)、(6)、(10)、(16)、(18)
（２）発注仕様への機能要件及び非機能要件の取込と文書化	◎	◎	○	○			
２．開発段階における留意事項							
（１）システムライフサイクルプロセスの確立と文書化	◎	◎	◎		○		(1)、(2)、(3)、(4)、(5)、(6)、(7)、(8)、(9)、(10)、(11)、(13)、(14)、(15)、(16)、(17)、(18)
（２）役割分担・責任権限の利用者・供給者間での合意	◎	◎	◎			○	
（３）機能要件の実現に向けた利用者・供給者間での合意	◎	◎	◎			○	
（４）非機能要件の実現に向けた利用者・供給者間での合意	◎	◎	○			○	
（５）利用者によるシステム要件に関する見解の統一	◎	◎	◎	○			
（６）定量的見積りの実施	◎	○	○		○		
（７）情報システムの複雑化の回避	◎	◎	○		○		
（８）情報システムの障害対応能力の向上	◎	◎	○	○	○		
（９）誤操作等防止への配慮	◎	◎	○	○	○		
（１０）テスト及びレビューの徹底	◎	◎	◎	○	○		
（１１）検収基準の明確化	◎	◎	◎	○	○		
３．保守・運用段階における留意事項							
（１）保守・運用に関する体制等の利用者・供給者間での合意	◎	◎	◎			○	(1)、(2)、(3)、(5)、(7)、(8)、(9)、(10)、(11)、(14)、(16)、(17)
（２）企画・開発・保守・運用の全体を通じたリスク管理	◎	◎	◎	○	○		
（３）保守・不具合の取扱方針の利用者・供給者間での合意	◎	◎	○			○	
（４）恒常的な運用状況の把握	◎	◎	○	○	○		
（５）リリース手順等の整備と訓練	◎	◎	◎	○	○		
（６）問題追跡性の確保	◎	◎	◎	○	○		
４．障害対応に関する留意事項							
（１）緊急時対応の利用者・供給者間での合意	◎	◎	○			○	(6)、(8)、(9)、(11)、(14)、(16)、(17)
（２）原因究明手順等の明確化	◎	◎	○	○	○		
（３）情報システム障害に関する情報の利用者・供給者間での共有化	◎	◎	○	○	○		
（４）関連・類似システムの障害情報収集	◎	○	－	○	○		
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							
（１）経験則のみによらないプロジェクトマネジメントの導入	◎	◎	○	○	○		(3)、(5)、(8)、(10)、(16)、(17)
（２）定量データを活用した管理	◎	○	○	○	○		
（３）健全なプロジェクト運営に向けた活動の実施	◎	◎	◎	○	○		
（４）第三者によるレビュー及び監査の実施	◎	○	○	○	○		
（５）仕様変更の取扱に関する利用者・供給者間での合意	◎	◎	◎			○	

