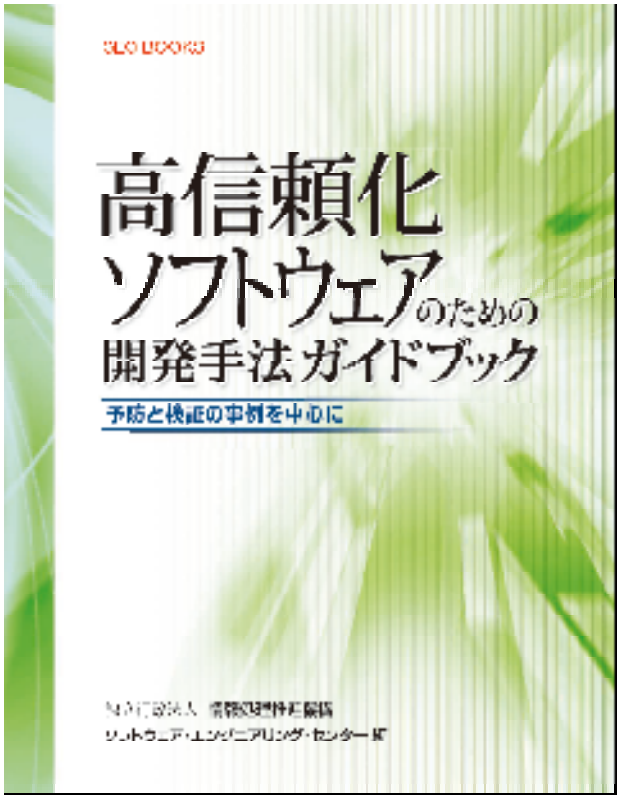
IPA Software Engineering Center
Information-technology Promotion Agency, Japan

情報システムのソフトウェア信頼性のために必要な組織の取組み
～ソフトウェア開発・保守の現場で必要なこと、
現場を支えるために必要なこと ～



「高信頼化ソフトウェアのための開発手法ガイドブック ～予防と検証の事例を中心に～」のご紹介

2011年6月10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
専門委員 （㈱三菱総合研究所）
藤瀬 哲朗

本日の内容

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

■「高信頼化ソフトウェアのための開発手法ガイドブック ～予防と検証の事例を中心に～」のご紹介

本日は、本ガイドブック全体の概要をご説明します！

- はじめに
  - ◆ 高信頼システムとソフトウェア高品質化
  - ◆ 高信頼化が期待される新しい情報システム
  - ◆ このガイドにおける高信頼化に向けた取組み
  - ◆ 高信頼化のための手法WGの活動内容
- 予防活動： トレーサビリティ管理による予防活動
- 予防活動： 障害事例から学ぶ予防活動
- 検知活動： 品質レビュー手法のポジショニングマップ
- 検知活動： テスト技法ポジショニングマップ
- 検知活動： テスト網羅性の高度化技法
- 検知活動： 高度化技法（ピンポイントテスト）
- 【参考】エンタプライズ系代表的企業10社のテスト実態分析
- 第2部： 各社の信頼性向上への取組事例の紹介

テストの話は、単独で詳細なセミナーを企画したいと考えております。

はじめに

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

- 「高信頼化ソフトウェアのための開発手法ガイドブック
  ～予防と検証の事例を中心に～」

⇒　IPAソフトウェア・エンジニアリング・センターに2008年6月から2010年6月まで設置されていた「高信頼化のための手法WG」でまとめられたものです。

高信頼化のための手法WG（継承略　2011年3月現在）

WGリーダー	太田　忠雄	株式会社ジャステック
委員	秋山　浩一	富士ゼロックス株式会社
	育野　准治	日本ユニシス株式会社
	石井　信也	株式会社テプコシステムズ
	岩切　博	三菱電機株式会社
	小野　直子	東京海上日動システムズ株式会社（現在東京海上日動火災保険株式会社）
	大原　道雄	社団法人情報サービス産業協会
	加藤　和彦	NTTデータシステム技術株式会社
	木村　光宏	法政大学理工学部
	黒澤　義次	株式会社JALインフォテック
	鈴木　三紀夫	TIS株式会社
	徳武　康雄	富士通株式会社
	中田　雅弘	株式会社日立製作所
	西　康晴	電気通信大学情報理工学部
	庭野　幸夫	株式会社ジャステック
	水野　浩三	日本電気株式会社
	藤瀬　哲朗	IPA/SEC（現在株式会社三菱総合研究所）
	三毛　功子	IPA/SEC

高信頼システムとソフトウェアの高品質化(1/2)

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

- このガイドでの「高信頼」

⇒

ソフトウェア品質特性 (JIS X0129-1:2003)の信頼性(Reliability)の話ではない

JIS Z 8115:2000　ディペンダビリティ(信頼性)用語

ディペンダビリティ

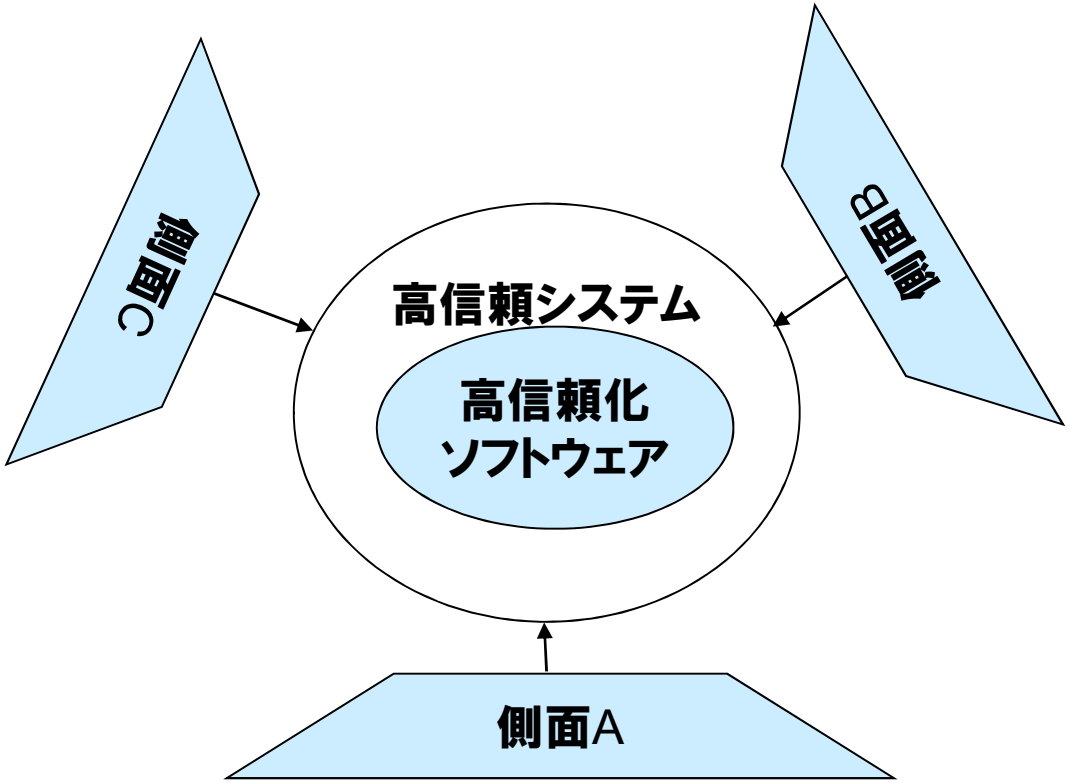
アベイラビリティ性能及びこれに影響を与える要因、すなわち信頼性性能、保全性性能及び保全支援能力を記述するために用いられる包括的な用語。

備考1　ディペンダビリティは非定量的用語として一般的記述に限り用いられる。

備考2　この用語はソフトウェア自身ではなく、ソフトウェアを含むシステム又は製品に適用する。ソフトウェアではシステム又は系の要素からなる製品若しくはサブシステムのディペンダビリティの、ソフトウェア的側面として扱われる。

⇒　障害が少ない(重大な障害を引き起こさない)システムの実現を目指す

高信頼システムとソフトウェアの高品質化(2/2)



このガイドでは、
「ソフトウェア品質という側面」
に注目

高信頼システムの実現のために
ソフトウェアのどの品質特性を
どのように高くすべきか
⇒
品質特性の「信頼性」だけではない

【参考】その他の側面：
アーキテクチャ等

ソフトウェアの開発時を中心に、ソフトウェアの様々な高品質化へのアプローチが

高信頼システム＝障害が少ない（重大な障害を引き起こさない）システム

の実現に、どのように寄与できるのかを現場の知見を中心にまとめた。

【ご参考】 高信頼システム実現のための注釈

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

- 本書では、障害を起しにくいシステムの実現に向け、「現場の技術者視点」かつ「技術的」な話を中心にまとめています。

- ただし、本書では触れていないアーキテクチャ的側面からのアプローチなど信頼性を高める技術も重要です。

- これらの技術論を展開する起点は、「リスクの洗出し」です。
- (ビジネス及びIT)アーキテクトは技術面から「ビジネスリスクやITリスク」を解決する(ビジネス及びITシステムの)設計を実施することが求められます。
  ⇒ 一方で、トレードオフポイントについて説明することも求められるはずです。

- これらのITリスクやビジネスリスクに関する知識が組織として豊富なほど、アーキテクトが技術力を発揮して障害に強いシステム(ITを含むビジネスシステムのこと)を構築することができると考えます。

