■JaSST'08 Kyushu

現場目線で考える
テスト技法およびテスト充分性

2008年11月7日（金）

品質本部　秋山 浩一

FUJI xerox

自己紹介＆関連イベント

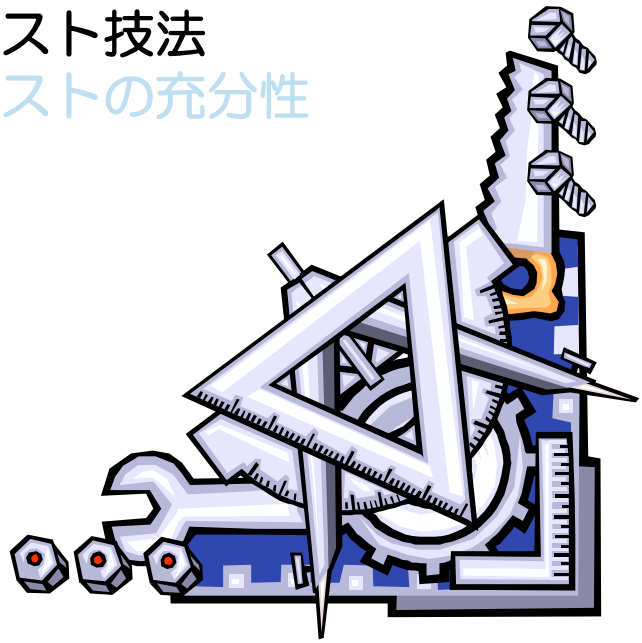





1985年4月1日：	富士ゼロックス（株）入社（Smalltalk, UNIX WS, Net.）
1996年11月21日：	ソフトウェアテスト技法＆ツールを作る仕事に従事
2000年5月30日：	Webで見つけたTEFへ参加（http://www.swtest.jp/）
2000年9月25日～9日：	2WCSQ：田口氏の発表の後に西さんの発表を聞く
2000年10月1日：	TCS翻訳プロジェクト始動
～2001年11月26日：	『基本から学ぶソフトウェアテスト』出版（共訳）
2002年5月8日：	LLST翻訳プロジェクト始動
～2003年4月21日：	『ソフトウェアテスト293の鉄則』出版（共訳）
2003年3月6日：	JaSST'03（第一回）（JaSST実行委員）
2004年1月27日～28日：	JaSST'04「HAYST法」発表
2005年1月5日：	『ソフトウェア・テストPRESS Vol. 2』出版（特集２）
2005年12月12日：	NPO法人　ASTER設立
2006年1月31日：	JTCB（現在のJSTQB）FL開始（ステアリング委員）
2007年7月31日：	『ソフトウェアテストHAYST法入門』出版（共著）
2008年5月10日：	『ソフトウェアテスト入門』出版（共著）
2008年7月15日：	『ソフトウェアテストの基礎』出版（共訳）
2008年10月7日：	日経品質管理文献賞受賞
	SQiPテストWG、SEA: SS2008テストWG、IPA/SEC委員

FUJI xerox

Ⅰ. ソフトウェアテスト技法
Ⅱ. ソフトウェアテストの充分性



ソフトウェアテストのカバーする範囲

ソフトウェアテストの一般的なイメージ

1. 開発者が自分が作ったものを誰かに渡す前に動作確認する作業
2. ソフトウェアテスト部門が実施するバグ出し
3. xUnitなどによるユニットテストフレームワークを使うこと
4. 自動ツールによる負荷テスト
5. リリース前のβ版をWebで配布して有志（勇者？）が使用

これらはソフトウェアテストの一部に過ぎません

ソフトウェアテストで考慮する領域は大まかに言って3つ

開発	テスト対象システム	お客様

FUJI xerox

それぞれは、さらに3つ

開発	システム	お客様
要件	入力	未来
仕様	人	現在
設計	データ	過去

FUJI xerox

それぞれは、さらに3つ

開発	システム	お客様
◆ 要件（Why?） 必要性、正当性、実現性、完全性、曖昧さ、優先順位、検証可能性	◆ 入力（Fuzz） 因子（6W2H）、ノイズ、水準（defact, dejure）、fuzzing、 有則、無則、禁則	◆ 未来（Monopoly） 負荷、保守性、アクティブノイズ（セキュリティ）、業界標準化
◆ 仕様（What?） 集合、論理（事前・事後・不変）、状態（変数の組合せ）、例外	◆ 人（Man,Method） プロセス、教育、経験、チームビルディング、メトリクス（複雑度）、組織体制、変化（3H）	◆ 現在（New Target） シナリオ（Where? When? Who? Whom?）と非機能要件
◆ 設計（How?） コアと変動要素、フレームワーク、クラス、変数の型	◆ データ（Machine） CRUD、大きさ、量、衝突、ロック、重複、0（NULL）、異常処理	◆ 過去（Customer） 販売数、売上げ、利益、バグ分析、VOC（Voice of Customer）

FUJI xerox

対策カテゴリ

開発	システム	お客様
◆ 要件（Why?） 必要性、正当性、実現性、完全性、曖昧さ、優先順位、検証可能性	◆ 入力（Fuzz） 因子（6W2H）、ノイズ、水準（defact, dejure）、fuzzing、 有則、無則、禁則	◆ 未来（Monopoly） 負荷、保守性、アクティブノイズ（セキュリティ）、業界標準化
◆ 仕様（What?） 集合、論理（事前・事後・不変）、状態（変数の組合せ）、例外	◆ 人（Man,Method） プロセス、教育、経験、メトリクス（複雑度）、組織体制、変化（3H）	◆ 現在（New Target） シナリオ（Where? When? Who? Whom?）と非機能要件
◆ 設計（How?） コアと変動要素、フレームワーク、クラス、変数の型	◆ データ（Machine） CRUD、大きさ、量、衝突、ロック、重複、0（NULL）、異常処理	◆ 過去（Customer） 販売数、売上げ、利益、バグ分析、VOC（Voice of Customer）