ITサービス継続ガイドライン（案）」

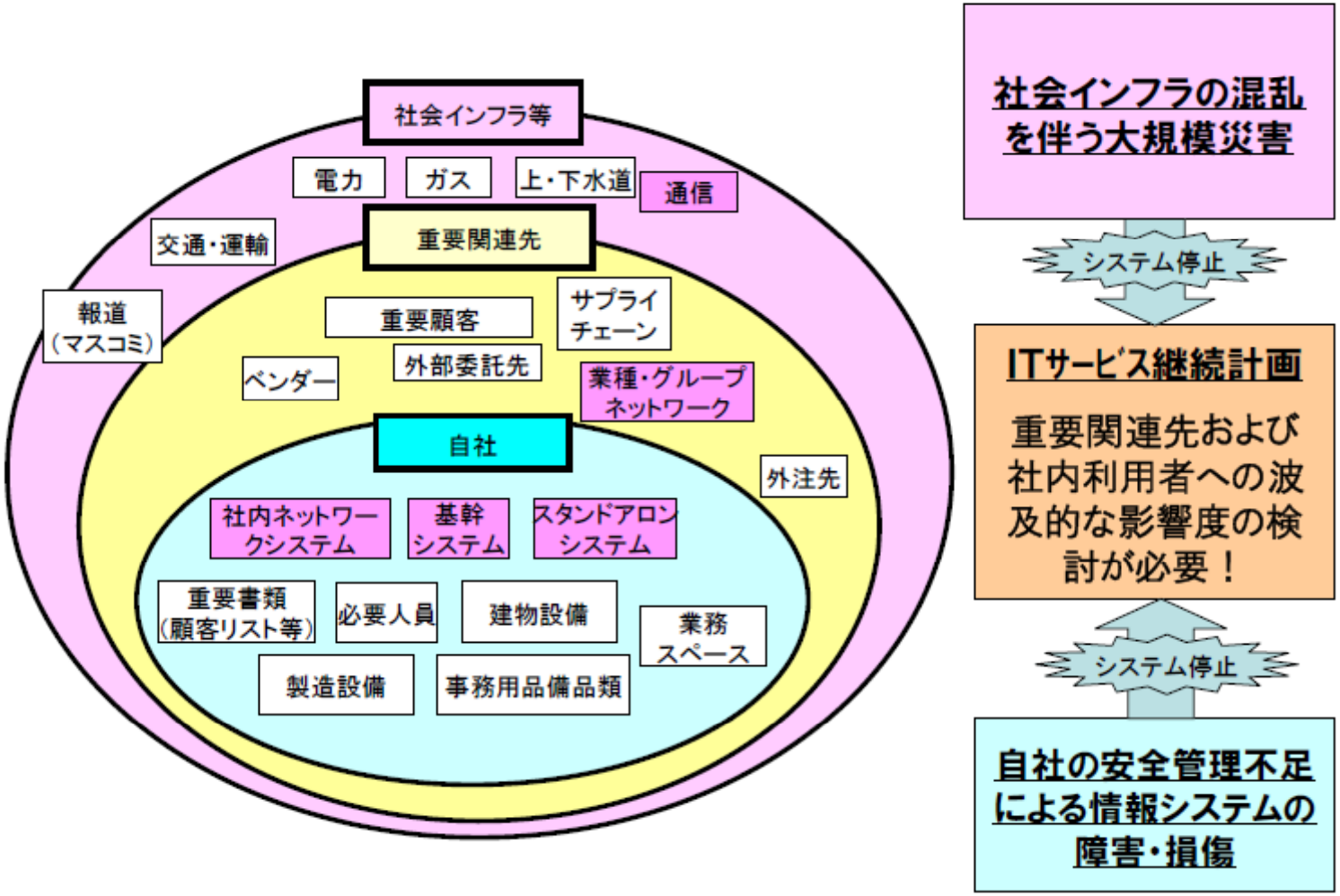


図 4.2-2　情報システムを取り巻く社会環境と IT サービス継続計画

ITサービス継続ガイドライン（案）」

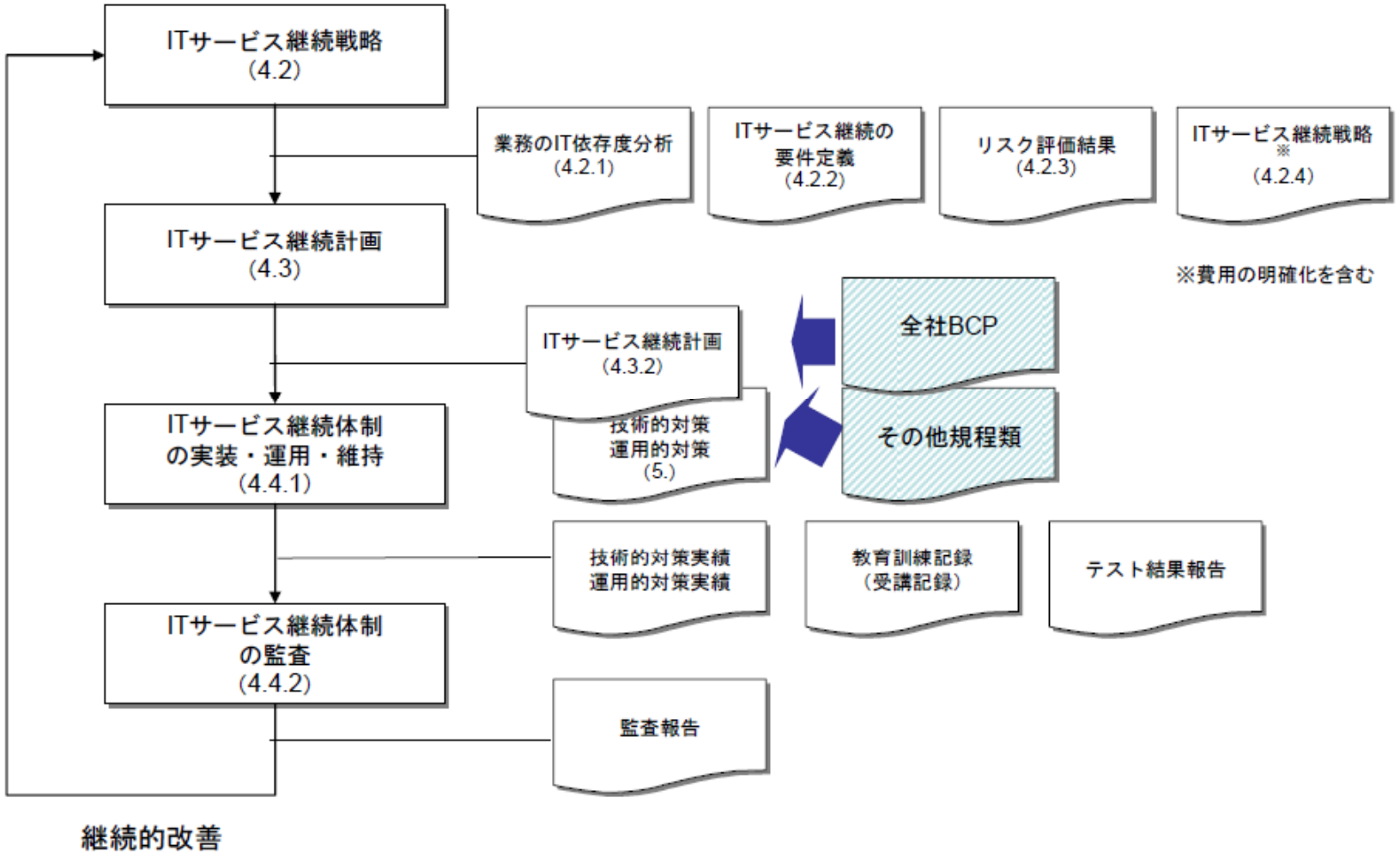


図 4.1-1　IT サービス継続マネジメントのフレームワーク

ITサービス継続ガイドライン（案）」

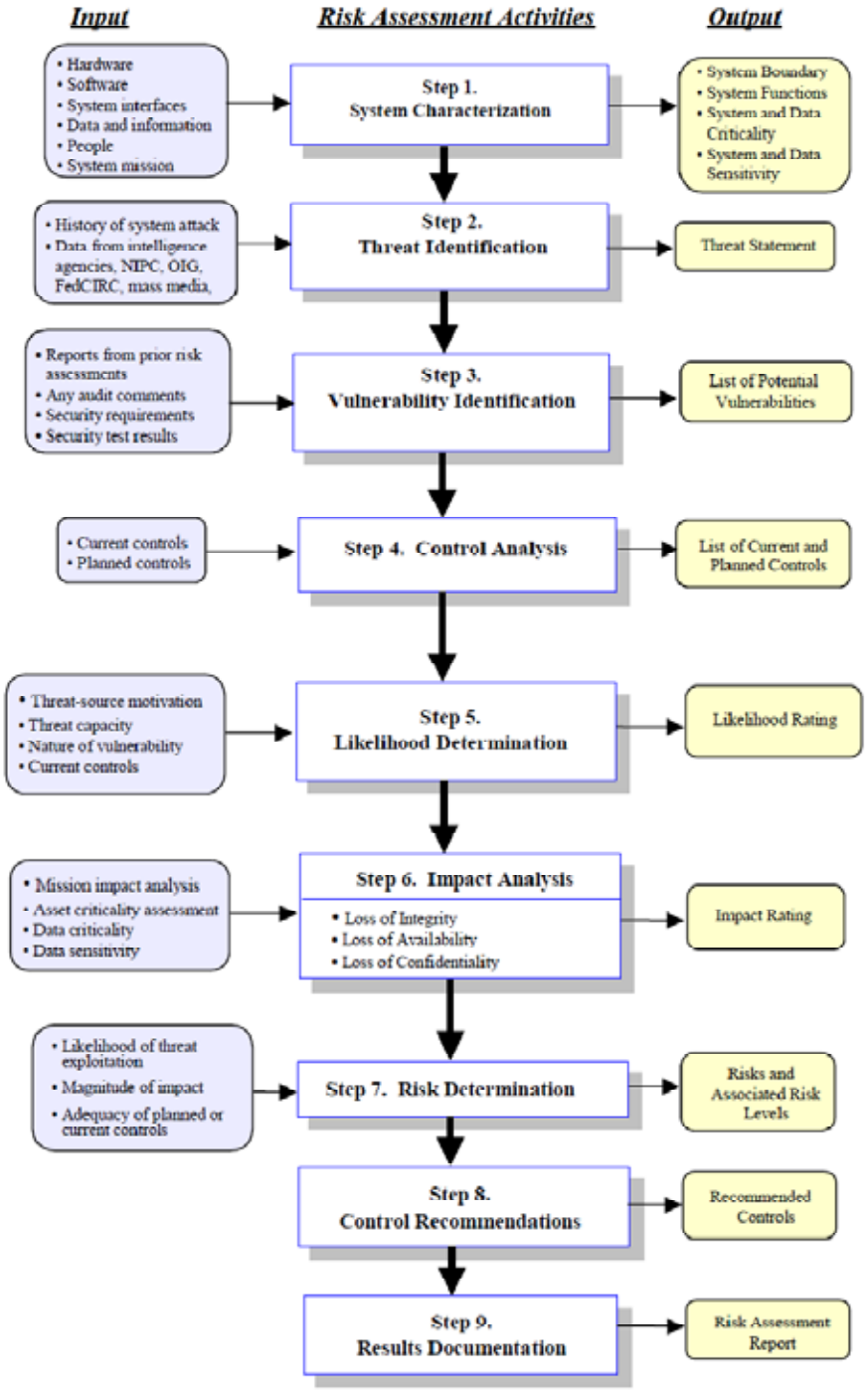
表 5.4-1　テストの種類と概要

テストの種類	実施内容	メリット	デメリット
机上チェック	・ 計画の内容をレビューし、不具合を修正する。 ・ 計画に定めた各種内容の有効性を検証する。	早期に実施可能であり、事業への影響が少ない。必要要員も最小である。	対応能力の向上や対応手順の良否の検証は難しい。
ウォークスルー	・ 計画に定めた各種内容の有効性を検証する。	早期に実施可能であり、事業への影響が少ない。必要要員も最小である。机上チェックよりも、より末端の対応手順を検証できる。	計画自身の整合性の検証が中心であり、計画発動時の具体的な課題提示は難しい。
シミュレーション	・ 計画発動時に予想される状況を前提として、計画の実行に必要かつ十分な情報が記載されていることを確認する。	状況を与えることで、より深い計画の検証を行う。あらかじめ与えられた状況内であるが、これに沿って、例えば対応チームごとに対応手順内容を検証できる。	必要要員は多くなる。

ロールプレイング	・ テスト実施の途中で状況を追加付与し、参加者の状況判断や意思決定の可否、連絡体制などを検証する。	計画を実行する判断者の訓練になり、判断資料の手当てなどが確認できる。	想定状況を多数設定するため、事前準備の負荷は大きい。参加者の十分な知識も必要となる。必要要員は多い。
実機訓練	・ 実際の設備などを用いたテストを実運用及び実作業で行えることを検証する。	代替施設や設備に関して実際の手順を適用し、実効性の有無を確認できる。代替システム切り替えなど	業務に影響する可能性があり、周到な準備が必要である。現場レベルで多数の要員確
		実際の手順を経験できる。	保も必要となる。