■ リスク管理が良くできている組織、すなわち「ガバナンス」が非常に重要です。
⇒ 本日後半の「重要インフラ」のお話

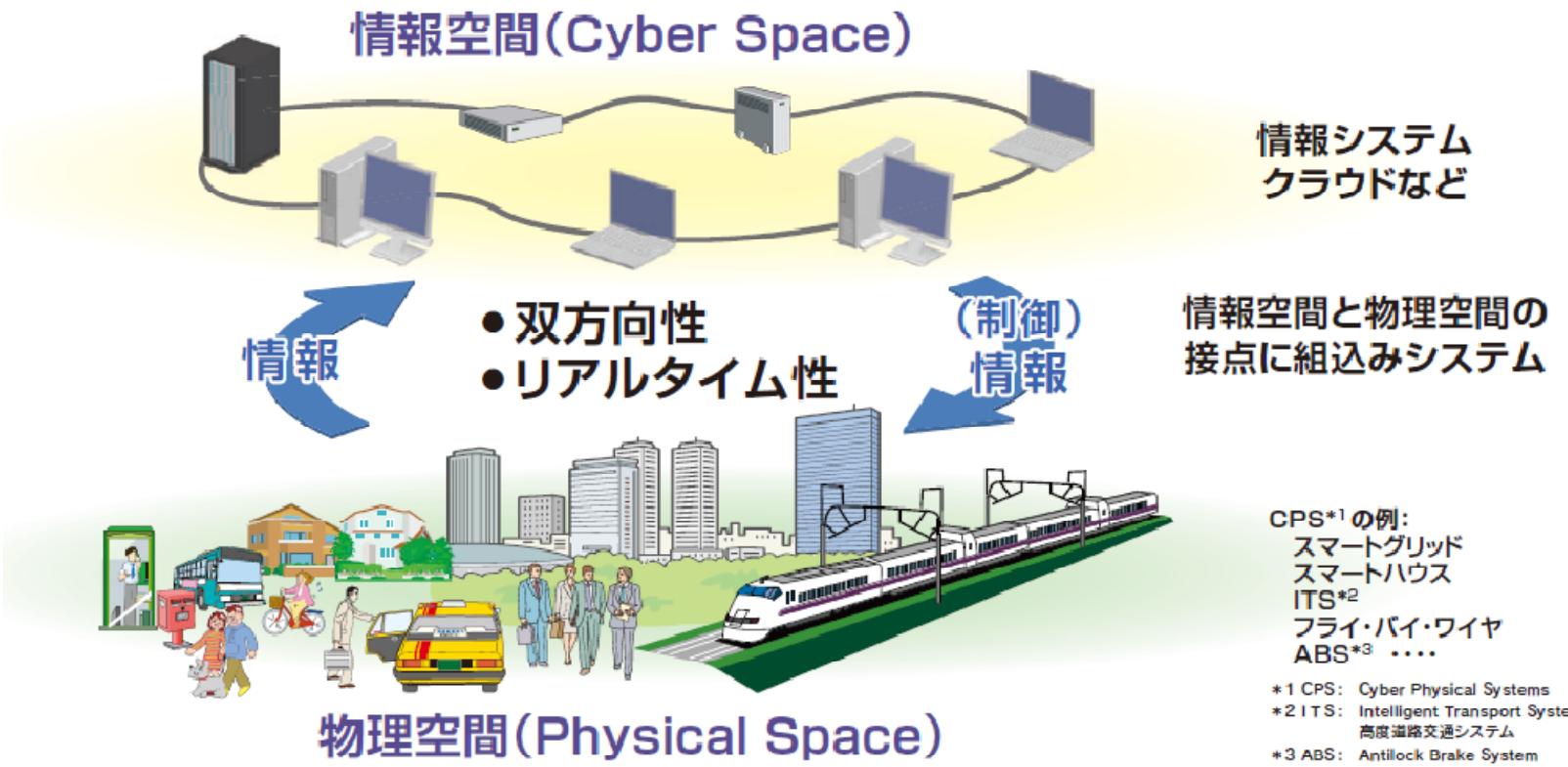
組織として「技術管理」と「ガバナンス」の両立が必要と考えます。

高信頼化が期待される新しい情報システム（1/4）

例 「**CPS（Cyber Physical System）**」もしくは「**統合システム**」

組込みシステムや情報システムがつながったsystem of systems

⇒ 異なった品質のもの同士をつないでしまうことは危険を伴う



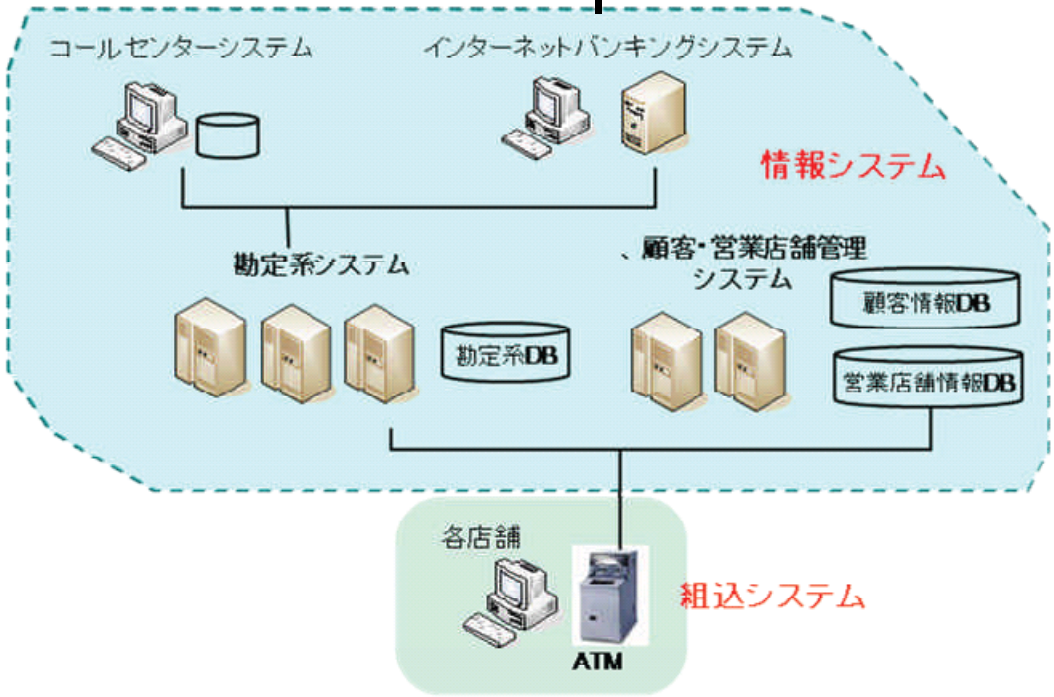
高信頼化が期待される新しい情報システム(1/3)

今後： 携帯デバイスの一機能としての情報システム

スマートフォン利用者が期待する
情報システムの信頼性

スマートフォン

組込システム

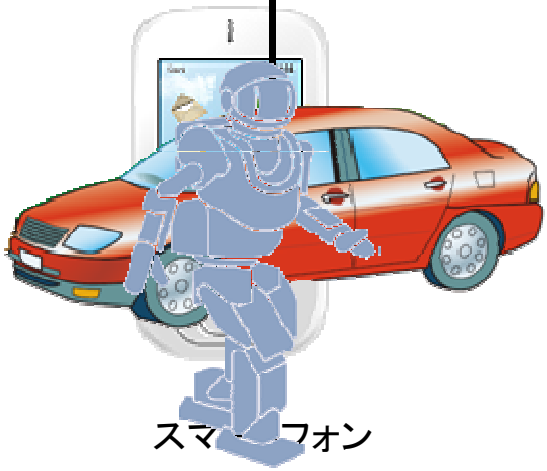


決裁システム

高信頼化が期待される新しい情報システム(2/3)

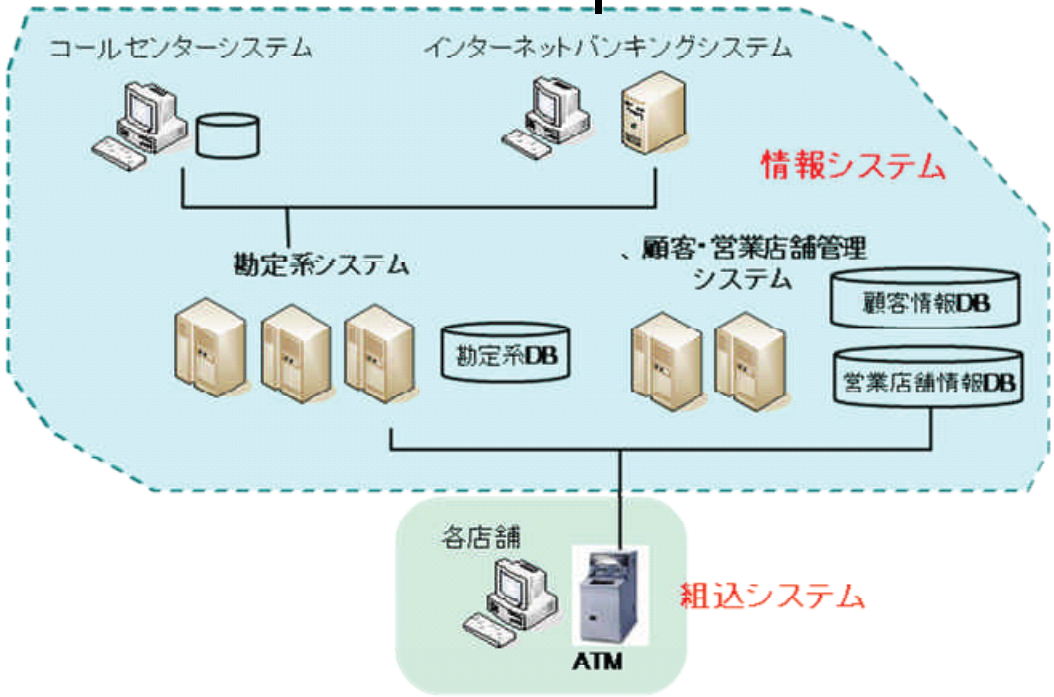
今後： デバイスがもつ一機能としての情報システム

組込みシステム利用者が期待する
情報システムの信頼性



スマートフォン

組込システム



いつの間にか人間が介在しない！

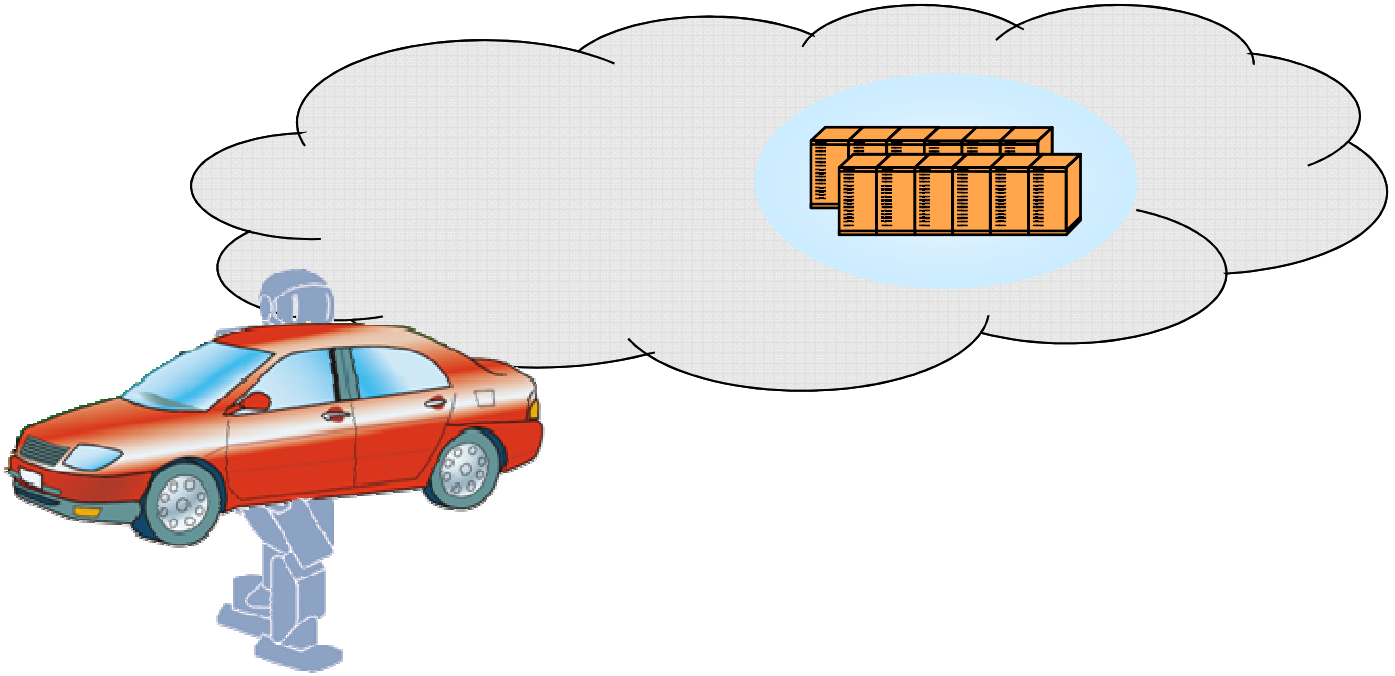
決裁システム

高信頼化が期待される新しい情報システム(3/3)

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

今後： デバイスがもつ一機能としてのデバイスクラウドサービス

組込みシステムが期待する
情報システムの信頼性



組込システム

このガイドにおける高信頼化に向けた枠組み

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

（開発プロセスからの視点）

予防活動　（欠陥混入の抑止）

- 技法（設計など）の開発・適用　===>　形式手法など
- 要件管理（機能および非機能要件）===>　要件管理手法（トレーサビリティなど）

上流工程　（各工程成果物）　下流工程

修正活動

検知活動　（製品の検証および妥当性確認）

設計検証方法
- レビュー（インスペクション、ウォークスル等）
- シミュレーション
- プロトタイピング

テスト方法（テスト網羅性の確保）
- 単体テスト
- 統合テスト
- システムテスト
- 受入れテスト

高信頼化のための手法WGの活動内容(1/2)