テスト技法

SPI

Software Product Line

Model Checking

複数の業務DB連携

テスト技法ポジショニングマップ　V1.0

アドホック
テスト
モンキー
テスト

顧客要求

Black Box

探索的テスト
例外ユースケース
α・βテスト

ユースケース

全数テスト
ファズテスト
ランダムテスト

シナリオ
業務フロー

要件

機能リストに
基づく単機能
確認テスト

HAYST法・直交表
オールペア・k-way

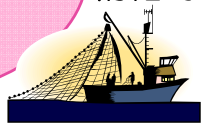
リスクベース

統計的テスト

同値分割
境界値分析
デシジョンテーブル
原因結果グラフ
CFD法

ピンポイント

網羅的



静的解析結果の活用
による
ピンポイントテスト

欠陥仮説法

仕様

状態遷移テスト

異常値・特異値分析
エラー推測

タイミングテスト

カバレッジ
モデルチェッキング

設計

White Box

FUJI xerox

テスト技法ポジショニングマップ V2.0γ

探索的テスト
例外ユースケース
α・βテスト

a1 → b1
a2 ← b2
a3 b3

並列処理

全数テスト

並行状態グラフ
ペトリネット到達グラフ

ファズテスト
ランダムテスト

事象同期グラフ
割り込み処理テスト

モデル
チェッキング

シナリオ
業務フロー

S1 ⇄ S2

状態遷移
時間/順序性

機能図式
状態遷移テスト

タイミング

統計的テスト

データフロー
カバレッジ

HAYST法

ピンポイント

リスクベース

ランダム

オールペア・k-way

網羅的

ユースケース

欠陥仮説法

C1 ∧ C2 → E1

論理関係

CFD法
原因結果グラフ
デシジョンテーブル

異常値・特異値分析
エラー推測

機能リストに
基づく単機能
確認テスト

境界値分析

ドメイン分析

同値分割法

ピンポイント的な
静的解析結果の活用

A
B
C

単一機能

制御カバレッジ

網羅的
静的解析

http://www.swtest.jp/wiki/index.php?TEF%2F%A5%C6%A5%B9%A5%C8%B5%BB%CB%A1

同値分割・境界値分析・デシジョンテーブル

同値分割法

■仕様から判断して、プログラム中で同一の手順により処理されると思われるデータ領域を同値クラスと呼ぶ。各同値クラスから代表値を取り出してテストデータとする方法

境界値分析

■同値クラスの中から代表を選ぶときに「端」、つまり、境界の値を選ぶ方法

デシジョンテーブル

■プログラムロジックを条件と動作に分け、マトリクスで表現する方法

FUJI xerox

同値分割法（EP）

equivalence class（同値クラス）：

equivalence partition を参照のこと。

equivalence partition（同値分割）：

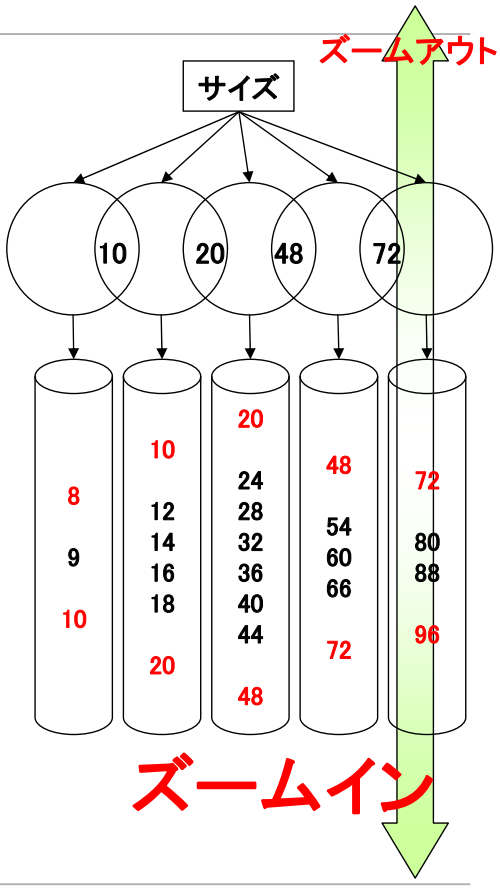
仕様上、コンポーネントやシステムの動作が同じと見なせる入力定義域や出力定義域の部分。

equivalence partitioning（同値分割法）：

ブラックボックステスト設計技法の１つ。同値領域から代表値を実行するテストケースを設計する。最低1 回各同値領域を実行するように設計するのが原則。テストに使う入力値が、同様の結果をもたらす場合、その入力値を「同値」と呼び、同値の取りうる範囲を「同値クラス」と呼ぶ。

ex)　文字フォントサイズ

- 文字フォントサイズの選択肢には「8, 9, 10, 12, 14, 16, 18, 20, 24, 28, 32, 36, 40, 44, 48, 54, 60, 66, 72, 80, 88, 96」の22種類ある。サイズの増え方に着目すると、「８~10」、「10~20」、「20~48」、「48~72」、「72~96」の５グループに同値分割できる。
- ここから「10, 20, 48, 72」の4つを代表に選び小さい因子にすることが出来る。
- 同値クラスの抜けは、もっとも防ぎにくい
  ⇒　同値クラスの発見は慎重にしよう！