リスク管理

NIST

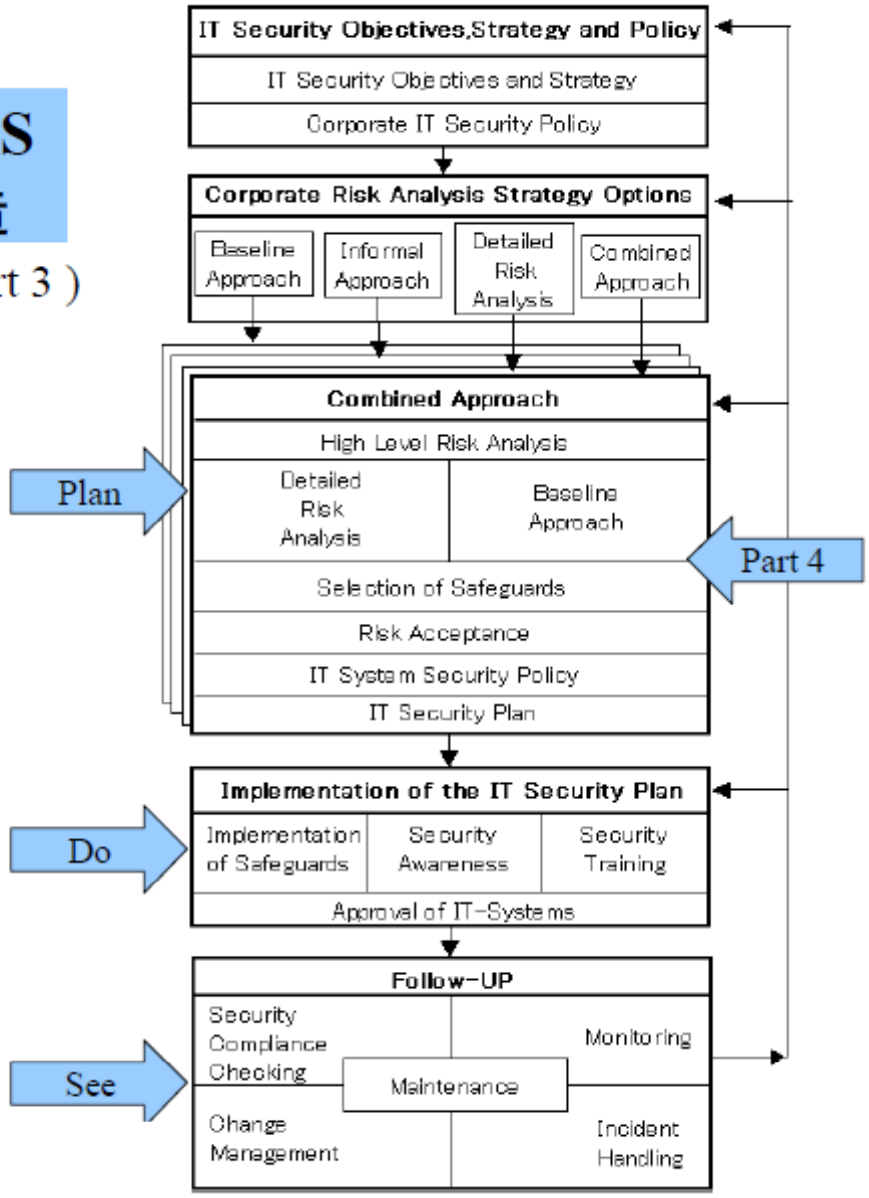

Input
Risk Assessment Activities
Output
• Hardware
• Software
• System interfaces
• Data and information
• People
• System mission
Step 1.
System Characterization
• System Boundary
• System Functions
• System and Data Criticality
• System and Data Sensitivity
• History of system attack
• Data from intelligence agencies, NIPC, OIG, FedCIRC, mass media,
Step 2.
Threat Identification
Threat Statement
• Reports from prior risk assessments
• Any audit comments
• Security requirements
• Security test results
Step 3.
Vulnerability Identification
List of Potential Vulnerabilities
• Current controls
• Planned controls
Step 4. Control Analysis
List of Current and Planned Controls
• Threat-source motivation
• Threat capacity
• Nature of vulnerability
• Current controls
Step 5.
Likelihood Determination
Likelihood Rating
• Mission impact analysis
• Asset criticality assessment
• Data criticality
• Data sensitivity
Step 6. Impact Analysis
• Loss of Integrity
• Loss of Availability
• Loss of Confidentiality
Impact Rating
• Likelihood of threat exploitation
• Magnitude of impact
• Adequacy of planned or current controls
Step 7. Risk Determination
Risks and Associated Risk Levels
Step 8.
Control Recommendations
Recommended Controls
Step 9.
Results Documentation
Risk Assessment Report

GMITS (The Guidelines for the management of IT Security)

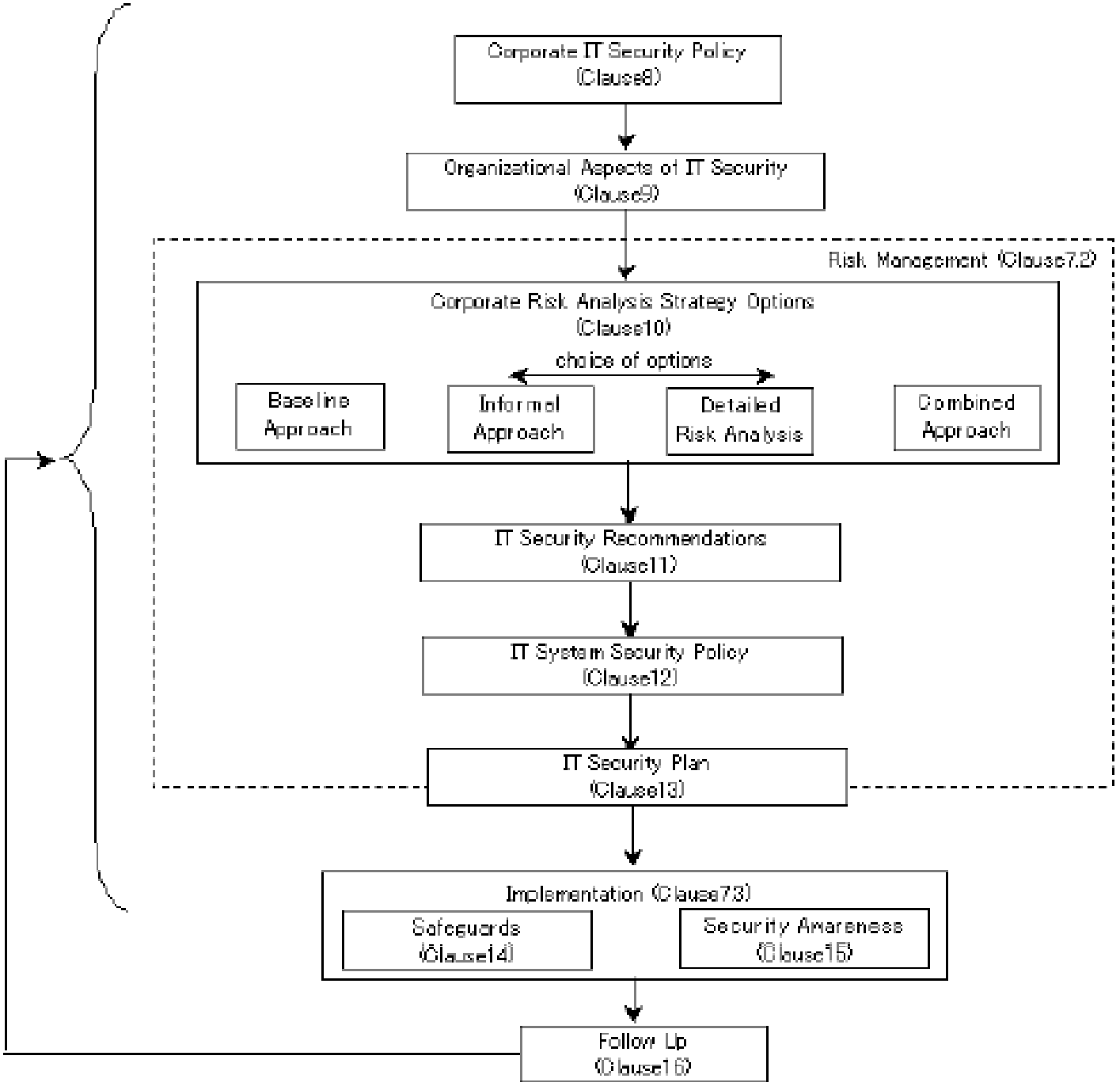
- Part1：Concepts and models for IT Security
- Part2：Managing and planning IT Security
- Part3：Techniques for the management of IT Security
- Part 4：Selection of safeguards
- Part 5：(Management guidance on network security

GMITS の構造

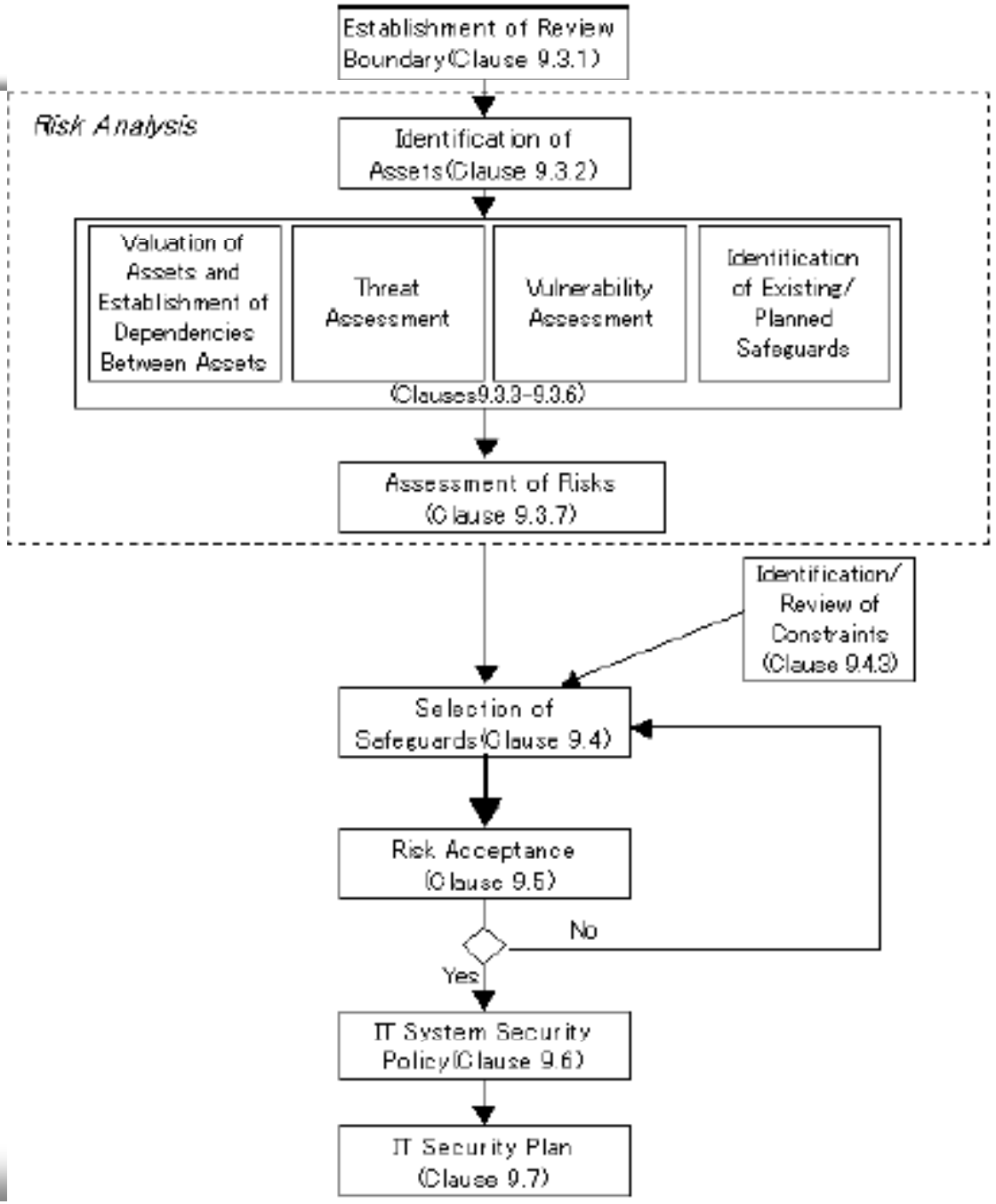
(Part 1 – Part 3)