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

- 国内企業のベストプラクティス事例を収集し、特徴を分析
  （14企業・団体　⇒　7企業分を本に掲載）
  - 予防活動および検知活動に関わる各社事例紹介と討議

⇒ 「高信頼化ソフトウェアのための開発手法ガイドブック」
第2部　事例編

- 代用特性を利用した信頼性向上への工夫事例を収集
- エンタプライズ系代表企業「10社」のテスト実態分析(テストの観点の分析)

⇒ 「高信頼化ソフトウェアのための開発手法ガイドブック」
第1部　総　論

高信頼化のための手法WGの活動内容(2/2)

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

■ 国内企業のベストプラクティス事例を収集し、特徴を分析

No.	事例発表企業・団体	予防活動	検知活動		備考
			設計検証技法	テスト技法	
1	富士ゼロックス			●	テスト技法ポジショニングマップ、組み合わせテスト(HAYST法)
2	住友電工	●			構成管理ツール(予防活動)
3	日本ユニシス			●	単体テストアセスメント
4	日立製作所	●	●		高信頼性の為のデザインレビュー・品質保証支援ツール
5	富士通	●	●		上流工程の第三者検証(トレーサビリティを含む)
6	三菱電機	●	●		重大障害の予防活動
7	テプコシステムズ		●	●	Web系システム開発標準、SQLガイドラインの自動検証等
8	東京海上日動システムズ	●	●	●	レビュー制度、運用設計ガイド、サービスイン後の信頼性向上策等
9	ジャステック	●	●		トレーサビリティによる予防およびレビュー方法
10	JALインフォテック	●			システムプロファイル、リリース判定および監理等
11	NTTデータシステム技術		●	●	セルフチェック、成果物レビュー、テスト項目抽出等
12	TIS			●	ソフトウェアテスト教育
13	日本電気	●			OMCS(オープン&ミッションクリティカルシステム)による信頼性向上
14	PM学会	●	●		品質機能展開(QFD)の活用

高信頼化ソフトウェアのための開発手法ガイドブック(1/4)

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

第1部　総論
- 第1章　高信頼ソフトウェアにおける問題意識（太田）
- 第2章　高信頼ソフトウェアとは（太田）
- 第3章　予防活動および検知活動に係わる手法（太田、（庭野、）秋山）
- 第4章　障害事例から学ぶ予防活動（太田）
- 第5章　トレーサビリティ管理の手法（庭野）
- 第6章　テスト網羅性の高度化技法（秋山、鈴木（、西））
- 第7章　高信頼ソフトウェア開発に関する海外事例および研究動向（藤瀬、木村）

手法の提案ご紹介

第2部　事例編
- 第1章　事例編の概要（太田＆各社担当者）
- 第2章　東京海上日動システムズ（小野）
- 第3章　富士ゼロックス（秋山）
- 第4章　日本ユニシス（育野）
- 第5章　日立（中田）
- 第6章　富士通（徳武）
- 第7章　三菱電機（岩切）
- 第8章　ジャステック（太田）

各社の取組み事例のご紹介

高信頼化ソフトウェアのための開発手法ガイドブック(2/4)

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

第１部　総　論

高信頼化ソフトウェアのための開発手法ガイドブック(3/4)

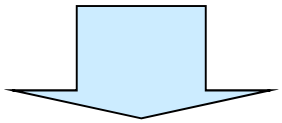
SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

高信頼化ソフトウェアのための開発手法ガイドブック(4/4)

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

検知活動と予防活動

- 検知活動としてのテストだけでソフトウェアの欠陥をすべて見つけようと考えがち
- もちろんソフトウェアテストは、欠陥の検出に非常に有効な方法



「要求　要件　仕様　設計　コード」と「テスト」について、
トレーサビリティを取ってソフトウェア開発活動全体のなかで、
欠陥発生を予防していくということを同時に考えていく必要性

トレーサビリティ管理による予防活動

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

トレーサビリティ ソフトウェアの

製品に関する トレーサビリティ	ソフトウェアアイテム（ソフトウェア部品の最小構成要素）が製品にどのように組み込まれているかが追跡できること。（構成管理によって保全）
文書およびデータに関する トレーサビリティ	顧客の要求事項がソフトウェアのどの要求で実現されているか、およびソフトウェアの要件が製品のどの部分で実現できるか追跡できること。

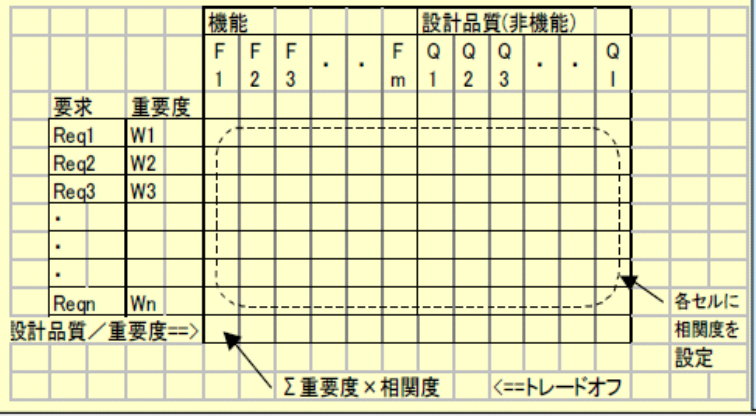
JIS Q 9025;2003

マネージメントのパフォーマンス改善 ―品質機能展開の指針―

（量産品などに偏っているものをソフトウェアへ応用）

トレーサビリティの確保に関する手法事例

（例）QFD(Quality Function Deployment)



品質展開（ＱＦＤ）の期待効果

①ソフトウェア製品に対する要求を品質機能表現にて一覧化
- 要求品質表現規則による要求項目粒度の均一化効果
- 品質機能表現（機能達成水準付加等）による非機能要求の具体化

②要求品質の相対的な優先順位を設定
- 対立要求項目のトレードオフへの客観的根拠
- コスト分析による要求の取捨選択への客観的根拠
- セールスポイント等を可視化した重点指向での要求項目の絞り込み

③要求品質から機能～部品への一貫したトレーサビリティ管理
- 品質機能展開観点の利用による有用性、実現性、経済性、安定性の確認
- 要求品質を技術視点での代用特性に変換しﾄﾚｰｻﾋﾞﾘﾃｨ確保を容易に確認

トレーサビリティ管理による予防活動

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

代用特性という考え方

製品開発において、開発目標を要求品質として洗い出す。

目標達成を計測するために計測可能な要素を代用特性（右表では品質特性）として洗い出す。

例えば、行単位で重み付け平均等により達成度の評価を行う。

2元表の例

品質特性展開表 / 要求品質展開表		1次	操作性				ソフト充実度				形状寸法				質量			話題性			
1次	2次	2次	接続時間	メモリ容量	CPU速度	携帯性	ソフト互換度	ソフト拡張性	キャラクタ充実度	ソフト多様性	本体の厚さ	外形寸法	操作部寸法	開口部寸法	本体質量	操作部質量	附属品質量	意匠性	安全性	注目度	リアル度
使いたくなる	面白い							○	◎									○		○	○
	会話できる							○													
	体感できる								○				○			△					◎
	デザインが良い											◎						◎		○	
ソフトが良い	どのソフトも使える						◎			◎	○										
	ソフトが多く入る			○				○													
	ソフトが作れる						○	◎													
長く楽しめる	多人数で楽しめる							○												◎	
	若者が好む					○												◎		◎	○
	長く使える			○	△					◎											
頑丈である	水に強い													○					○		
	ほこりに強い													◎							
	熱に強い											○							◎		
使いやすい	接続しやすい		◎			○								○							
	コードがない		○			○															
	持ち運べる					◎					○	○			◎		◎				
高性能である	音質が良い																				◎
	ロードが早い				◎					○											
	画像がきれい			○				○													○
操作しやすい	簡単にセーブできる			○	○					○											
	ボタンが押しやすい													◎		○					
	片手で操作できる													◎		○	○				