FUJI xerox

境界値分析（BVA）

boundary value（境界値）：

同値分割した領域の端、あるいは端のどちらか側で最小の増加的距離にある入力値または出力値。例えばある範囲の最小値または最大値。

boundary value analysis（境界値分析）：

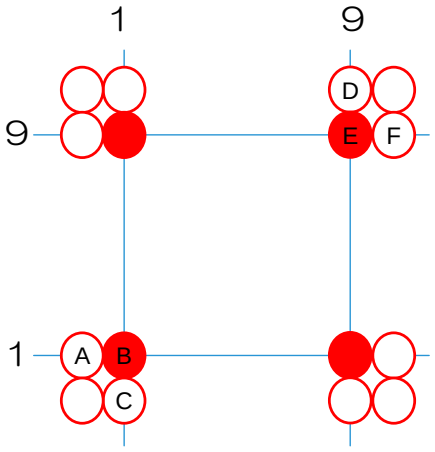
境界値を基にしてテストケースが設計される、ブラックボックステスト設計技法。

boundary value testing（境界値テスト）：

boundary value analysis を参照のこと。

ex) 1けたの正の数2つを足し算するプログラム。

- 「1+1」と「9+9」
- 「0+0」と「10+10」
- もう少し考えると、
  「0+0」、「0+1」、「1+0」、「1+1」、
  「0+9」、「0+10」、「1+9」、「1+10」、
  「9+0」、「9+1」、「10+0」、「10+1」、
  「9+9」、「9+10」、「10+9」、「10+10」
- さらに考えると、
  「0+1」、「1+1」、「1+0」、
  「9+10」、「9+9」、「10+9」



FUJI xerox

演習： 同値分割法、境界値分析法

BMI値は、体重(kg)／身長(m)2で計算され、次の判定表に従い肥満度を測るものである。
肥満のテストのための境界値データは何か？

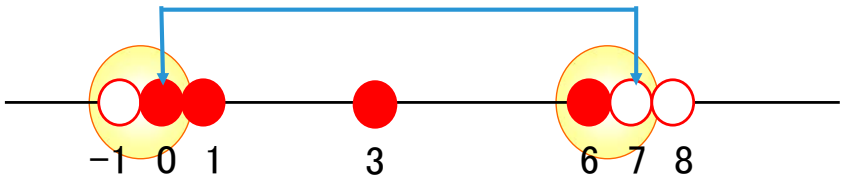
18.5未満	やせ
18.5～25未満	標準
25～30未満	肥満
30以上	高度肥満

FUJI xerox

もう一つの境界値分析

問題：　以下のモジュールのテストを設計しなさい

- int型の引数aがある
- モジュール内にはif (0<=a) と if (a<7)という２つの条件文がある

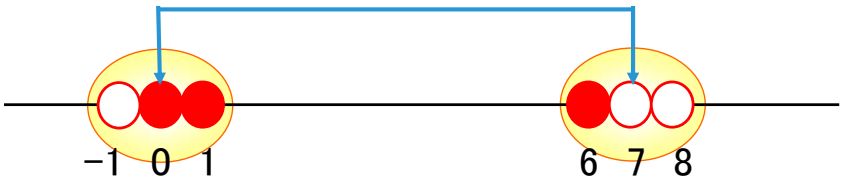


境界（値の間にある）の前後、合わせて計4点を取ってテストする方法

境界として指定されているポイントをON点と呼ぶ。ON点が、
・ 閉（同値クラスに含まれる）　同値クラスに含まれない近傍点がOFF点
・ 開（同値クラスに含まれない）　同値クラスに含まれる近傍点がOFF点
つまりOFFポイントは「境界に関する条件を満足しない近傍点」となる

COOOOI (Closed OFF Outside, Open OFF Inside)と覚えるとよい

Beizerの方法（境界 = 値の間）
JSTQBで採用
（高橋寿一氏も支持）
テストは、ONポイントの「0と7」、OFFポイントの「-1と6」を実施
（＋よく使われる3など）

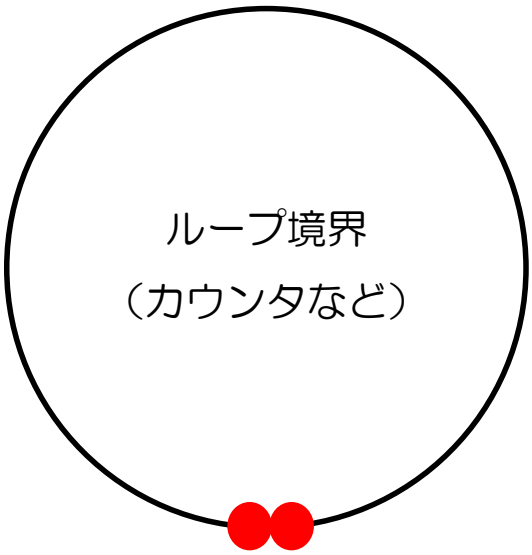


境界として指定されているポイントに着目し、その前後、あわせて合計6点を取ってテストする方法

Jorgensenの方法（境界 = 値）
英国標準(BS 7925-2)で採用
テストは、-1, 0, 1, 6, 7, 8を実施

FUJI xerox

境界値分析のTIPS



※ ANSI C標準では、
UINT_MAX (+65,535)

正規表現

- ひらがなを指定しようとして[あ-ん]とすると、「ぁ」が漏れる
- 片仮名を指定しようとして[ア-ン]とすると、[ァヴヵヶ]が漏れる（半角カナは[ｱ-ﾝ]とする）

特別な境界

- 数値の場合、-1と0と2のべき乗（大きいほうから）入力してみる
- 文字列長などサイズの境界値
- 閏日などの特異値の洗い出し

2008年8月4日にNTTは、249日のルータ連続稼動で「ひかり電話」が発着信不能になる旨のバグ告知をしました。
248日×24h×60m×60s×100=2,142,720,000
2^31 = 2,147,483,648
249日×24h×60m×60s×100=2,151,360,000

10msのカウンタが INT のループ境界を超えたのでしょう。

FUJI xerox

デシジョンテーブル

デシジョンテーブルのポイント

- 考えられる条件の組み合わせという点において、漏れや抜けがなくなる
- 入力条件が複雑に絡み合う場合に有効
- テスト実施の進捗状況が分かり、他者にも伝えやすい
- テスト実施を分業可能