資料：IPAセキュリティセンター



資料：IPAセキュリティセンター



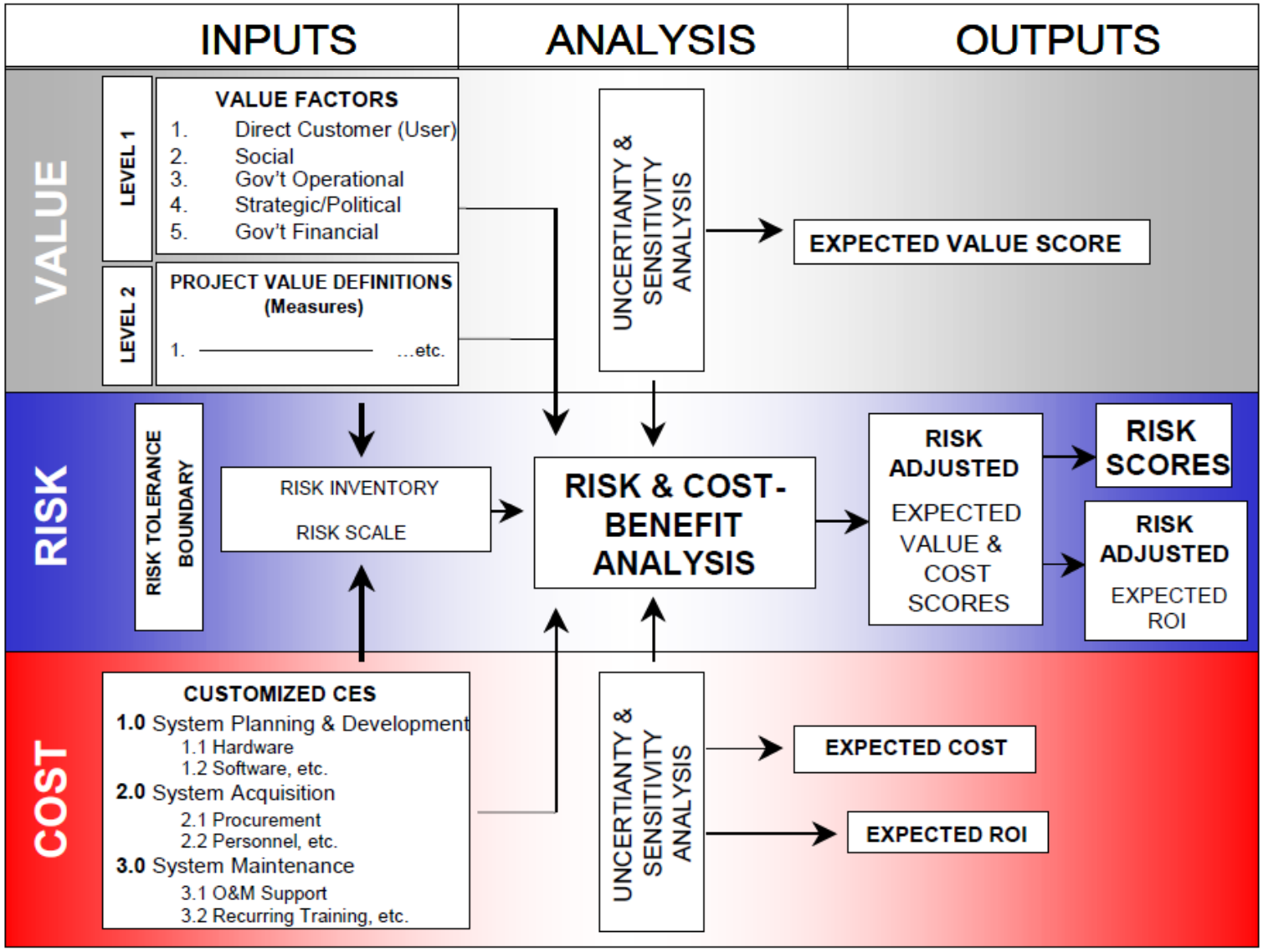
資料：IPAセキュリティセンター

リスク対策

- リスクコントロール
- リスクファイナンス
  - リスク保有
  - リスク移転

IT投資を考える上でもリスクは重要項目

- VMM (Value Measurement Methodology)



迷惑度指数

◆ JUASの提案する方式

事故	迷惑度指数	業務の重要度	時間的要素	影響範囲	影響地域	発生時のピーク性	再発	可能性

リスクマトリックスによる管理

リスクマトリックス

リスク発生見込	インパクト		
	Low (10)	Mid (50)	High (100)
High (1)	Low (10)	Mid (50)	High (100)
Mid (0.5)	Low (5)	Mid (25)	Mid (50)
Low (0.1)	Low (1)	Low (5)	Low (10)

判定リスクレベル

High	運用は継続できるが、早急に対策を取らなければならない
Mid	適切な期間内に対策を取るか計画をしなければならない
Low	リスク対策を取るのか、リスクを受け入れるのかを判断しなければならない

リスク発生見込の定義

	ターゲットとしての魅力	コントロール可能性
High (1)	侵入者等にとって魅力的なターゲットである	コントロールすることができない
Mid (0.5)	侵入者等にとってターゲットである	訓練していればコントロールできる
Low (0.1)	侵入者等にとって魅力のないターゲットである	十分対処できる

インパクトの定義

	コストと損害	業務継続	生命への危険
High (100)	コスト的に大きな損害が出る	組織の目的実施に重大な支障が出る	生命の危険がある
Mid (50)	コスト的に損害が出る	組織の目的実施に支障が出る	けがなどのおそれがある
Low (10)	若干の損害が出る	組織の目的実施に注意が必要になる	

リスク管理表

リスク	判定リスクレベル	推奨される対策	優先度	実施する対策	必要なリソース	責任者	期限	備考

NIST　Risk Management Guide for Information Technology

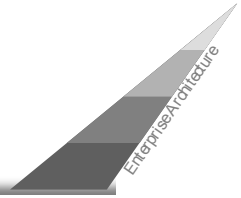
リスク分析表

No	情報資産名	資産価値評価				備考
		①機密性	②完全性	③可用性	資産価値(①+②+③)	
1	サーバ					
2	クライアントPC					
3	ネットワークプリンタ					
4	アプリケーションプログラム					
5	アプリケーションマニュアル					
6	パッケージプログラム					
7	パッケージマニュアル					
8	バックアップデータ					
9	職員名簿					
10	座席表					
11						
12						
13						
14						

No	情報資産名	脅威レベル																																備考
		物理的脅威												技術的脅威												管理的脅威								
		自然災害				故障				持ち出し				漏洩				改ざん				ウィルス感染				故意による				ミスによる				
		①被害レベル	②発生頻度レベル	③脅威レベル(①×②)	④脆弱性レベル	①被害レベル	②発生頻度レベル	③脅威レベル(①×②)	④脆弱性レベル	①被害レベル	②発生頻度レベル	③脅威レベル(①×②)	④脆弱性レベル	①被害レベル	②発生頻度レベル	③脅威レベル(①×②)	④脆弱性レベル	①被害レベル	②発生頻度レベル	③脅威レベル(①×②)	④脆弱性レベル	①被害レベル	②発生頻度レベル	③脅威レベル(①×②)	④脆弱性レベル	①被害レベル	②発生頻度レベル	③脅威レベル(①×②)	④脆弱性レベル	①被害レベル	②発生頻度レベル	③脅威レベル(①×②)	④脆弱性レベル	
1	サーバ																																	
2	クライアントPC																																	
3	ネットワークプリンタ																																	
4	アプリケーションプログラム																																	
5	アプリケーションマニュアル																																	
6	パッケージプログラム																																	
7	パッケージマニュアル																																	
8	バックアップデータ																																	
9	職員名簿																																	
10	座席表																																	
11																																		
12																																		