◎（重み＝5）：強い対応がある
○（重み＝3）：対応がある
△（重み＝1）：対応が予想される

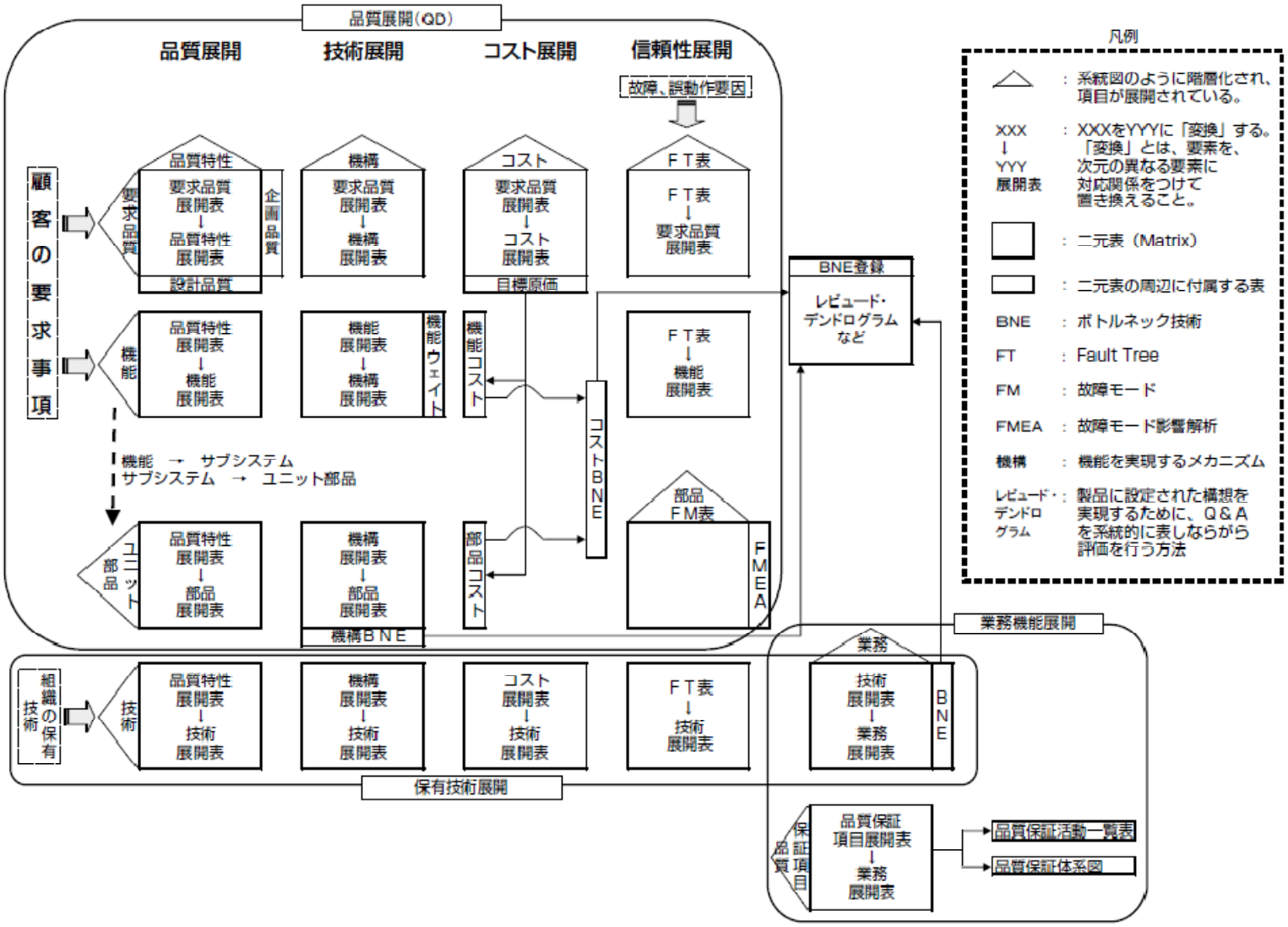
（出典）JIS ハンドブック 2008　品質管理

【参考】 ソフトウェア評価にも適用可能

トレーサビリティ管理による予防活動

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

品質機能展開の全体構想図例



障害事例から学ぶ予防活動

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

障害対応プロセスの違い

		予防活動に結びつかない障害対応プロセス（負のスパイラル）	予防活動に結びついた障害対応プロセス（正のスパイラル）
情報公開・共有		■非オープン（個人、組織、社会） 一部の組織や利害関係者のみが情報を保有し、秘密裏に対応しようとする。	■オープン（個人、組織、社会） 全社レベルで情報を共有し、障害の発生防止を最優先タスクとして対応方針を定め、必要なリソース、期間を準備する。
障害分析		■直接原因のみの追究 早期復旧のため直接原因の究明に注力してしまい、根本原因の究明まで至らない。	■根本原因の追究 発生障害の直接原因究明にとどまらず、障害を誘発させた根本原因を追究し課題を明らかにする。
障害対処	縦展開（二次障害）	■緊急療法のみの対処 根本原因が解決されないまま障害復旧作業（データパッチ等の直接原因の排除）を実施し、後続処理の影響を検討せずに対処完了としてしまうため二次災害が防止できない。	■二次災害防止を考慮した対処 障害復旧のための対処は当然のこと、後続処理への影響を考慮した障害対処案を立案し、これを実施する。
	横展開（類似障害）	■類似障害を点検しない対処 障害の根本原因を組織的に共有しないため、同時期に開発した他の部分の類似する作り込みを点検せず、同様な障害発生を防止できない。	■類似障害を撲滅するための対処 障害の根本原因から関連する開発全体の点検項目を洗い出し、組織的な協力により点検を行い、検知されたリスクを排除する。
情報整理・蓄積		■人的記憶に依存 「障害事例から得られるノウハウを保有システムの開発・保守局面で展開しよう」とする組織的な取り組みは期待できず。ノウハウの展開は担当者個人に依存してしまう。	■新たな開発又は保守作業への利用 障害分析と対処内容を体系的に整理しノウハウとして蓄積しておき、新規開発や次回の改修時に利用することで、システムの品質向上・障害発生防止を図る。

予防活動

代用特性を利用した再発防止策（品質特性による分類）の提案

- 障害の発生要因（根本原因）から想定される品質要求を特定

本書では、障害の発生要因が保守および運用に起因していても、

「ソフトウェア開発での予防活動および検知活動」

としての再発防止策ととらえている。

そこで、根本要求分析を通じて

「本来実現すべき要求品質を想定（想定要求品質）」

する。

- 想定要求品質を代用特性展開により具体的な実装へ

代用特性を利用した再発防止策（品質特性による分類）の提案

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

■ 想定要求品質を代用特性展開により具体的な実装へ

例）

障害の発生原因	想定要求品質	代用特性展開			
		代用特性（1次）		代用特性（2次）	代用特性（3次）
		品質特性	品質副特性		
不正（誤入力）データ投入・運用	操作の容易さ	使用性	運用性	利用時のメッセージをわかりやすくする	画面項目の日本語表示機能の実装
		使用性	運用性	画面操作時のナビゲーションを工夫する	エラー時の項目カラー表示機能の実装
		使用性	習得性	ユーザレベル（初・中・上級）などを配慮した習得容易性を向上させる	本番稼動前でのオペレーション教育の実施
	不正データの早期発見	保守性	安定性	システム保有データの整合性を維持する	不整合データ抽出機能の実装

信頼性向上への知識フレーム

注）代用特性（1次）はJIS X0129-1:2003

⇒　想定要求品質を再発防止策へ

代用特性を利用した再発防止策（品質特性による分類）の提案

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

■代用特性展開のための工夫のポイント

- 開発（要件定義、設計、実装、テスト）、保守および運用フェーズごと、または成果物（例：データベース設計書での品質に関係する記述内容など）ごとに代用特性を分類します。
- 重要度の高い要求品質に的を絞っての展開、さらにボトルネックになりそうなサブシステムやプログラムに対して重点的に展開します。
- 関係する所属メンバでチームを作り、KJ 法やブレーンストーミングを利用して要求品質の収集、整理を行います。
- 「想定品質要求の達成度合いを評価するための手段」、「評価をするためにはどんな手立てが必要か」などを自問してヒントを得ることも有効です。

再発防止の施策への参考として、
「代用特性を利用したシステムの信頼性向上への工夫事例」
⇒ IPA/SEC のWeb サイト
URL： http://sec.ipa.go.jp/reports/20100915.html

代用特性を利用した再発防止策（品質特性による分類）の提案

- 知識フレームワークに従った再発防止策の蓄積と活用
  - ⇒ 各企業が実施している信頼性向上のための工夫事例を収集
  - ⇒ 工夫事例を知識フレームワークに従って整理
  - ⇒ ソフトウェア開発局面で活用

- 活用方法
  - 新たなソフトウェア開発または保守作業への利用
  - 定期障害予防診断の実施

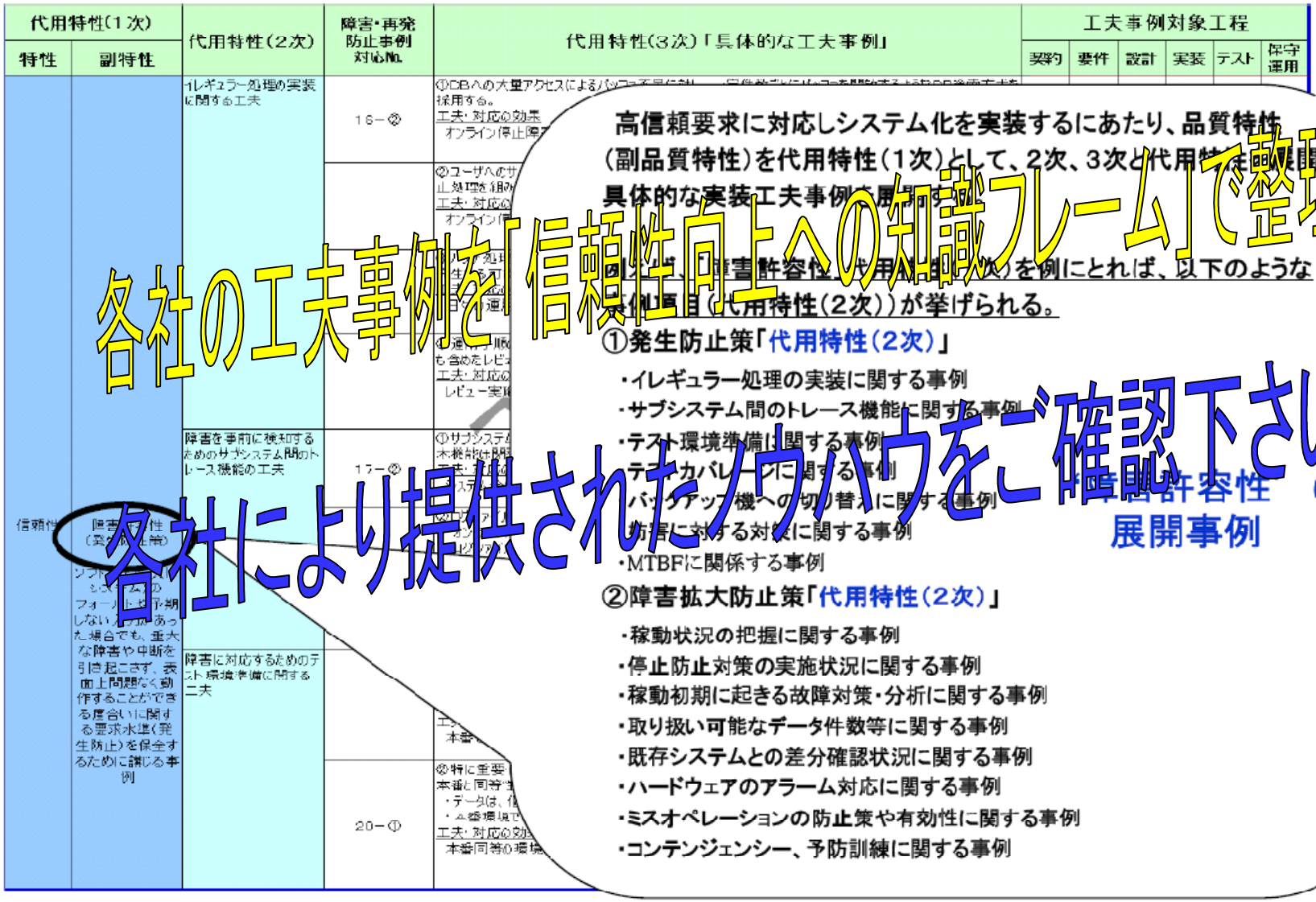
高信頼化のための手法WGの活動手順

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

- 国内企業のベストプラクティス事例を収集し、特徴を分析
  （14企業・団体　⇒　7企業）
  - 予防活動および検知活動に関わる各社事例紹介と討議
- 代用特性を利用した信頼性向上への工夫事例を収集
- エンタプライズ系代表企業「10社」のテスト実態分析(テストの観点の分析)
- 「高信頼化ソフトウェアのための開発手法ガイドブック」の作成

各社の工夫事例を再発防止策として整理

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



各社の工夫事例を「信頼性向上への知識フレーム」で整理

各社により提供されたノウハウをご確認下さい

（留）2010年3月時点（331事例；55件/品質特性） http://sec.ipa.go.jp/reports/20100915.html

代用特性を利用した障害再発防止事例

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

●障害影響度指標一覧表を用意しました。

階級	サービス停止時間	業務への影響	ユーザへの影響	監督官庁からの要求	経済的損失
1	数分(10分)未満でサービス停止となる	サービス停止業務の担当者は、遂行業務と併行して障害対応を行わなければならない			
2	10分以上1時間未満の範囲でサービス停止となる	サービス停止業務の担当者は、遂行業務を一時中断し、障害の復旧ならびにユーザ対応を行わなければならない	システム利用ユーザの5%未満に影響を与え、クレームが発生する		
3	1時間以上4時間(半日)未満の範囲でサービス停止となる	サービス停止業務の関連部門担当者は、遂行業務を中断し、障害の復旧ならびにユーザ対応を行わなければならない	システム利用ユーザの5%以上20%未満に影響を与え、賠償を伴うクレームが発生する	監督官庁より注意をうける	自社経常利益の1%未満の損失が発生する
4	半日以上サービス停止となる	サービス停止業務の関連部門の全担当者は、全ての遂行業務を中断し、障害復旧ならびにユーザ対応を行わなければならない	システム利用ユーザの20%以上40%未満に影響を与え、賠償を伴うクレームが発生する	監督官庁より勧告をうける	自社経常利益の1%以上3%未満の損失が発生する
5		全社レベルで遂行業務を全て中断し、障害復旧ならびにユーザ対応を行わなければならない	システム利用ユーザの40%以上に影響を与え、賠償を伴うクレームが発生する	監督官庁より業務改善命令以上の処罰をうける	自社経常利益の3%以上の損失が発生する