- 一方で、以下には弱い
  - 条件そのものが間違っている
  - 条件の洗い出しに失敗　など

  - 条件の数が多い
  - → 条件の洗い出しに時間がかかりすぎてテスト時間がなくなってしまう！？

デシジョンテーブル

もし 条件	
ならば アクション	

＋

項目	記入

＝

条件項目	ルール（条件の組合せ）
結果項目	ルールに基づいたアクション

「BMI値の診断」のデシジョンテーブル

BMI値の診断	ルール1	ルール2	ルール3	ルール4
A. 18.5未満	Y	N	N	N
B. 18.5～25未満		Y	N	N
C. 25～30未満			Y	N
D. 30以上				Y
E. やせ	×			
F. 標準		×		
G. 肥満			×	
H. 高度肥満				×

基本的なディシジョンテーブルのパターン

◆ 傘を持つかどうかの判断

雨が降っている	Y	N	N	N	N
降水確率80%		Y	N	N	N
空気が湿っている			Y	N	N
下駄を投げたら逆さになった!?				Y	N
傘を持っていく	×	×	×	×	
傘を持っていかない					×

OR条件

2^4＝16条件
確認している
わけではない

◆ 出国審査の判断

パスポートあり	N	Y	Y	Y	Y
搭乗券あり		N	Y	Y	Y
出国カードあり			N	Y	Y
手荷物検査合格				N	Y
出国審査合格					×
出国審査不合格	×	×	×	×	

AND条件

2^4＝16条件
確認している
わけではない

勉強会参加の判断

仕事が立て込んでなくて、プライベートの予定もOK。
さらに、テーマに興味があるか、講演者に興味がある人が参加する。

仕事OK	Y	Y	Y	Y	N
プライベートOK	Y	Y	Y	N	
テーマに興味あり	Y	N	N		
講演者に興味あり		Y	N		
勉強会参加	×	×			
勉強会不参加			×	×	×

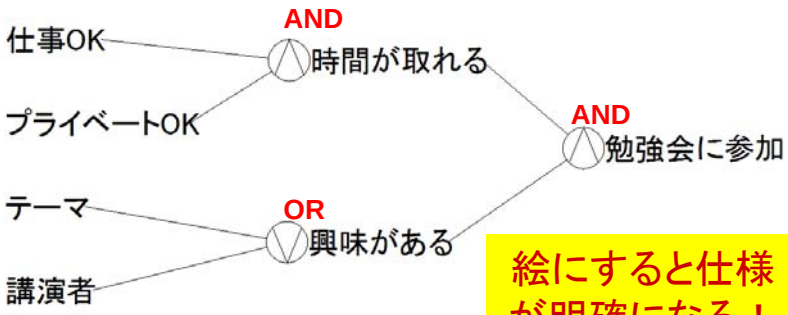
混合条件

2^4＝16条件
確認している
わけではない

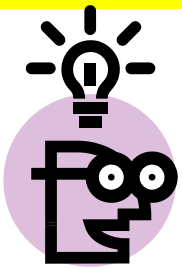
勉強会参加の判断（原因結果グラフ）

仕事が立て込んでなくて、プライベートの予定もOK。
さらに、テーマに興味があるか、講演者に興味がある人が参加する。

仕事OK	Y	Y	Y	Y	N
プライベートOK	Y	Y	Y	N	
テーマに興味あり	Y	N	N		
講演者に興味あり		Y	N		
勉強会参加	×	×			
勉強会不参加			×	×	×



絵にすると仕様が明確になる！



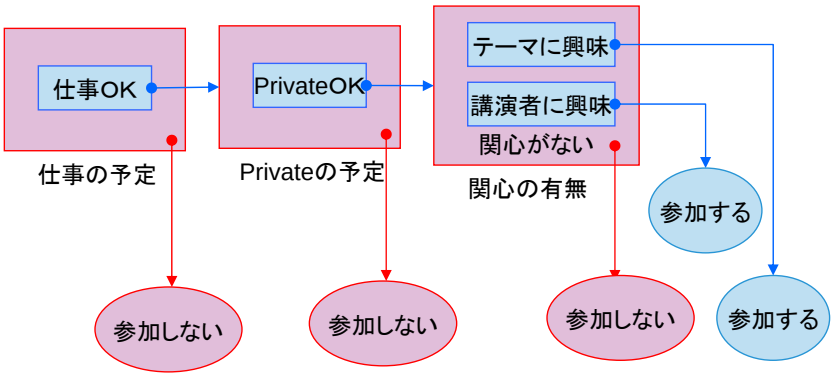
時間がある ＝ 仕事 AND プライベート
興味がある ＝ テーマ OR 講演者
参加する ＝ 時間がある AND 興味がある

FUJI xerox

勉強会参加の判断（CFD法）

仕事が立て込んでなくて、プライベートの予定もOK。
さらに、テーマに興味があるか、講演者に興味がある人が参加する。

仕事OK	Y	Y	N	Y	Y
プライベートOK	Y	Y		N	Y
テーマに興味あり	Y	N			N
講演者に興味あり		Y			N
勉強会参加	×	×			
勉強会不参加			×	×	×