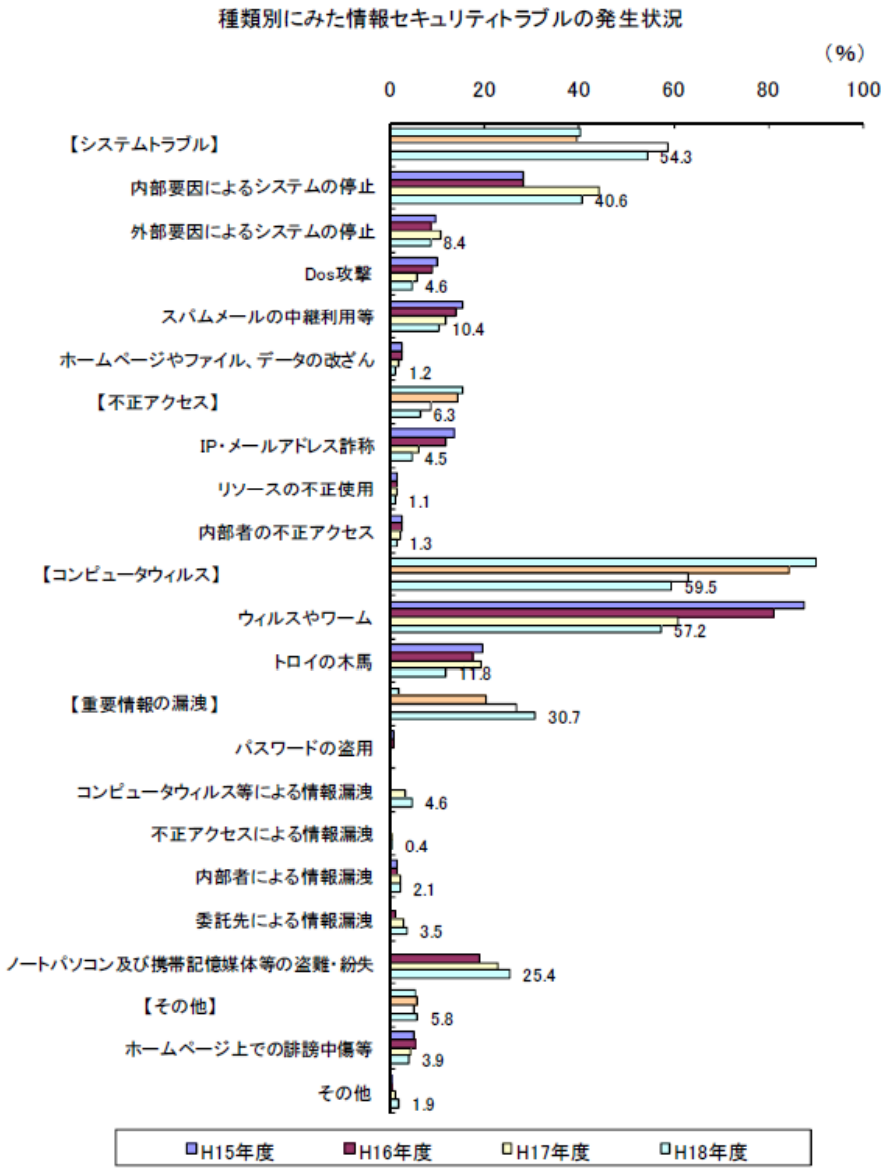
No	情報資産名	リスク評価																備考
		物理的脅威						技術的脅威						管理的脅威				
		自然災害		故障		持ち出し		漏洩		改ざん		ウィルス感染		故意による		ミスによる		
		リスクの大きさ((情報資産評価値＋脅威レベル)×脆弱性レベル)	具体的なリスク	リスクの大きさ((情報資産評価値＋脅威レベル)×脆弱性レベル)	具体的なリスク	リスクの大きさ((情報資産評価値＋脅威レベル)×脆弱性レベル)	具体的なリスク	リスクの大きさ((情報資産評価値＋脅威レベル)×脆弱性レベル)	具体的なリスク	リスクの大きさ((情報資産評価値＋脅威レベル)×脆弱性レベル)	具体的なリスク	リスクの大きさ((情報資産評価値＋脅威レベル)×脆弱性レベル)	具体的なリスク	リスクの大きさ((情報資産評価値＋脅威レベル)×脆弱性レベル)	具体的なリスク	リスクの大きさ((情報資産評価値＋脅威レベル)×脆弱性レベル)	具体的なリスク	
1	サーバ																	
2	クライアントPC																	
3	ネットワークプリンタ																	
4	アプリケーションプログラム																	
5	アプリケーションマニュアル																	
6	パッケージプログラム																	
7	パッケージマニュアル																	
8	バックアップデータ																	
9	職員名簿																	
10	座席表																	
11																		