注)各社の状況に合わせて利用

代用特性を利用した障害再発防止事例

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

＜障害事例＞

過去、社会的に問題となったシステム障害から、出来る限り全ての品質特性（代用特性）を網羅するように39の障害事例を列挙

＜障害による影響度合い＞

システムの再発防止策などに関する水準を判定するために、システム障害の影響度合いを指標化

＜根本原因＞

障害事例から再発防止策を考える上で、根本原因を想定。（実際の根本原因は別にある可能性あり）

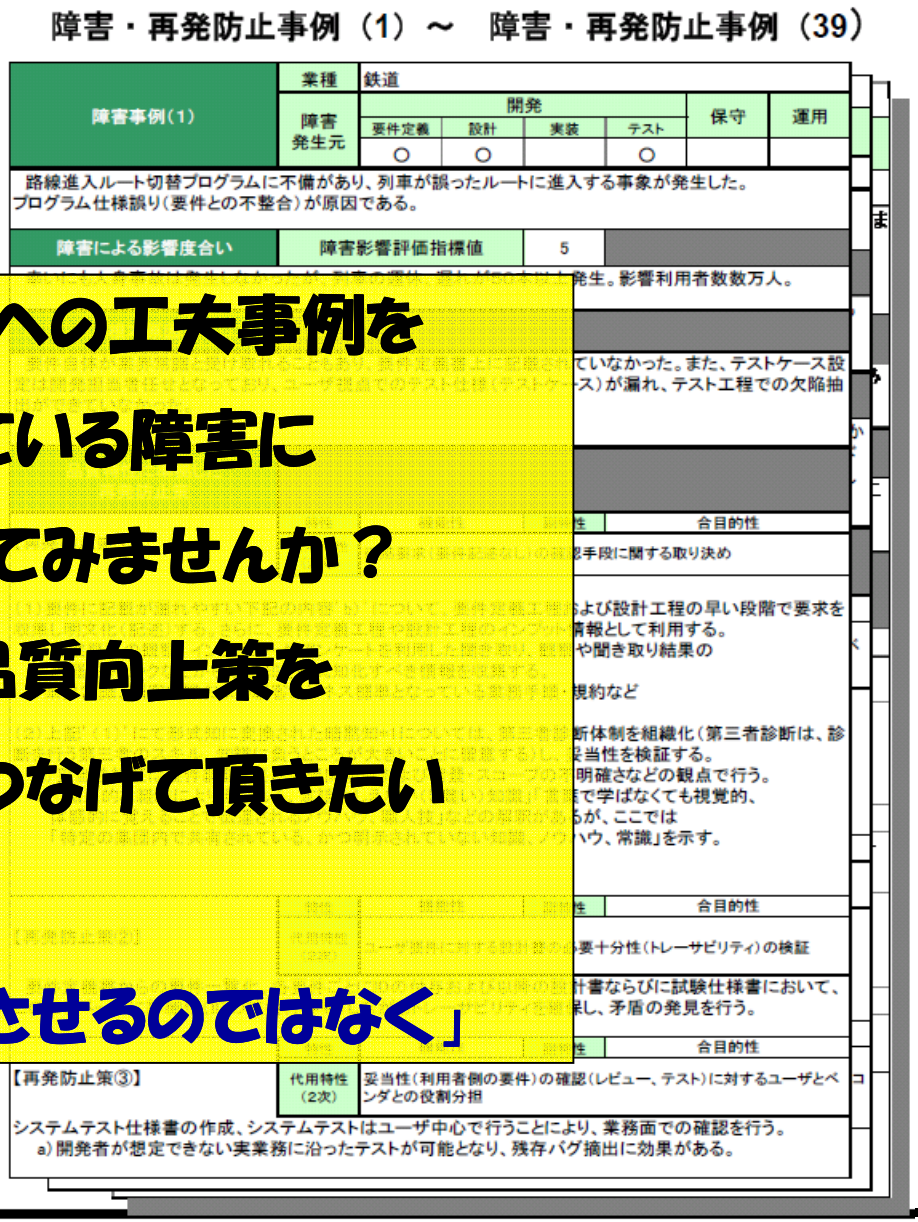
＜再発防止策＞

本WG各委員で作成した（代用特性を利用した）工夫事例‘33事例’）に基づいて再発防止策を事例化。（再発防止策が考えられるので、あくまでも参考事例）

様々な信頼性向上への工夫事例を
自社で収集している障害に
当てはめて整理してみませんか？
ソフトウェアの品質向上策を
障害再発防止策につなげて頂きたい

⇒

「闇雲に品質を向上させるのではなく」



代用特性を利用した障害再発防止事例

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

■ 品質特性ごとの予防活動の整理結果（機能性）

代用特性（1次）		代用特性（2次）	代用特性（最終）		
品質特性	品質副特性		事例		
機能性	合目的性	妥当性（利用者側の要件）の確認（レビュー、テスト）に対するユーザとベンダとの役割分担	1-③	2-①	3-①
			28-④	33-①	
		暗黙要求（要件記述なし）の確認手段に関する取り決め	1-①		
		仕様変更に対するユーザとベンダとの合意	4-①		
		ユーザ要件に対する設計書の必要十分性（トレーサビリティ）の検証	2-②	1-②	
	正確性	計算精度やデータ精度要求の実装確認	5-①		
		システム利用マニュアルの記述の正確性向上施策	6-①	7-③	
		検証（工程整合）確認に対するユーザとベンダとの合意	8-②		
		上位設計書に対する下位設計書（またはプログラム、テスト仕様）の必要十分性（トレーサビリティ）検証	9-①		
	相互運用性	外部システムとの接続要求および仕様の確認（レビュー、テスト）に対するユーザとベンダとの合意	5-②	10-①	11-①
		外部接続に対するデータ数、変換、編集（データ形式、コードなど）に関するコミュニケーション	11-②		
		相互接続する他のソフトウェアのバージョンアップによる影響に対する考慮	12-①		
		接続先外部システムとのコンテンジェンシープラン共有化	13-⑤		
	セキュリティ	データ暗号化対策	14-①		
		開発時でのデータ漏洩防止対策	14-②		
		セキュリティ事故発生に備えた、追跡可能性および監査容易性などの向上施策	14-③	15-①	
		セキュリティパッチなど脆弱性の予防対策			
		不正アクセス、不正ログインに備えたアクセス制御対策	16-①		
		なりすましのモニタリング、警告機能に関する工夫	16-②		
		データ損傷などのデータ保全対策	16-③		
		セキュリティ要件（実装検証含む）のユーザとベンダとの合意	15-②		
	機能性標準適合性	機能に対応する規定（業務、内部統制、ISMS、国際会計など）および開発標準の適合監査対策			

代用特性を利用した障害再発防止事例

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

■ 品質特性ごとの予防活動の整理結果（信頼性）

代用特性（1次）		代用特性（2次）	代用特性（最終）		
品質特性	品質副特性		事例		
信頼性	成熟性	潜在的障害予測（または潜在的障害低減）および障害解決状況の分析	17-①		
		仕様変更管理（傾向、内容分析）に基づく、仕様変更の制御	4-②		
		MTBF※3向上施策	18-①	30-③	
		「テストコスト」と「本番稼動後の欠陥による社会的影響や復旧にかかるコスト」とのトレードオフを考慮したテスト手法（テスト網羅率）の取り込み	2-③	8-①	35-③
	障害許容性（発生防止策）	イレギュラー処理の実装に関する工夫	18-②		
		障害を事前に検知するためのサブシステム間のトレース機能の工夫	17-②		
		障害に対応するためのテスト環境準備に関する工夫	19-①	20-①	37-③
		障害管理（傾向、内容分析）に基づく、障害の予防・是正処置の実施	21-③		
		運用時点での障害発生状況（傾向、内容分析）に基づく、障害の予防・是正処置の実施	21-②		
		ミスオペレーションの防止に関する工夫	13-①		
		システム資源（CPU、ディスクなど）の使用状況監視、警告に関する工夫	18-③	22-①	
	障害許容性（拡大防止策）	停止防止対策	20-②		
		稼働初期に起きる故障対策・分析に関する施策	23-②	24-①	
		取り扱い可能なデータ件数等システムのキャパシティオーバー時の誤動作防止に関する工夫	25-①	26-①	
		既存システム修正時における障害（リグレッションテスト漏れ、本番リリース不備による二次障害など）回避策	21-④		
		ハードウェアのアラーム対応に関する工夫	18-④		
		障害対応マニュアルに関する工夫			
		異常を検知し他システムなどへの影響を遮断する機能に関する工夫	27-①	28-①	
		予防訓練に関する施策	20-③		
	回復性	可用性（システムの稼働率、サービス提供時間、運転時間等の割合）向上施策	29-①	30-①	
		障害分類を配慮した障害回復時間短縮施策			
		サービス提供再開施策（復旧予測時間の配慮など）	11-③	21-①	33-③
		自動リカバリ機能など故障耐久能力向上施策	20-④		
		バックアップ機への切り替えに関する施策	18-⑤	31-①	
		広域・局所災害対策	30-②		
		要件定義時点におけるユーザとベンダ間のコンテンジェンシープラン共有化	19-②		
	信頼性標準適合性	信頼性に対応する規定（業務、内部統制、ISMS、国際会計など）および開発標準の適合に関する監査対策			

代用特性を利用した障害再発防止事例

■ 品質特性ごとの予防活動の整理結果（使用性）

代用特性（1次）		代用特性（2次）	代用特性（最終）		
品質特性	品質副特性		事例		
使用性	理解性	ユーザインタフェース（メニュー、アイコンなど）の工夫	13-②		
		利用者側の教育効果向上施策	6-②	23-③	
	習得性	ユーザレベル（初級、中級、上級）などを配慮した習得容易性向上施策	23-①		
		システム利用マニュアルおよびシステム運用マニュアルを判り易くするための工夫			
		ヘルプ機能の容易性などに関する工夫	13-③		
	運用性	利用者に対する操作のナビゲート、操作結果の確認のし易さに関する工夫	6-③	7-④	13-④
		システム運用の容易性向上、誤操作防止施策	28-②		
		インストールやバージョンアップの容易性（バージョンアップ後の確認の容易性、配布容易性を含む）向上施策	32-①		
		誤り修正（取り消し作業を含む）の容易性向上施策			
		誤操作からの回復性向上施策	7-①		
		利用時におけるメッセージのわかりやすさの向上施策	7-②		
		オペレータの介入操作に関する工夫	28-③		
	魅力性	少ないオペレーションで多くの処理を実現する工夫			
	使用性標準適合性	使用性に対応する規定（業務、内部統制、ISMS、国際会計など）および開発標準の適合に関する監査対策			

代用特性を利用した障害再発防止事例

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

■ 品質特性ごとの予防活動の整理結果（保守性）

代用特性（1次）		代用特性（2次）	代用特性（最終）		
品質特性	品質副特性		事例		
保守性	解析性	データログ実装、状況監視データ取得など活動記録保有能力に関する施策	33-②		
		診断機能実装および故障原因解析に関する工夫			
		ソフトウェアのトレース機能など解析容易性向上施策	17-③	29-②	
	変更性	変更履歴、構成管理などの変更制御に関する工夫	34-②		
		母体システムの構造化度、変更生産性など変更容易性向上に関する施策	35-②		
		保守ドキュメント、コメント率など変更容易性向上に関する施策	17-④		
	安定性	保守作業での欠陥混入是正に関する工夫	24-②		
		母体品質向上施策	17-⑤		
		システム保有データの整合性維持に関する施策	36-①		
		バージョンアップによるデグレード防止に関する施策	34-①	37-①	
		障害混入確率の低減に関する施策	35-①		
	試験性	本番リリースを保証するためのテスト範囲特定方法などに関する取り組み	9-②	21-⑤	22-②
		保守テストでのテストツール（自動再帰テストツールなど）、本番環境具備、標準テストセットの具備などのテスト環境整備に関する工夫	38-①		
		試験性に配慮したソフトウェアおよびデータの構造化、ならびに他システム接続方式に関する施策	5-③		
	保守性標準適合性	保守性に対応する規定（業務、内部統制、ISMS、国際会計など）および開発標準の適合に関する監査対策	39-①		