CFDからデシジョンテーブルを作るのは容易

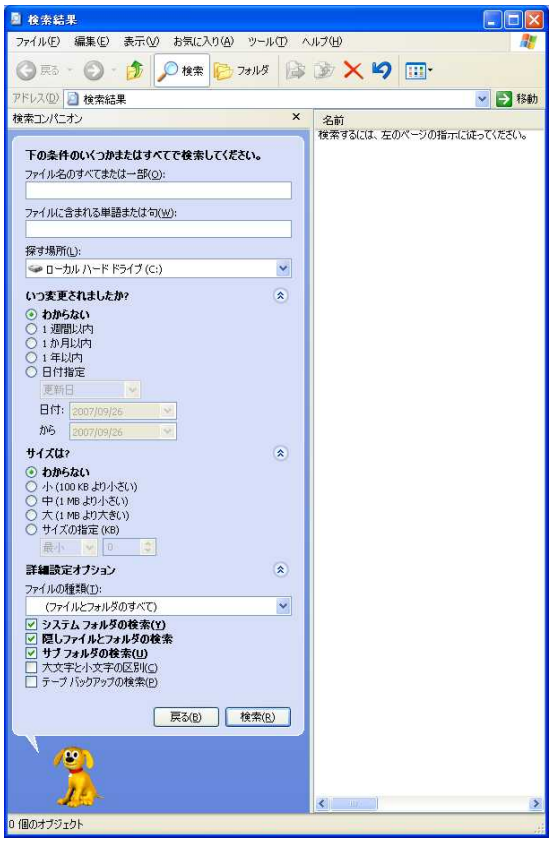
FUJI xerox

組合せテストといっても…

- そもそも組合せテストって必要なの？
  - ■ 機能Aと機能Bに依存関係がないことを機能Aと機能Bは直交しているという
    - そもそも機能の直交性を考慮するのが仕様や設計の目的
    - ところが現実には厳密性を要求されるプロトコルでさえ、HTTPヘッダーとHTMLの両方に同じ命令が書けた時期があった（そのため漢字コードなどの表示に混乱が起こった）
  - ■ 本当に機能Aと機能Bが直交していれば、単一網羅テストで良いはず
    - 機能A（a1, a2, a3）と機能B（b1, b2）の組合せテストはA×B（a1b1, a2b2, a3b1）でOK?
- 機能が直交していないことが分かっている場合の戦略
  - ■ 機能が直交していないのでテスト件数は掛け算となる
    - A（a1, a2, a3）とB（b1, b2）があったら、 A×B（a1b1, a1b2, a2b1, a2b2, a3b1, a3b2）
    - これではテストしきれないので直交していない部分を取り出して原因結果グラフを使用して、機能の論理関係の正しさを実証する
- 機能が直交している場合の戦略
  - ■ 本当は組合せテストは不要（単一網羅テストで良い）
    - **機能が直交していることの証明方法が無い**
      → 直交表を用いて２機能間、３機能間の直交性を実証し品質保証することが必要！

FUJI xerox

機能が直交（独立）している例



たぶん、このケースではHAYST法による組合せテストはしません。

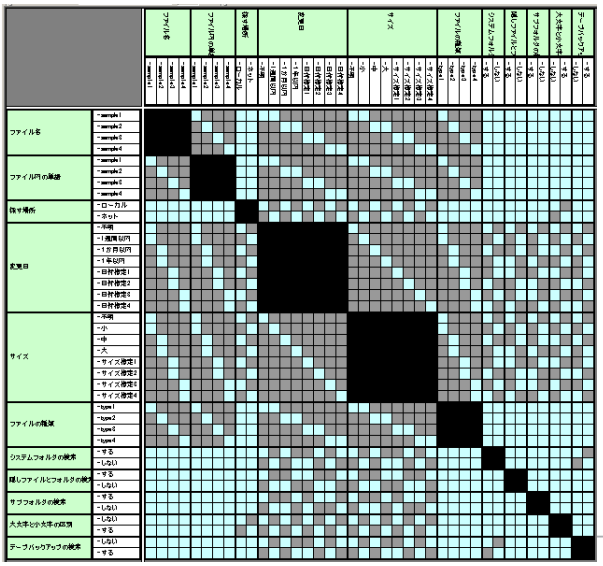
各因子の独立性が高いためです。

組合せテストとしては水準集約法（単一網羅テスト）を使用します。

FUJI xerox

単一網羅組合せ（HAYST法のテクニックの一つ：水準集約法）

No.	ファイル名	ファイル内の単語	探す場所	変更日	サイズ	ファイルの種類	システムフォルダの検索	隠しファイルとフォルダの検索	サブフォルダの検索	大文字と小文字の区別	テープバックアップの検索
1	sample1	sample1	ローカル	不明	不明	type1	する	する	する	しない	しない
2	sample1	sample1	ネット	1週間以内	小	type1	しない	しない	しない	する	する
3	sample2	sample2	ローカル	1か月以内	中	type2	する	しない	しない	しない	しない
4	sample2	sample2	ネット	1年以内	大	type2	しない	する	する	する	する
5	sample3	sample3	ローカル	日付指定1	サイズ指定1	type3	しない	する	しない	しない	する
6	sample3	sample3	ネット	日付指定2	サイズ指定2	type3	する	しない	する	する	しない
7	sample4	sample4	ローカル	日付指定3	サイズ指定3	type4	しない	しない	する	しない	する
8	sample4	sample4	ネット	日付指定4	サイズ指定4	type4	する	する	しない	する	しない



	2パラメータ間	3パラメータ間
組合せ網羅率	52.00%	18.12%
融合した単位での網羅率	52.00%	18.12%
禁則考慮の有無	有	有
禁則処理を除いた全ての組合せ数	700	7088
マトリクスが網羅する組合せ数	364	1284
マトリクスが網羅しない組合せ数	336	5804
禁則組合せの全ての数	0	0
組合せに考慮したパラメータの数	11	11
組合せに考慮した値の数	40	40
自由度からのマトリクスサイズ見積り	30	30
テンプレートサイズ	L8	L8
テスト項目数	8	8

水準集約法 → L8

	2パラメータ間	3パラメータ間
組合せ網羅率	100.00%	71.78%
融合した単位での網羅率	100.00%	71.78%
禁則考慮の有無	有	有
禁則処理を除いた全ての組合せ数	700	7088
マトリクスが網羅する組合せ数	700	5088
マトリクスが網羅しない組合せ数	0	2000
禁則組合せの全ての数	0	0
組合せに考慮したパラメータの数	11	11
組合せに考慮した値の数	40	40
自由度からのマトリクスサイズ見積り	30	30
テンプレートサイズ	L64	L64
テスト項目数	64	64