セキュリティ



セキュリティの状況

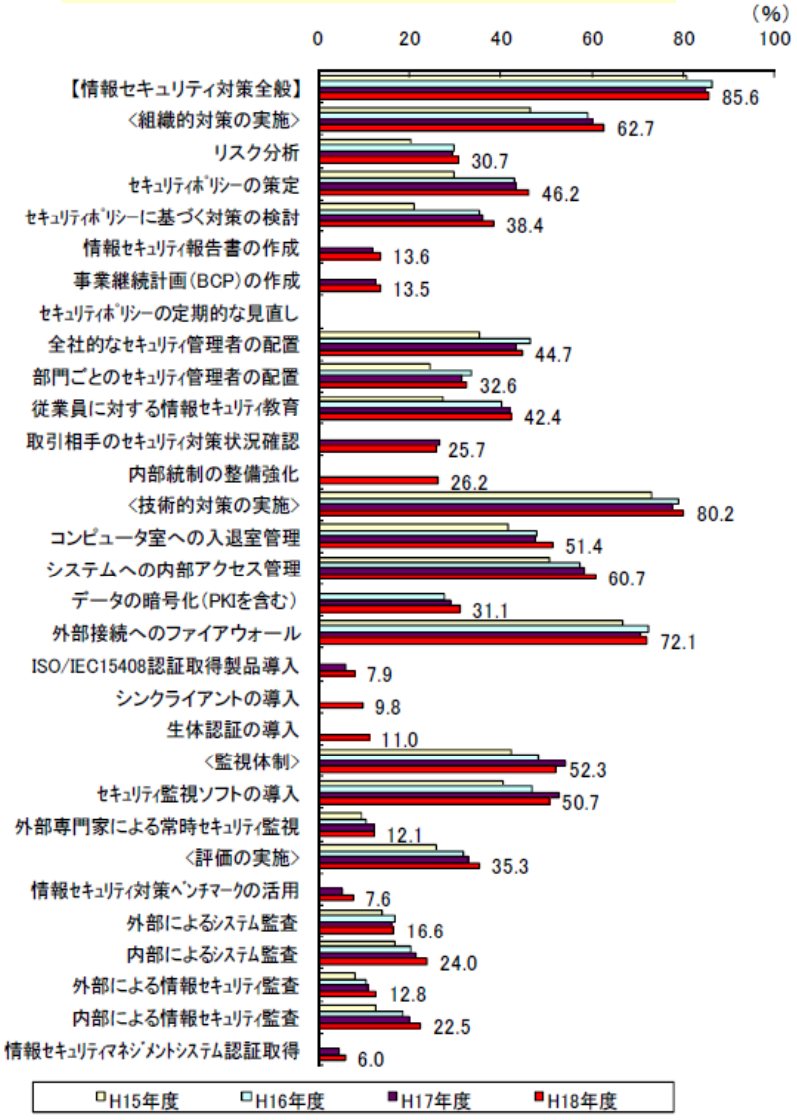
- 平成１９年情報処理実態調査，経済産業省



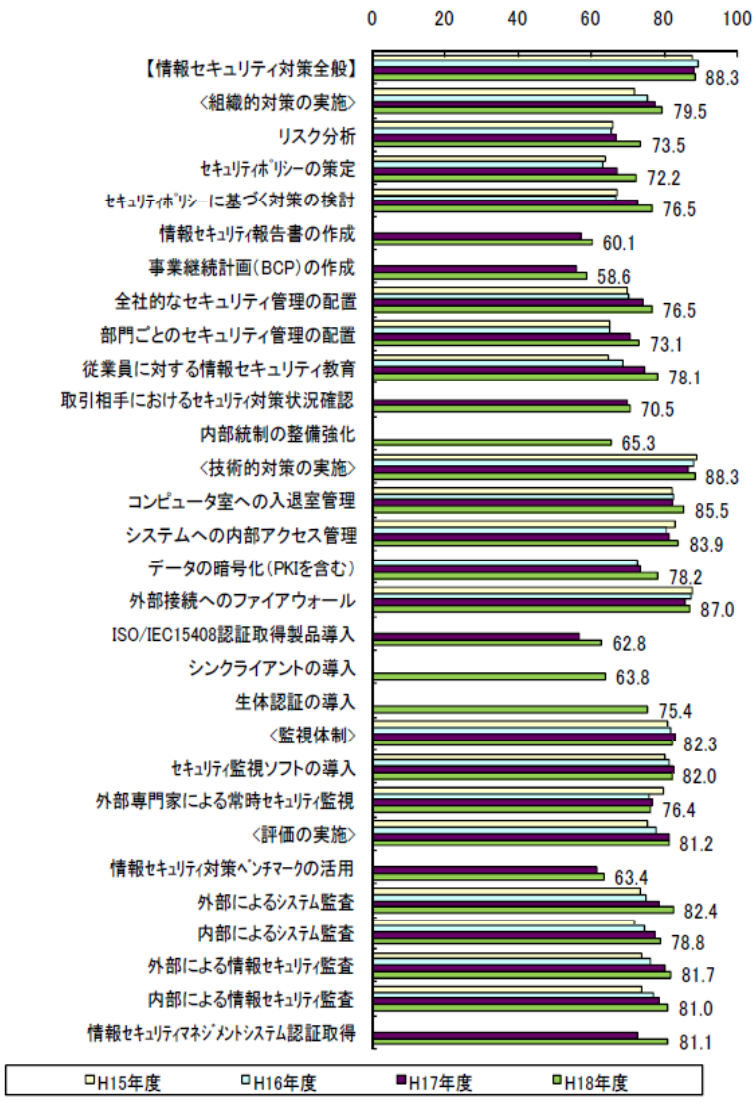
セキュリティ対策

- 平成19年情報処理実態調査，経済産業省

対策を実施している企業の割合



対策がセキュリティ向上に寄与した企業の割合



電子政府基本方針の整理

- 電子政府基本方針から求められる事項は以下のとおりです。
  - 政府機関の情報セキュリティ対策のための統一基準（第三版）
    - 組織と体制の整備、情報についての対策、情報セキュリティ要件の明確化に基づく対策、情報システム構成要素についての対策、個別事項についての対策として整理され、網羅的にセキュリティの基準を設定している
  - 行政機関におけるIT人材の育成・確保指針
    - PJMOの役割
      - 企画段階では、最適化計画の作成における、業務・システムの現状分析、業務の十分な見直しや情報システムを安全・安心に活用するためのセキュリティ面の十分な検討
      - 運用・保守段階では、SLAの管理指標などを活用したシステム運用状況の定期的な把握、制度改正などの環境変化に伴う機能追加などの保守業務の管理及び障害発生時の事業者に対する適切な指示やセキュリティの運用
  - 電子政府推進計画
    - 情報セキュリティ対策等として以下に言及
      - 効果的な情報通信技術の導入
      - 情報セキュリティ対策
      - 府省共通的なセキュリティ機能向上の推進
      - 個人情報保護対策

個別マニュアル群の整備

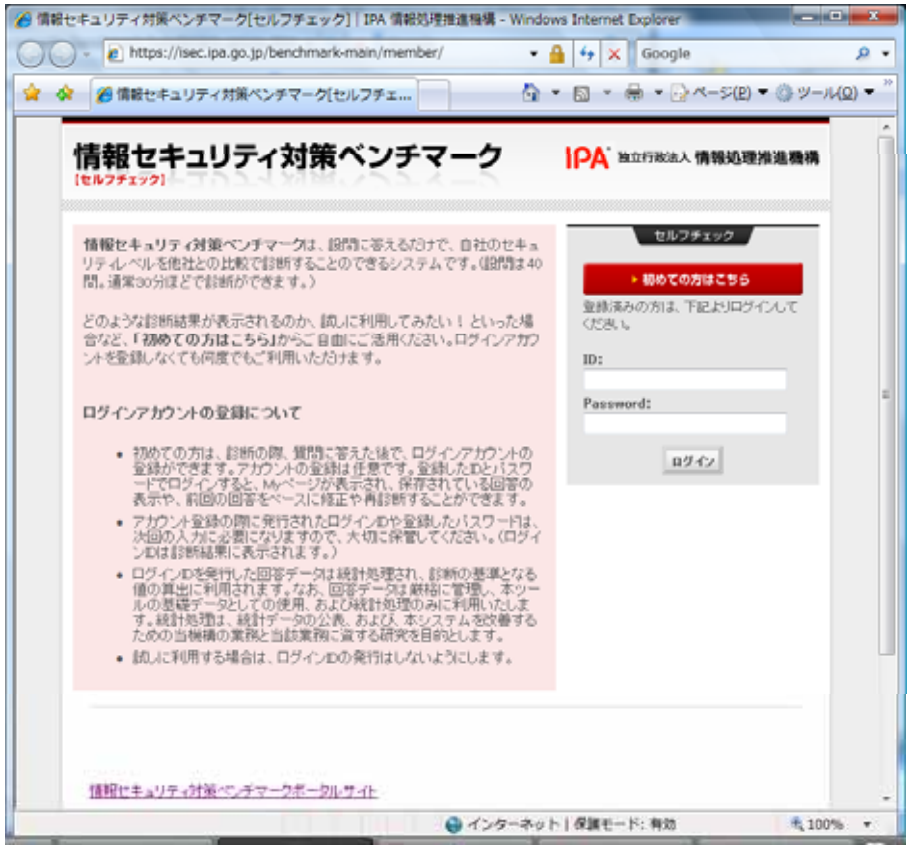
政府機関統一基準適用個別マニュアル群

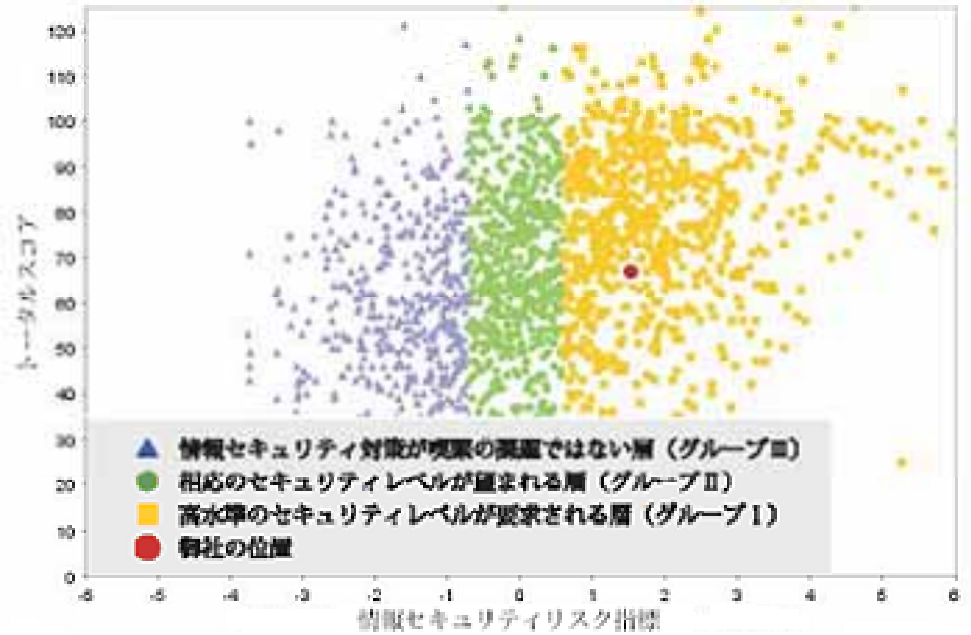
政府機関統一基準適用個別マニュアル群とは、各府省庁が「政府機関の情報セキュリティ対策のための統一基準」(以下、政府機関統一基準と呼ぶ)に基づき策定した省庁基準を、実際に適用する際に対策を円滑に実施するための文書(実施手順、規程及びマニュアル等)を作成する際の参考資料として作成したものです。