代用特性を利用した障害再発防止事例

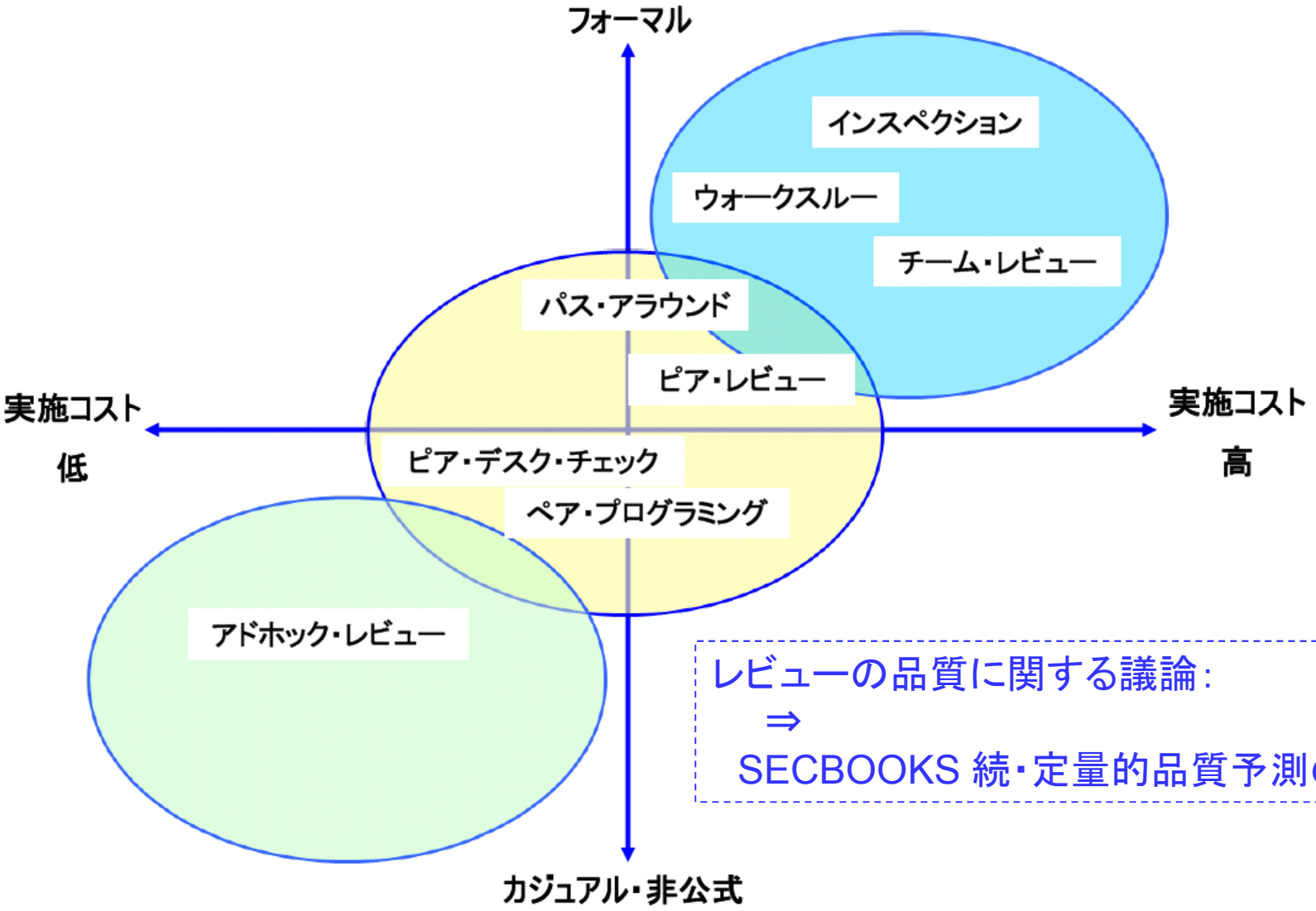
SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

■ 品質特性ごとの予防活動の整理結果（効率性、移植性）

代用特性（1次）		代用特性（2次）	代用特性（最終）		
品質特性	品質副特性		事例		
効率性	時間効率性	非平常時（ピーク時、障害および災害時など）に対応するレスポンスタイム、スループット、ターンアウンドタイムの考慮点などに関する施策	25-②		
		非平常時（ピーク時、障害および災害時など）に対応する業務のスループット、デリバリタイムの考慮点などに関する施策			
	資源効率性	非平常時(ピーク時,障害および災害時など)に対応する資源(CPU、メモリ、伝送、入出力、設置条件（スペース、電源容量など))、体制などの考慮点に関する施策	26-②		
	効率性標準適合性	効率性に対応する規定（業務、内部統制、ISMS、国際会計など）および開発標準の適合に関する監査対策			

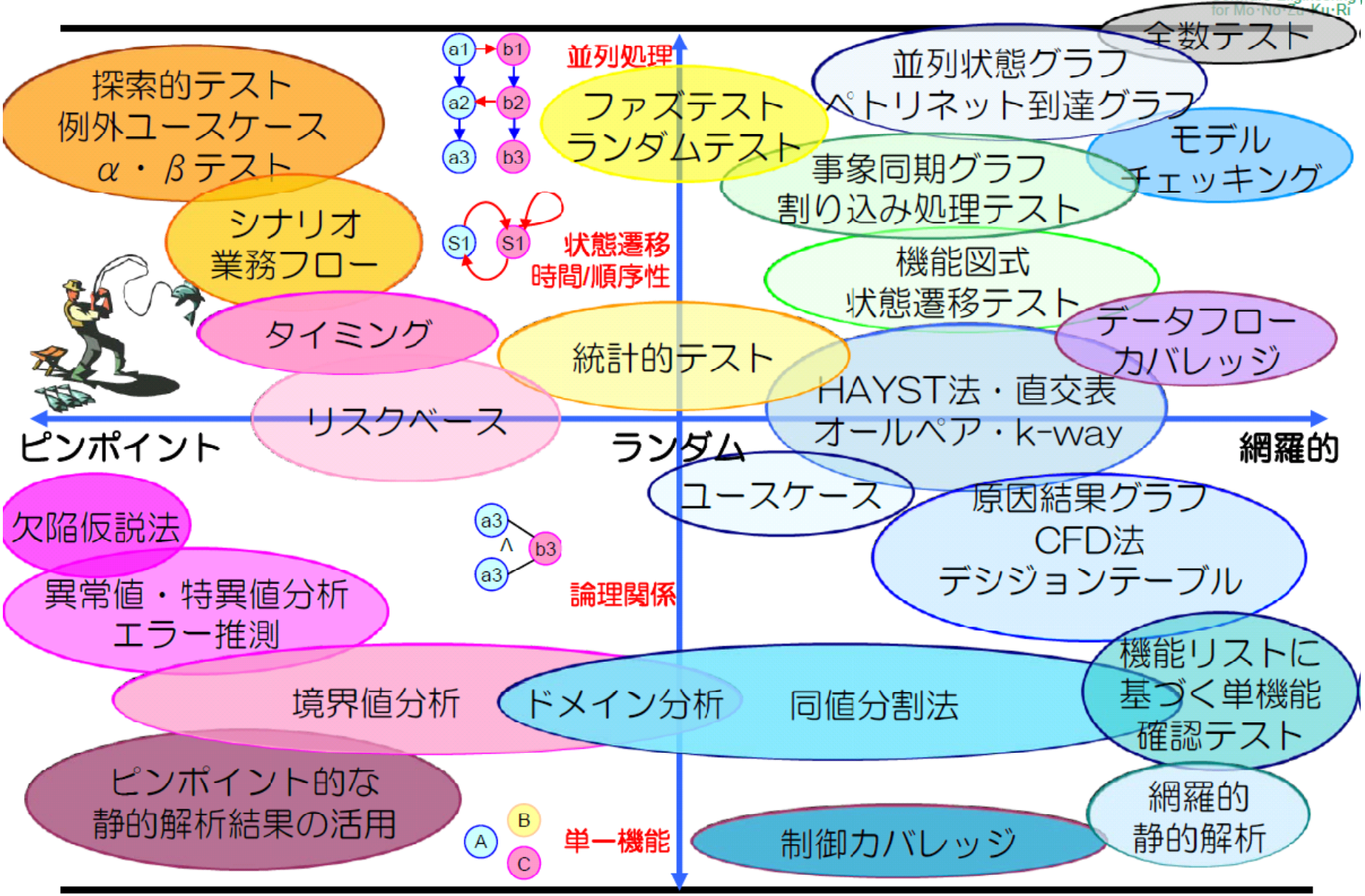
代用特性（1次）		代用特性（2次）	代用特性（最終）		
品質特性	品質副特性		事例		
移植性	環境適応性	ソフトウェア（アプリケーション）の環境適用性向上施策（OS、ミドルウェア、DBMSなど）	37-②		
	設置性	移植時の設置作業の確実性向上施策（移植作業支援ツール（移植箇所特定）など）			
	共存性	同一環境下（同一サーバ、同一DBMSインスタンスなど）で、複数アプリケーションを共存させるための施策			
	置換性	制度変更などシステム機能を置換しやすくするための施策（部品化など）			
	移植性標準適合性	移植性に対応する規定（業務、内部統制、ISMS、国際会計など）および開発標準の適合に関する監査対策			

検知活動：品質レビュー手法のポジショニングマップ



レビューの品質に関する議論：
⇒
SECBOOKS 続・定量的品質予測のススメ

検知活動：テスト技法ポジショニングマップ



検知活動:テスト網羅性の高度化技法

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

テスト要求分析手法

- お客様情報の収集(真の要求の把握を行うために)
- 開発プロジェクト情報の収集

  「ここに障害があるとシステム全体が止まってしまう」といったシステムの急所情報等
- これらをテストベースとして構成管理
- テストベースを参照しながら、以下のような手法を使ってテスト全体を明確化

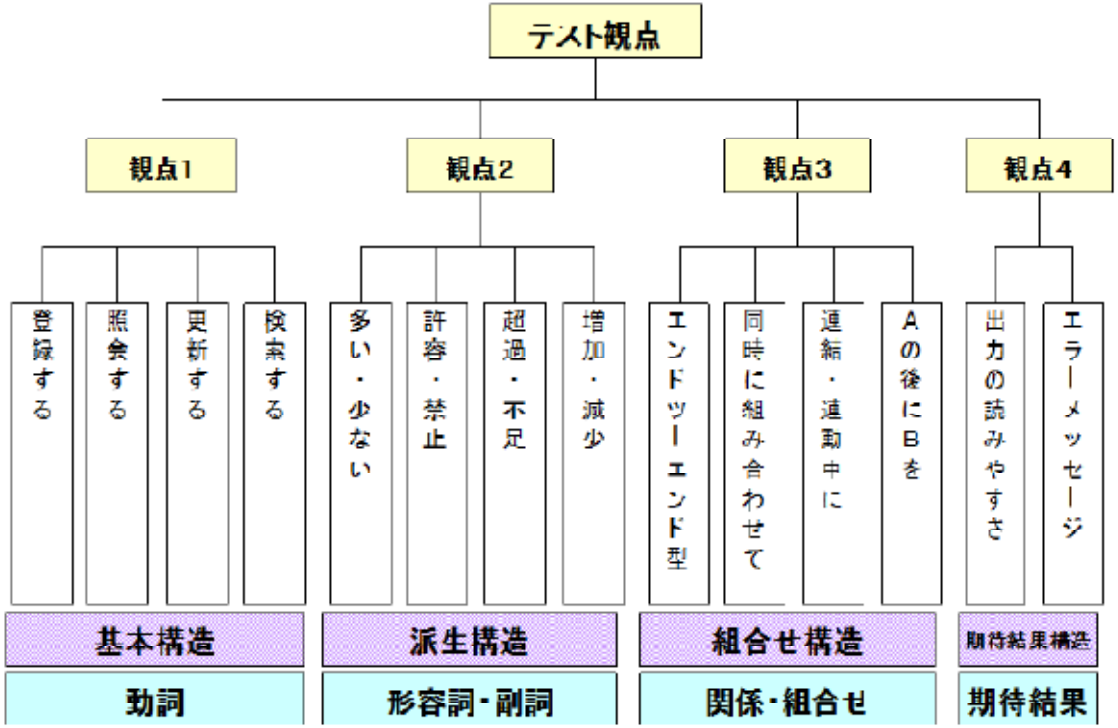
手法名	表現方法	特徴	プロセス
NGT	ツリー	テスト全体をテスト観点で網羅	VSTeP
FV 表	表形式	目的機能の切り口で V&V を網羅	HAYST 法
ゆもつよメソッド	表形式	機能×テストタイプで網羅をチェック	ゆも豆
Tiramis	ツリー (MM)	テストカテゴリ分析を実施。機能は MM	—
TAME	ツリー (MM)	テスト設計思考の可視化とレビューによるテスト観点の洗い出しおよび整理	—