HAYST法では L64になります

FUJI xerox

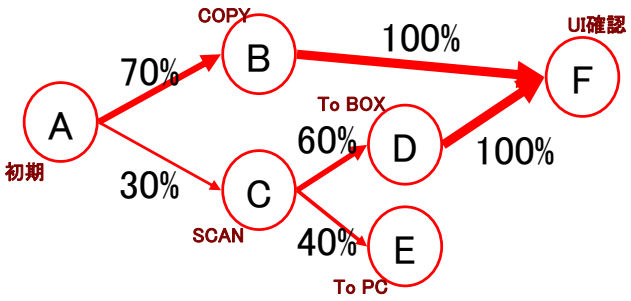
組合せテスト技法の選択

相互関係 / 要因の数	あり	なし
3～15	デシジョンテーブル（DT） 原因結果グラフ（CEG） CFD法	内部独立・設定集約 （http://www.jasst.jp/archives/jasst07e/pdf/A2-3.pdf） 水準集約法 直交表（OA） Pairwise
16～	適切な機能分割が できているか設計を見直す	

FUJI xerox

統計的テスト

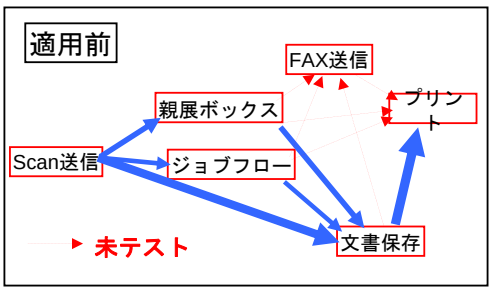
- 新製品のターゲットとなるお客様の使い方を調査し、シナリオを作成するとともに、信頼性の検証のため統計的テストを実施する。

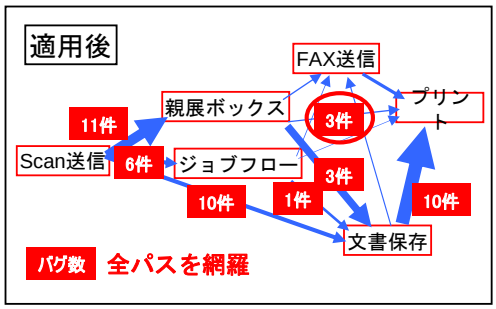


10000件テストを自動生成した場合

A→B→F:　7000件
A→C→D→F:　1800件
A→C→E:　1200件

のテストケースがランダムに生成される



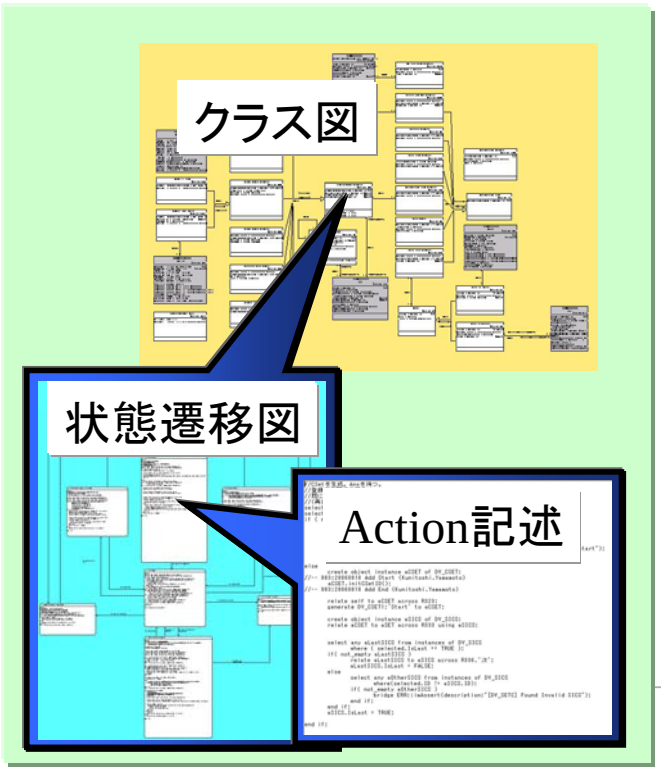


FUJI xerox

モデルチェッキング

Model

・シミュレーション実行可能（Executable）
・モデルがレビュー対象



C言語への自動変換結果

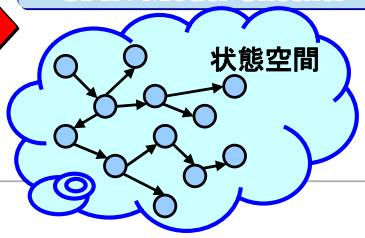
・製品そのもの（完全自動変換）
・コードはレビュー対象外

```
/* Set Close */
void
IOTim_dev_DV_SETC_Action_1( IOTim_dev_DV_SETC_
OoaEvent_t * const event )
{
  IOTim_dev_DV_
  IOTim
  /* SEL                    ROM INSTANCES OF DV_OUTS */
  v458 =  dev_DV_OUTS_s
*)Escher_SetGetAny( pG_IOTim_dev_DV_OUTS_extent );

  /* SELECT ONE aRegistingOUT RELATED BY aOUTS->DV_OUT[R311] */
  v461 = v458->mc_DV_OUT_R311;

  ......
```