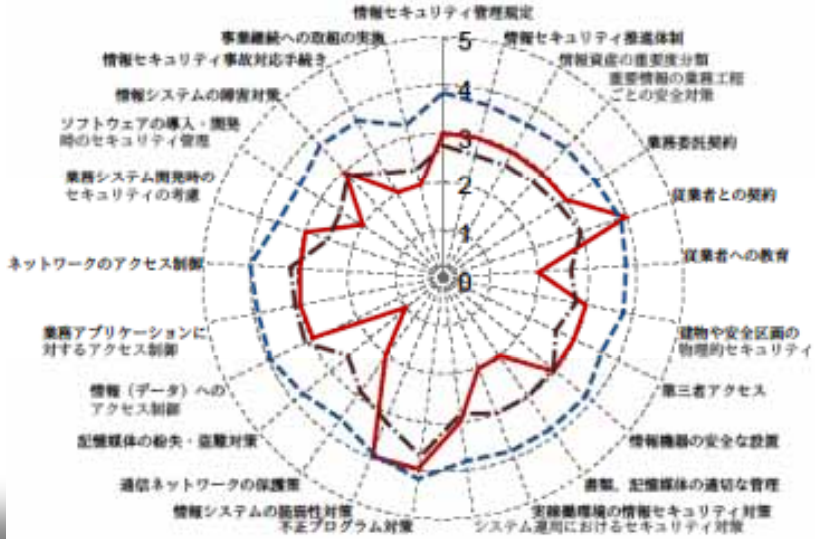
統一基準章番号	マニュアル番号	マニュアル名称	対応する統一基準	
			初版(052版)	第2版(071版)
第2章	DM2-01	政府機関統一基準で定める責任者等の役割から見た遵守事項一覧	○	○
	DM2-02	人事異動等の際に行うべき情報セキュリティ対策実施規程 策定手引書	○	○
		人事異動等の際に行うべき情報セキュリティ対策実施規程 雛形	○	○
	DM2-03	違反報告書に関する様式 策定手引書	○	－
	DM2-04	例外措置手順書 策定手引書	○	○
		例外措置手順書 雛形	○	○
		例外措置申請・終了報告書に関する様式 策定手引書	○	－
		例外措置申請・終了報告書	○	－
	DM2-05	障害等対処手順書 策定手引書	○	－
		障害等対処手順書 雛形	○	－
		障害等報告書に関する様式 策定手引書	○	－
		障害等報告書	○	－
		障害等再発防止策報告書に関する様式 策定手引書	○	－
		障害等再発防止策報告書	○	－
	DM2-06	自己点検の考え方と実務への準備 解説書	○	○
	DM2-07	情報セキュリティ監査実施手順 策定手引書	○	－
第3章	DM3-01	情報の格付け及び取扱制限に関する規程 策定手引書	○	○
	DM3-02	情報取扱手順書 策定手引書	○	○
		情報取扱手順書 雛形	○	○
		機密性3情報印刷書面管理表に関する様式 策定手引書	○	○
		機密性3情報印刷書面管理表	○	○
		機密性3情報移送・提供許可申請書に関する様式 策定手引書	○	○
		機密性3情報移送・提供許可申請書	○	○
		機密性2情報移送・提供届出書に関する様式 策定手引書	○	○
		機密性2情報移送・提供届出書	○	○
第4章	DM4-01	情報システムにおける情報セキュリティ対策実施規程 策定手引書	○	※
		情報システムにおける情報セキュリティ対策実施規程 雛形	○	※
	DM4-02	セキュリティホール対策計画に関する様式 策定手引書	○	※
		セキュリティホール対策計画	○	※
第5章	DM5-01	庁舎内におけるPC利用手順 PCの取扱編 策定手引書	○	※
		庁舎内におけるPC利用手順 PCの取扱編 雛形	○	※
	DM5-02	庁舎内におけるクライアントPC利用手順 電子メール編 策定手引書	○	※
		庁舎内におけるクライアントPC利用手順 電子メール編 雛形	○	※
	DM5-03	庁舎内におけるPC利用手順 ウェブブラウザ編 策定手引書	○	※
		庁舎内におけるPC利用手順 ウェブブラウザ編 雛形	○	※
	DM5-04	モバイルPC利用手順 策定手引書	○	※
		モバイルPCの利用手順 雛形	○	※
	DM5-05	サーバ設定確認実施手順 ウェブサーバ編 策定手引書	○	※
	DM5-06	電子メールサービス提供ソフトウェアのセキュリティ維持に関する規程 策定手引書	○	※
		電子メールサービス提供ソフトウェアのセキュリティ維持に関する規程 雛形	○	※
第6章	DM6-01	機器等の購入における情報セキュリティ対策実施規程 策定手引書	○	○

内閣府

IPAツール


情報セキュリティ対策ベンチマーク[セルフチェック] | IPA 情報処理推進機構 - Windows Internet Explorer
https://isec.ipa.go.jp/benchmark-main/member/
Google
情報セキュリティ対策ベンチマーク
【セルフチェック】
IPA 独立行政法人 情報処理推進機構
情報セキュリティ対策ベンチマークは、設問に答えるだけで、自社のセキュリティレベルを他社との比較で診断することのできるシステムです。(設問は40問。通常30分ほどで診断ができます。)
どのような診断結果が表示されるのか、試しに利用してみたい！といった場合など、「初めての方はこちら」からご自由にご活用ください。ログインアカウントを登録しなくても何度でもご利用いただけます。
ログインアカウントの登録について
初めての方は、診断の際、質問に答えた後で、ログインアカウントの登録ができます。アカウントの登録は任意です。登録したIDとパスワードでログインすると、Myページが表示され、保存されている回答の表示や、前回の回答をベースに修正や再診断することができます。
アカウント登録の際に発行されたログインIDや登録したパスワードは、次回の入力に必要になりますので、大切に保管してください。(ログインIDは診断結果に表示されます。)
ログインIDを発行した回答データは統計処理され、診断の基準となる値の算出に利用されます。なお、回答データは厳格に管理し、本ツールの基礎データとしての使用、および統計処理のみに利用いたします。統計処理は、統計データの公表、および、本システムを改善するための当機構の業務と当該業務に資する研究を目的とします。
試しに利用する場合は、ログインIDの発行はしないようにします。
セルフチェック
初めての方はこちら
登録済みの方は、下記よりログインしてください。
ID:
Password:
ログイン
情報セキュリティ対策ベンチマークポータルサイト
インターネット | 保護モード: 有効
100%


トータルスコア
情報セキュリティリスク指標
情報セキュリティ対策が現状の満足ではない層（グループⅢ）
相応のセキュリティレベルが維持される層（グループⅡ）
高水準のセキュリティレベルが要求される層（グループⅠ）
貴社の位置


情報セキュリティ管理規定
事業継続への取組の実施
情報セキュリティ推進体制
情報セキュリティ事故対応手続き
情報資産の重要度分類
重要情報の業務工程ごとの安全対策
情報システムの障害対策
ソフトウェアの導入・開発時のセキュリティ管理
業務委託契約
業務システム開発時のセキュリティの考慮
従業者との契約
ネットワークのアクセス制御
従業者への教育
業務アプリケーションに対するアクセス制御
建物や安全区画の物理的セキュリティ
情報（データ）へのアクセス制御
第三者アクセス
記憶媒体の紛失・盗難対策
情報機器の安全な設置
通信ネットワークの保護策
書類、記憶媒体の適切な管理
情報システムの脆弱性対策
不正プログラム対策
システム運用におけるセキュリティ対策

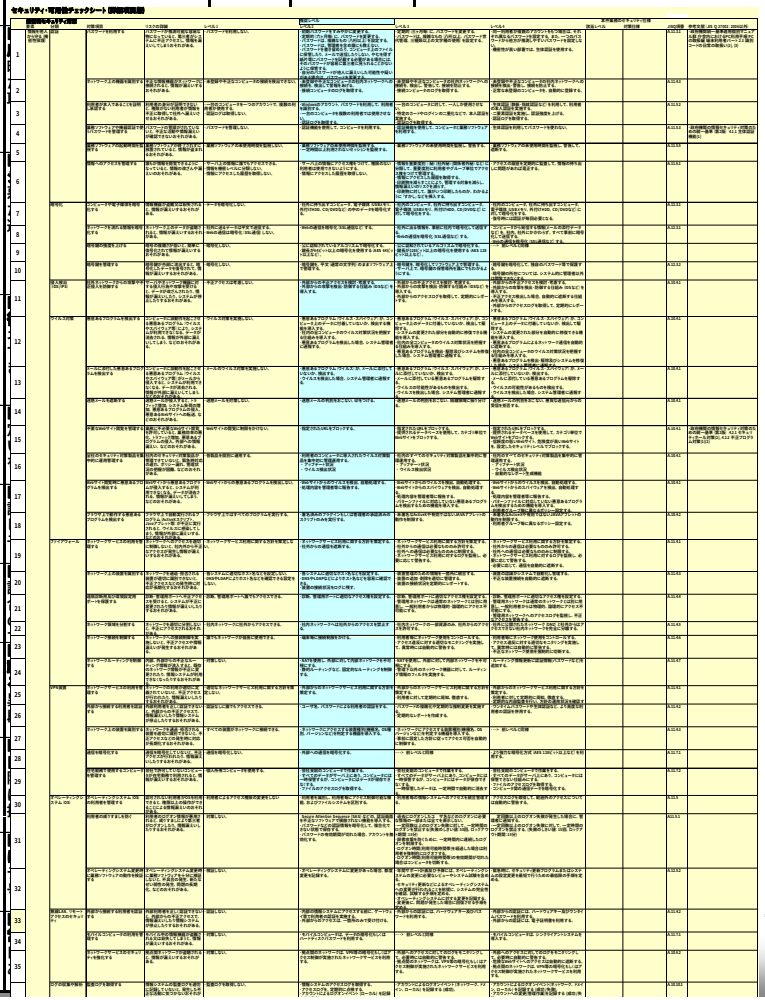
CSAJセキュリティチェックシート

セキュリティ・可用性チェックシート(上位概念定義)