検知活動：テスト網羅性の高度化技法

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

■ テスト観点とテストアーキテクチャ設計

エンタプライズ系代表企業10社のテストの実態分析よりテストの観点を整理した。



● テスト観点を上手に組み合わせるパターン（文法構造）を構造化しておくと、文章を作るようにテスト条件を網羅的に作成できる。

（文法構造の例）
観点１；基本構造
テストタイプ；「入力網羅テスト」
文法構造；｛テスト対象｝に（目的語）を｛テスト観点１｝させる

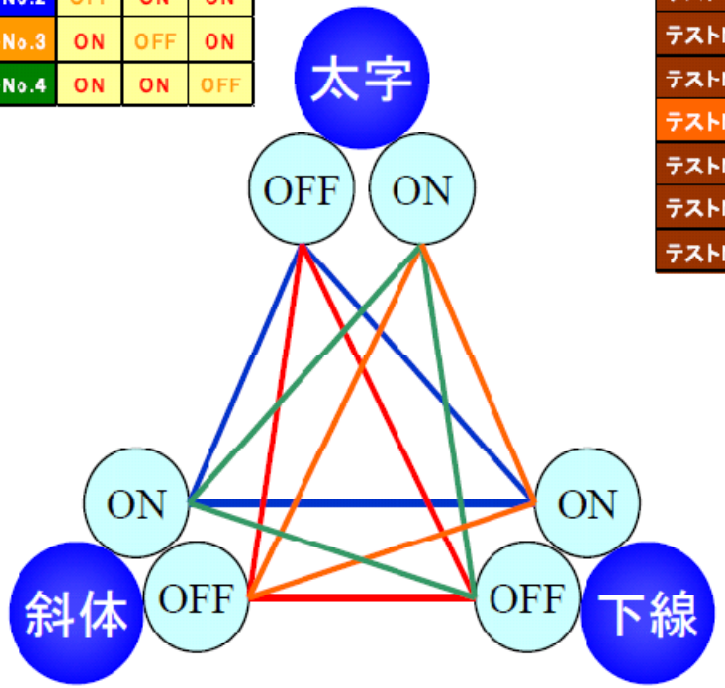
●アーキテクチャ設計とは、この文法構造を用いた組み合わせ方（‘基本構造’に‘派生構造’を追加など）を指す。

検知活動：テスト網羅性の高度化技法（直交表）

■ 直交表を活用した網羅的な組合わせテスト

（直交表が網羅する組合せ）

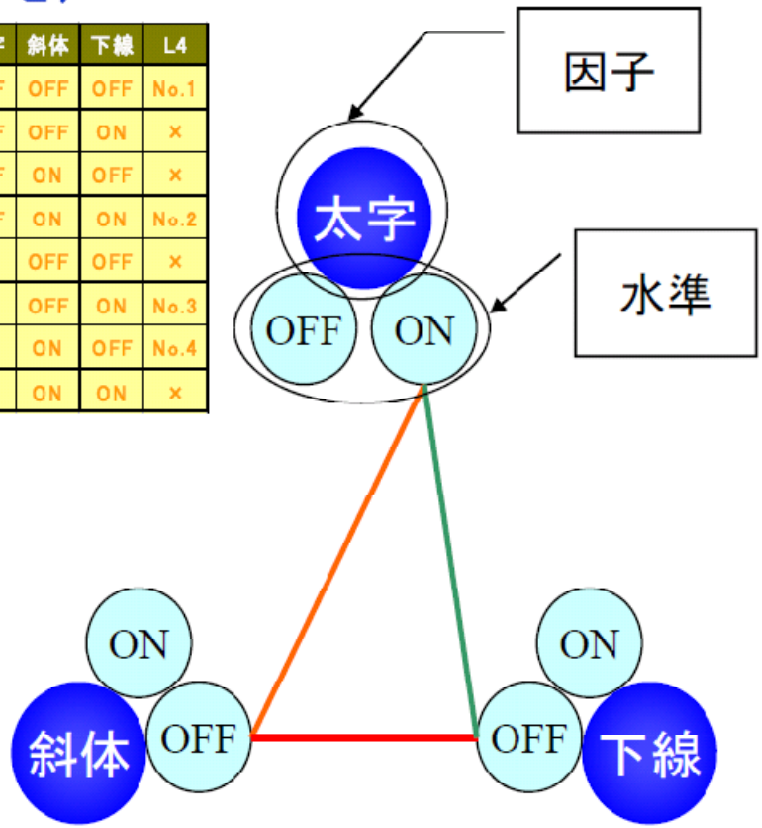
L4直交表	太字	斜体	下線
テストNo.1	OFF	OFF	OFF
テストNo.2	OFF	ON	ON
テストNo.3	ON	OFF	ON
テストNo.4	ON	ON	OFF



2機能間の組合せが全て存在している
（この状態を組合せ網羅率100%と定義する）

（全組合せ）

全組合せ	太字	斜体	下線	L4
テストNo.1	OFF	OFF	OFF	No.1
テストNo.2	OFF	OFF	ON	×
テストNo.3	OFF	ON	OFF	×
テストNo.4	OFF	ON	ON	No.2
テストNo.5	ON	OFF	OFF	×
テストNo.6	ON	OFF	ON	No.3
テストNo.7	ON	ON	OFF	No.4
テストNo.8	ON	ON	ON	×



全組合せでは直交表に存在しない組合せも出現する
よく見ると直交表では現れないNo2, 3, 5は単機能テスト内容

（出典）「ソフトウェアテストにおける直交表の活用」（富士ゼロックス株式会社　秋山浩一）

検知活動：高度化技法（ピンポイントテスト）

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

■ シナリオを用いた効果的なピンポイントのテスト

ピンポイントな技法とは、リスクの高い部分にピンポイントでテストを実施するための方法

●業務フローの観点によるシナリオテスト

業務フロー定義書を基に、テスト設計でテスト技術者が不足分を補い、それを顧客と　合意することで、業務フローの観点によるテストシナリオを作成。

●業務周期の観点によるシナリオテスト

業務周期「週次、曜日、月次、期末、年末」等々、サイクルごとに業務をリストアップし、それを基に業務フローを運用サイクル化することで、テストシナリオを作成。

●業務データの観点によるシナリオテスト

データの変化に着目したシナリオテストで、これをエンティティ・ライフヒストリーに基づくシナリオテストと呼ぶ。顧客の状態や口座の状態に着目したりする。データの以外に人・物・金・承認の流れを盛り込むことで、より現実的なシナリオを作成。

高信頼化のための手法WGの活動手順

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

- 国内企業のベストプラクティス事例を収集し、特徴を分析
  （14企業・団体　⇒　7企業）
  - 予防活動および検知活動に関わる各社事例紹介と討議
- 代用特性を利用した信頼性向上への工夫事例を収集
- エンタプライズ系代表企業「10社」のテスト実態分析(テストの観点の分析)
- 「高信頼化ソフトウェアのための開発手法ガイドブック」の作成

【参考】エンタプライズ系代表的企業10社のテスト実態分析

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

SLCP で定めるテスト工程		ユーザ系			独立系	独立系	ベンダ系				
		A社	B社	C社	D社	E社	F社	G社	H社	I社	J社
1.6.7	ソフトウェアコード作成およびテスト	単体テスト	コードカバレッジ確認 画面構成確認 画面操作確認 画面処理内容確認 DB 更新確認 出力項目確認 実行時間確認 大量データ性能確認	プログラム機能テスト 機能テスト 異常値テスト ゼロ件テスト 少量データテスト 大量データテスト	カバレッジテスト DB 単体テスト 入力データバリエーションテスト 出力データバリエーションテスト エラーテスト 操作確認テスト 入力チェックテスト 入力バリエーションテスト 遷移バリエーションテスト 照合バリエーションテスト 検索バリエーションテスト	カバレッジテスト フラグ確認テスト データマトリクステスト データバリエーションテスト DB アクセステスト エラーテスト レイアウト確認テスト 入力チェックテスト 画面機能テスト 単画面性能テスト	機能単体ホワイトボックステスト 機能単体ブラックボックステスト	命令網羅試験 分岐網羅試験 機能試験 画面操作試験	商品テスト モジュールテスト 表示仕様確認 入力チェック確認 基本機能確認 出力項目確認	データアクセス層単体テスト ビジネスロジック層単体テスト プレゼンテーション層単体テスト バッチ単体テスト	JOBSTEP テスト JOE 機能テスト 画面機能テスト 性能テスト 標準準拠確認テスト
1.6.8 1.6.9	ソフトウェア結合 ソフトウェア適合性確認テスト	サブシステム内結合テスト サブシステム間結合テスト	画面遷移処理バリエーション確認 バッチ・プロセス間連携動作確認 業務・サブ業務機能間テスト 総合運用管理システム連携テスト 機能テスト 運用機能運転テスト バッチ・リラン・リカバリ方法確認 性能テスト 特殊な性能テスト 外部システム連携テスト	オンライン画面遷移テスト バッチ入出力確認テスト Web アプリ脆弱性診断	画面遷移テスト 排他制御確認テスト 操作組合せテスト JOB 正常系テスト JOB 異常系テスト 外部システム連携テスト セキュリティテスト リグレッションテスト	画面遷移機能テスト 画面遷移網羅テスト 画面遷移性能テスト JOB 正常系テスト JOB 異常系テスト サブシステム内結合テスト サブシステム間結合テスト	機能連携テスト サブシステム連携テスト	並行／同時処理試験 競合性試験 プログラム間 I/F 試験 機能間 機能試験 機能／ロック間機能試験 機能試験 障害回復試験 障害発生時稼動確認 エラーメッセージ動作確認 平均性能試験 限界性能試験 容量試験 インストール試験	サブシステム内結合テスト サブシステム間結合テスト 業務機能テスト システム機能異常テスト 性能テスト 限界テスト	モジュール連動テスト JOB フローテスト ユースケースシナリオテスト 性能テスト 接続テスト セキュリティテスト 業務フローテスト 原新確認テスト	チーム内結合テスト チーム間結合テスト 排他テスト 方式設計確認テスト 画面負荷テスト ボリュームデータテスト 外部インタフェーステスト
1.6.10 1.6.11	システム結合 システム適格性確認テスト	サイクルテスト 導入性テスト 操作文書テスト 業務処理文書テスト 運用文書テスト 運用テスト プロシジャーテスト リカバリーテスト 性能テスト ストレステスト 大容量テスト ストレージテスト スケーラビリティテスト セキュリティテスト ファシリティテスト	災害対策テスト サービスレベルテスト 性能テスト 負荷テスト 機能テスト セキュリティテスト	JCL 稼働確認テスト シナリオテスト プレ本番テスト オーナーテスト 有償全体テスト 有償月例テスト 性能テスト 負荷テスト 機能テスト（条件組合せ） セキュリティテスト 総通テスト リグレッションテスト	通常業務シナリオテスト 例外業務シナリオテスト 本番データテスト データ移行リハーサル リカバリテスト パフォーマンス測定 資源測定 セキュリティテスト	運用サイクルテスト 主要業務シナリオテスト 詳細業務シナリオテスト 例外業務シナリオテスト 本番データテスト 移行データ確認テスト 移行シナリオテスト 操作マニュアルテスト フィールドテスト 運用手順テスト 運用監視テスト 障害発生テスト 障害対応テスト バックアップテスト リカバリーテスト 性能要件テスト 負荷時／ラッシュテスト ルーノット検証テスト 可用度テスト ロングランテスト サイジングテスト スケーラビリティテスト ボリュームテスト リソース不足テスト ユーザビリティテスト セキュリティテスト 総通確認テスト 外部接続テスト デグレード確認テスト 保守マニュアルテスト 構成テスト データ互換性テスト	業務運用テスト 導入テスト インフラ運用テスト リカバリーテスト パフォーマンステスト 負荷テスト ストレステスト ロングランテスト スケーラビリティテスト ボリュームテスト 機能性テスト ユーザビリティテスト セキュリティテスト 構成テスト	運用繰り返し試験 業務フロー機能試験 システム起動停止試験 休日運転試験 障害発生時稼動確認試験 縮退運転試験 エラーメッセージ対応動作確認試験 障害回復試験 耐久試験 連続運転安定性試験 他システム I/F 試験 並行／同時処理試験	業務サイクルテスト 業務機能テスト 特殊処理フロー確認 可用性テスト システムリカバリテスト 性能測定 システム負荷テスト 使用性テスト セキュリティテスト 保守性テスト	業務系ジョブネット連動テスト 運用系ジョブネット連動テスト 業務・運用 ジョブネット連動テスト 日回しテスト 移行プログラム連動テスト エンドユーザ参加テスト コンティンジェンシー観点テスト 運用確認テスト 運用確認（異例運用） 運用確認（監視端末） 業務系ジョブネット障害テスト 障害時リラン 信頼性テスト 性能テスト 性能テスト（高負荷） セキュリティテスト 外部センタ接続テスト システム切替 並行運用	運用サイクルテスト 正常系業務テスト 異常系業務テスト 非通常運用テスト バッチ性能テスト 画面負荷テスト 長時間運用テスト ボリュームデータテスト 保守テスト
1.6.13	ソフトウェア受入れ支援	導入性テスト 受入テスト 運用テスト		運用テスト	運用テスト			運用試験	運用テスト	運用テスト	