スケルトンではない

Promelaへの自動変換

SPIN Model Checker

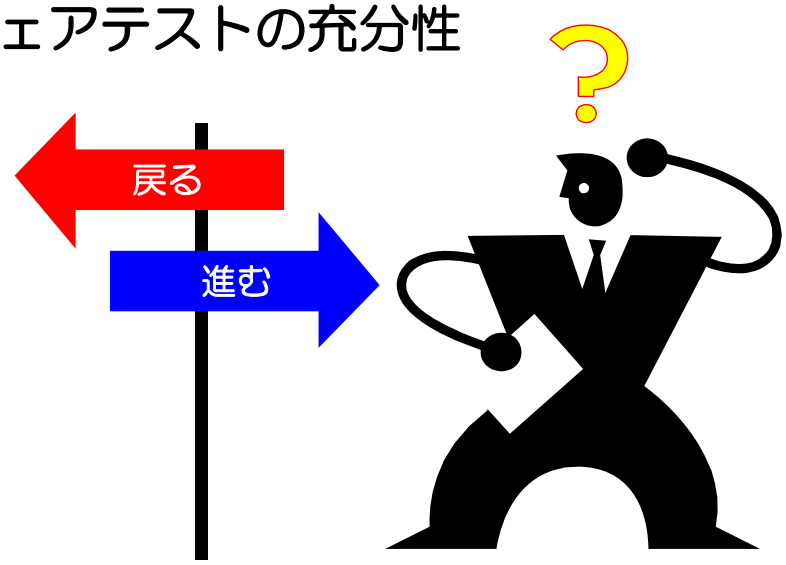


非同期通信に関する検証を実施（3万行、300万テストケース）

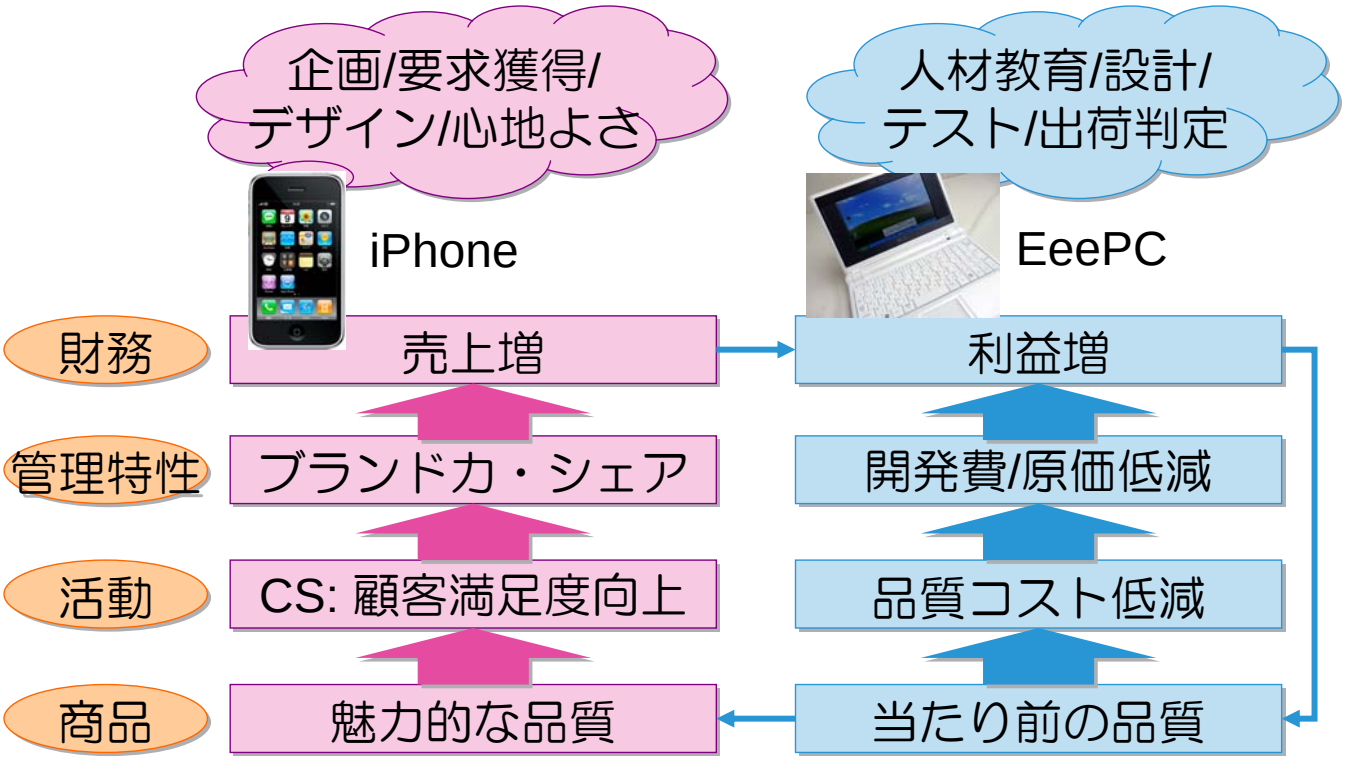
FUJI xerox

Ⅰ. ソフトウェアテスト技法

Ⅱ. ソフトウェアテストの充分性



品質と売上・利益との関係

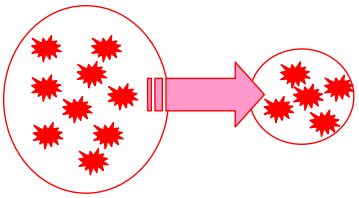


FUJI xerox

品質コスト

人材教育/ISO9001

予防コスト



バグ総数を減らす

レビュー/テスト

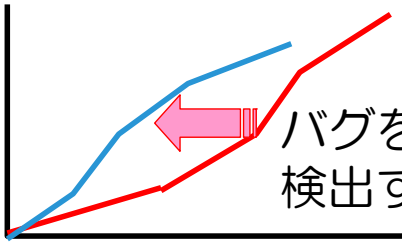
評価コスト

品質獲得

テスト/市場バグ対応

内部失敗コスト

外部失敗コスト

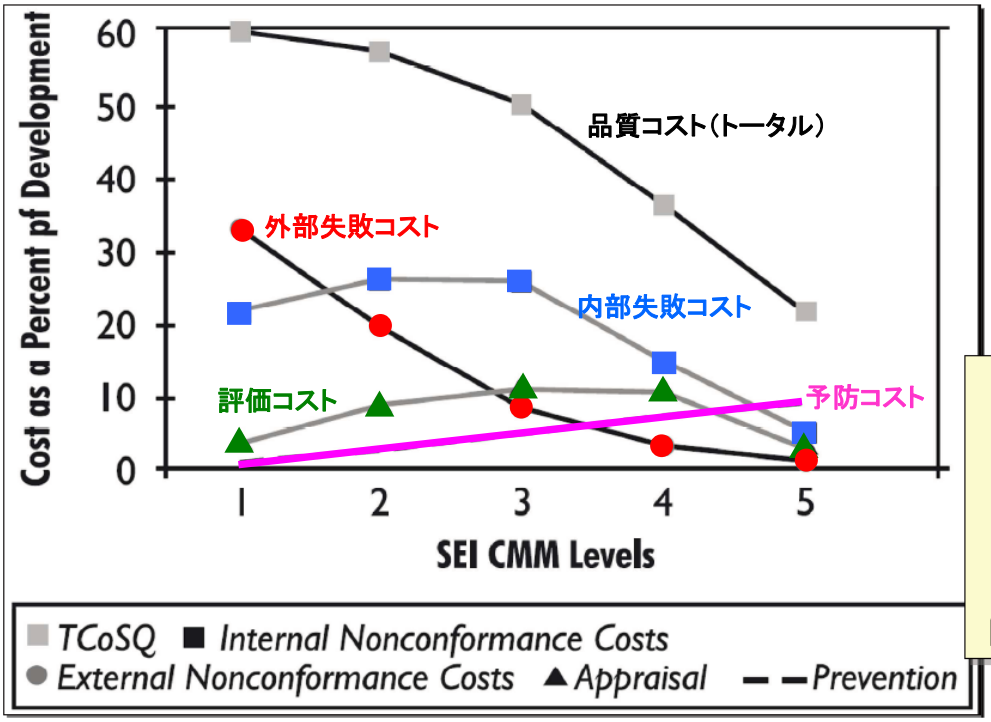


バグを前倒しで検出する

FUJI xerox

品質コストがどうなればよいのか？

品質向上への投資が正しく行われて改善が進んでいるかを確認する



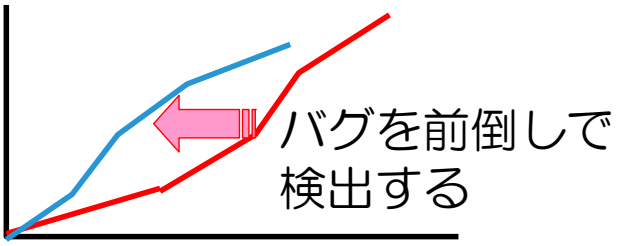
予防コスト投資
↓
失敗コストの低減
↓
評価コストの低減
↓
トータルコストの低減

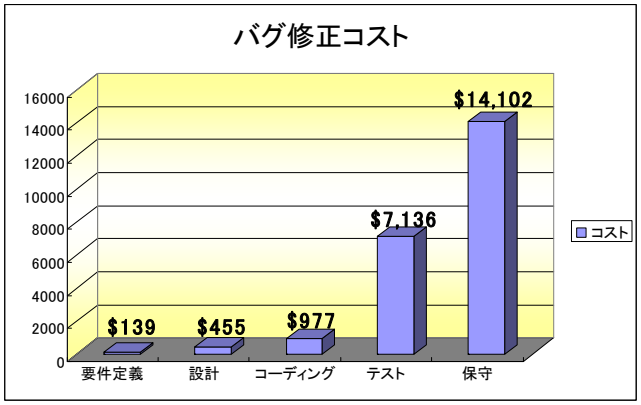
Krasner, H., "Using the Cost of Quality Approach for Software," CrossTalk, Nov. 1998, pp.6-11.

FUJI xerox

評価コストとテスト充分性

評価コスト