CSAJ/JCSSA情報システムの信頼性向上のための取引慣行・契約に関する検討委員会

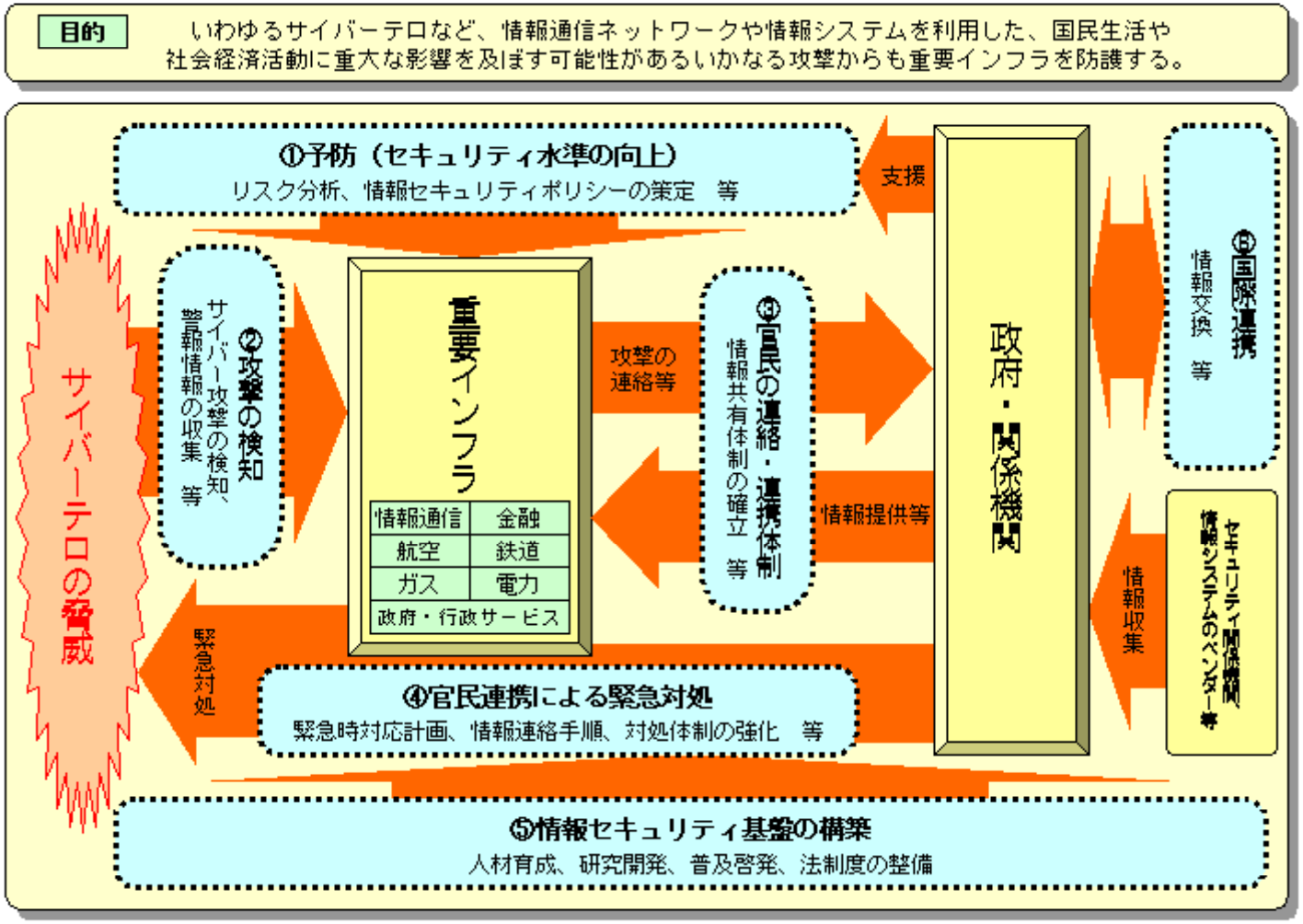
■技術的セキュリティ対策

対策項目	リスクの詳細	参考情報				本件業務での対応	
		レベル1	レベル2	レベル3	レベル4	対応レベル	仕様、候補製品等
■認証 情報を参照している人が本人であることを証明する。	情報を参照している人が、本人であるかを管理していないと、他人に重要な情報を見られる可能性がある。	**■何も決められていない** 情報を誰が参照しているか特定できない状態。	**■個人を認識できる** パスワードを利用して、個人を認識できるようにする。	**■本人認証の強化** 特定のカードやログインの二重化などで、本人認証を強化する。	**■絶対的な本人認証** 生体認証等を組み合わせ、定期的なポリシー変更を実施する。		
■アクセス権 個人情報、企業情報によって、アクセスできる人を制限・管理する。	誰でも情報アクセスできるようになっていると、削除、改ざん、複製、持ち出しされたりする。	**■何も決められていない** 情報に誰でもアクセスできてしまう。	**■コンピュータ単位で設定できる** サーバ単位、フォルダ単位で、個人・グループがアクセスできように設定する。	**■認証情報に基づき資源単位でアクセス権が設定できる** ファイル単位で、個人・グループがアクセスできるように設定する。	**■資源単位でアクセスした内容の収集、分析ができる** アクセスされた情報（ログ）を収集・分析できる。		
■暗号化 情報を暗号化して、紛失・盗難・盗聴の対策を施す。	情報機器（コンピュータやUSBメモリなど）が盗難又は紛失することにより、情報が漏えいするおそれがある。	**■何も対策されていない** 社外に持ち出すデータ、社内のコンピュータのデータに暗号化が実施されていない。	**■モバイルコンピュータやUSBメモリ単位で暗号化して持ち出す** 社外に持ち出すコンピュータ、USBメモリなどの中に入っているデータを暗号して持ちだす。	**■全てのコンピュータについて、データを暗号化する** 社内のコンピュータ、社外に持ち出すコンピュータ、業務で使用するUSBメモリ、外付けHDD、CD/DVDなど情報を書き込めるものに対して暗号化をする。			
■ウイルス等の悪意あるプログラムの取り扱い及び検出する機能の導入 悪意あるプログラムから情報資産を守る。	コンピュータに誤動作を起こさせる悪意あるプログラム（ウイルスやスパイウェア等）により、システムが利用できなくなる、データが消去される、情報が外部に漏えいしてしまう、などのおそれがある。	**■何も決められていない** ウイルス対策を実施していない。	**■ウイルス等を検出し侵入を停止・警告できる** コンピュータ上で悪意あるプログラムを検出して削除し、警告できる。	**■全システムに対するウイルス対策と集中管理** ネットワーク機器やコンピュータなど複数の対象に対して、悪意あるプログラムを検出、削除するための機能を導入し、被害状況の収集や定義ファイルの更新を集中的に管理できる。			
■ネットワークの運用 ネットワークを流れるデータ量の管理をする。	ネットワーク障害や大量のデータ転送により、ネットワークが正常に利用できなくなるおそれがある。	**■何も決められていない** ネットワーク管理ツールもしくはサービスを導入していない。	**■管理ツールを導入する** 障害検知やネットワーク負荷を検知するツール、サービスを導入する。	**■冗長化する、使用状況を監視して記録できるようになる** ネットワーク機器を冗長化して大量データに備えたり、ネットワーク障害時にネットワークが利用できなくなるのを回避したりする。			
■保守 OSやアプリケーション、ハードの保守を行なう。	ハードウェア保守がされていないと、不具合の発生や、故障が発生するおそれがある。	**■何も決められていない** メンテナンス作業をやっていない。	**■障害発生時に対応する** 障害が発生した時点で、保守作業を実施する。	**■定期保守を実施する** 定期的に機器の点検、整備を行い、耐用期間を過ぎた部品は交換する。			
	OS、ミドルウェアの保守がされていないと、不具合の発生や、セキュリティホールによって情報が漏洩するおそれがある。	**■何も決められていない** メンテナンス作業をやっていない。	**■障害発生時に対応する** 障害が発生した時点で、不具合修正版の適用を実施する。	**■定期保守を実施する** 定期的に不具合修正版を取得し、予備機でテストをおこない、適用する。			
	アプリケーション保守がされていないと、不具合や期待する正しい結果が得られないおそれがある。	**■何も決められていない** メンテナンス作業をやっていない。	**■障害発生時に対応する** 障害が発生した時点で、修正版の適用を実施する。	**■定期保守を実施する** 定期的に修正版を取得し、予備機でテストをおこない、適用する。			
■機器運用監視 サーバ、ネットワーク機器の稼働監視を行う。	システムの状況を把握できないことにより、障害の対応が遅れて情報システムへのアクセスが長時間停止するおそれがある。	**■何も決められていない** サーバ、ネットワーク機器の稼働状況を監視していない。	**■運用状況を遠隔で、手動で把握できる** 遠隔で稼働状況を手動で確認する。	**■運用状況を自動で把握、記録ができる** 稼働状況を常時把握し、異常があれば通知する。			
■障害発生時の対応 障害時の対応マニュアルの整備をする。	システム障害時の対応手順が決められていないと、適切に対応できず、復旧が遅延するおそれがある。	**■何も決められていない** システム障害時の対応を決めていない。	**■障害発生時は代替機を手配する** 代替機を手配し、到着後交換する。データはバックアップを利用する。	**■障害発生時は予備システムに手動で切り替える** 障害が発生した場合は、予備機に手動で切替える。データはバックアップを利用する。			
■データの保護 データが改ざんされないように防御する。	データが保護されていないと、データが改ざんされたり、漏えいしたりするなどのおそれがある。	**■何も決められていない** データの改ざんや、漏えいの対策を施さない。	**■特定のデータ、情報が保護されている** データ・情報をパスワードや暗号化で保護する。	**■すべてのデータ、情報が保護されている** すべてのデータを暗号化やアクセス権で保護する。			
■ログ管理 情報の持ち出し履歴をとって監査の証跡資料として管理する。	情報システムの適切な監査ログが管理されていないと不正な出来事に気づく事ができないおそれがある。	**■何も決められていない** ログを取得していない。	**■ログの取得のみ** ログを保存する。	**■ログを取得し、定期的なレポートを行う** ログを保存し、内容を分析し報告する。			



重要インフラに対する対策

重要インフラのサイバーテロ対策に係る特別行動計画の概要



身近なセキュリティの改善

- セキュリティの基礎の周知と意識の改革
  - テクノロジー、パスワードは絶対ではない
    - 一旦入ったら根こそぎ取られる
    - 暗号化ソフトも必要
  - ＰＣが故障したときの手順は決まってますか？
  - そうはいっても、パスワードは重要
    - パスワード教育は重要だが、あまりやられていない。