【参考】エンタプライズ系代表的企業10社のテスト実態分析

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

■エンタプライズ系企業10社のテストの観点を分析した。
■縦軸に品質特性、横軸にハイレベルな観点を配置できることがわかった。
　⇒　全部が埋まるという意味ではない。
■各社は名前は異なるが同一と分類できるテストを実施しているようだ。
■一方で、各社の重要視しているテストと品質によって、以下の表の埋まり具合の分布が異なることがわかった。
　⇒　対象ソフトウェアやソフトウェア開発の異なった状況に呼応したテスト戦略によって該当する項目が変わると解釈できる。

		開発					移行		業務		
		機能					移行・導入	受け入れ・本番	業務	保守	運用
			データ	状態	操作	コード					
機能性	機能性										
	相互運用性										
	セキュリティ										
信頼性	障害許容性										
	回復性										
使用性	使用性										
効率性	時間効率性										
	資源効率性										
保守性	変更性										
移植性	共存性										

【参考】エンタプライズ系代表的企業10社のテスト実態分析

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

■自社テスト標準への活用 （例： テストタイプ分類）

		開発					移行		業務		
		機能					移行・導入	受け入れ・本番	業務	保守	運用
			データ	状態	操作	コード					
機能性	機能性	2			2	1	1	1	1		2
	相互運用性										
	セキュリティ	1									
信頼性	障害許容性										
	回復性										1
使用性	使用性										
効率性	時間効率性	2	1								
	資源効率性		1								1
保守性	変更性										
移植性	共存性										

■プロジェクトへの反映 （例： 重視する観点で縮退）

		開発			移行		業務		
		機能			移行・導入	受け入れ・本番	業務	保守	運用
			データ	コード					
機能性	機能性								
	相互運用性								
	セキュリティ								
信頼性	障害許容性								
	回復性								
効率性	時間効率性								
	資源効率性								
保守性	変更性								

高信頼化のための手法WGの活動手順

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

- 国内企業のベストプラクティス事例を収集し、特徴を分析
  （14企業・団体　⇒　7企業）
  - 予防活動および検知活動に関わる各社事例紹介と討議
- 代用特性を利用した信頼性向上への工夫事例を収集
- エンタプライズ系代表企業「10社」のテスト実態分析(テストの観点の分析)
- 「高信頼化ソフトウェアのための開発手法ガイドブック」の作成

● 東京海上日動システムズ株式会社
（リスク管理の取組み）

高品質なシステム開発のためには、システムの重要度にあわせたリスク評価・対応と、上流工程での要件の精度の向上や設計書・プログラムミスの削減が必要である。

開発するシステムをシステム運用の観点から評価する、「システム重要度に基づいたリスク管理」（システムプロファイル）の取り組みを紹介する。

また、設計・製造工程の品質向上の取り組みとして、上流工程での要件精度向上の取り組みと、設計・プログラミングの開発局面の品質向上を目的とした取り組みについて紹介する。

● 富士ゼロックス株式会社
（ソフトウェアテストの改善から上流に向かった取組み）

ソフトウェアの品質保証の課題と対応について、下流工程であるソフトウェアテストの改善から上流に向かっていった事例である。一般に下流工程での問題点は、お客様に直結するため明白であり、また下流工程の改善は、その対策効果を把握しやすいという特徴がある。

富士ゼロックスでは、HAYST法と呼ばれる直交表の技術を利用したテスト技法を核として改善を進めている。組込み系の事例ではあるが、エンタプライズ系においても参考になるだろう。

● 日本ユニシス株式会社
（第三者品質保証体制での活動）

日本ユニシスでは、かねてより品質保証を支える重要技術として、プロセス管理とともにテスト技術を重視し、数年前からインドのテスト専門会社であるSTAG Software Private Limited社（以下STAG社）の技術を導入し、重要プロジェクトに適用してきた。

その結果を踏まえ、品質保証部門内にテスト技術を用いたソフトウェア検査を専門とする部隊を設置し、プロセスとプロダクトの両面からシステムの開発行為を組織として保証する体制が整った。

本稿では、今回強化したプロダクト面での第三者品質保証体制での活動について紹介する。

● 株式会社日立製作所
（高信頼なシステム開発のための公式レビューをプロモートする専任部署設置）

要求されるシステムの信頼度に応じて、システムの品質ランクを5段階に設定している。特に高信頼性を要求される社会インフラシステムについては、通常の開発プロセスで実施するレビューに加え、要件定義から運用・保守の各工程で厳格な公式レビューを実施することにした。

上記の公式レビューをプロモートするために、開発プロジェクトとは独立した専任部署を設置した。

専任部署を設置したことにより、レビュー精度の向上、レビュー指摘事項のトレーサビリティ確保、レビューノウハウの蓄積と共有が促進され、重大障害軽減に寄与することができた。

● 富士通株式会社
（上流工程での課題解決： 要件定義の妥当性確認監査など）

富士通では、システム開発の失敗原因の多くを占める上流工程で発生する課題・問題に着目した。要件定義の妥当性を確認するための監査と、外部設計ドキュメントの完成度を診断する施策の内容と実施状況を紹介する。

これらを実施した結果、プロジェクトのQCDに寄与する成果が得られており、機能性の品質確保を通して、使用性、効率性にも良い影響を与え、高信頼システムの実現に役立っている。

● 三菱電機株式会社
（非機能要件のトレーサビリティ管理手法）

三菱電機では、最近の重大障害の発生傾向から外部調達のISV/IHV製品を起因とする障害が増加傾向にあることに着目し、その発生を予防するために非機能要件のトレーサビリティ管理手法への取り組みを試行している。

非機能要求を具体的に開発し、確認するための作業手順をプロセスとして示すとともに、非機能要求管理の成果物として作成する非機能要求トレーサビリティマトリクスについて、その形式と利用方法を紹介する。

結果として、役割の明確化、作業項目や試験項目の具体化、見える化に寄与していることが確認できており、高信頼システム構築に役立っている。

ISV: Independent Software Vendor, IHV: Independent Hardware Vendor

● 株式会社ジャステック
（徹底した定量的指標に基づく取組み）

ジャステックでは、独自のソフトウェア生産管理方式「ACTUM」およびCMMIレベル5（V1.1およびV1.2ともレベル5達成）に基づき、高信頼ソフトウェアへの取り組みを行っている。

ここでの取り組み事例では、「ACTUM（環境変数）」を使用し、システムプロファイルごとの要求品質と開発コストとの関係、および出荷後の残存欠陥と開発コストとの関係について、弊社の統計分析および経験則に基づいて紹介する。

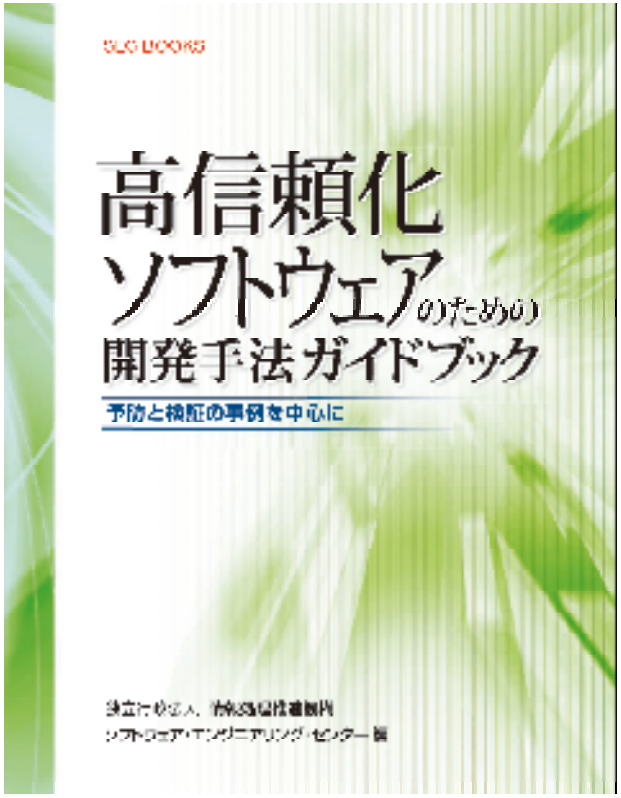
さらには「品質機能展開」を使い、要求機能および品質特性「代用特性」ごとの重要度、ならびに要求機能ごとの開発コスト（環境変数）をとらえ、開発予算に見合った機能および品質を調整する方法を紹介する。

まとめ

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

■「高信頼化ソフトウェアのための開発手法ガイドブック」のご紹介

- 高信頼システムとソフトウェア高品質化
- 高信頼化が期待される新しい情報システム
- 予防活動と検知活動による高信頼化
- 各社の信頼性向上への取組事例



ご清聴ありがとうございました。