IEEE Computer Society, Vol. 34, No. 1, 2001

評価コスト ＜ 外部失敗コスト － 内部失敗コスト
＜ $14,102 － $7,136
＜ $6,966

$7,000以下のテスト工数でバグが1件見つかるならテストすべき！？
ゼロディフェクトを目指すことが基本。ただし市場機会損失には注意。

FUJI xerox

テスト充分性

- 「バグが検出される」ならテストをすべき

- バグが検出されるかどうかの見極めが大切
  - 見極めができる前提を整える
  - 動的にテスト実施状況を捉えてコントロールする

FUJI xerox

見極めができる前提を整えるとは？

- 「バグを全部見つけるのは無理」（Cem Kaner）
  ＝　テストで検出しにくいバグがある
  （ループ、並列処理、仕様書に無い境界値）
  →　そこはテスト以外で取れている前提を作る

- 仕様レビュー・コードレビューを実施する
  - 開発リーダがメンバーの実力を知ることが大切
  - すべてでなくて良い
- 単体テストを実施する（C1：デシジョン）
  - 単体テストは全て自動化する（回帰テストのため）
  - 単体テスト担当者にテスト技法を教育する
    →　単体テストの質を上げる

動的にテスト実施状況を捉えてコントロールするとは？

- 「欠陥の偏在」（ISTQBテストの原則より）
  ＝　欠陥の多くは特定の少数のモジュールに集中する

- 「ホットスポット」を見つけ「力」を集中する
  - McCabeの複雑度やバグ発生状況から特定する
  - 欠陥のレベル（重要度・タイプ）に応じた対応を実施する
- 「重大バグ」を残さずに検出する
  - FTAを活用して発生要因を全て潰す
  - 全体を網羅的にテストする（論理網羅＝決定表、入力網羅＝直交表、状態網羅＝2回の状態遷移等々）

テスト最終段階で重視すること

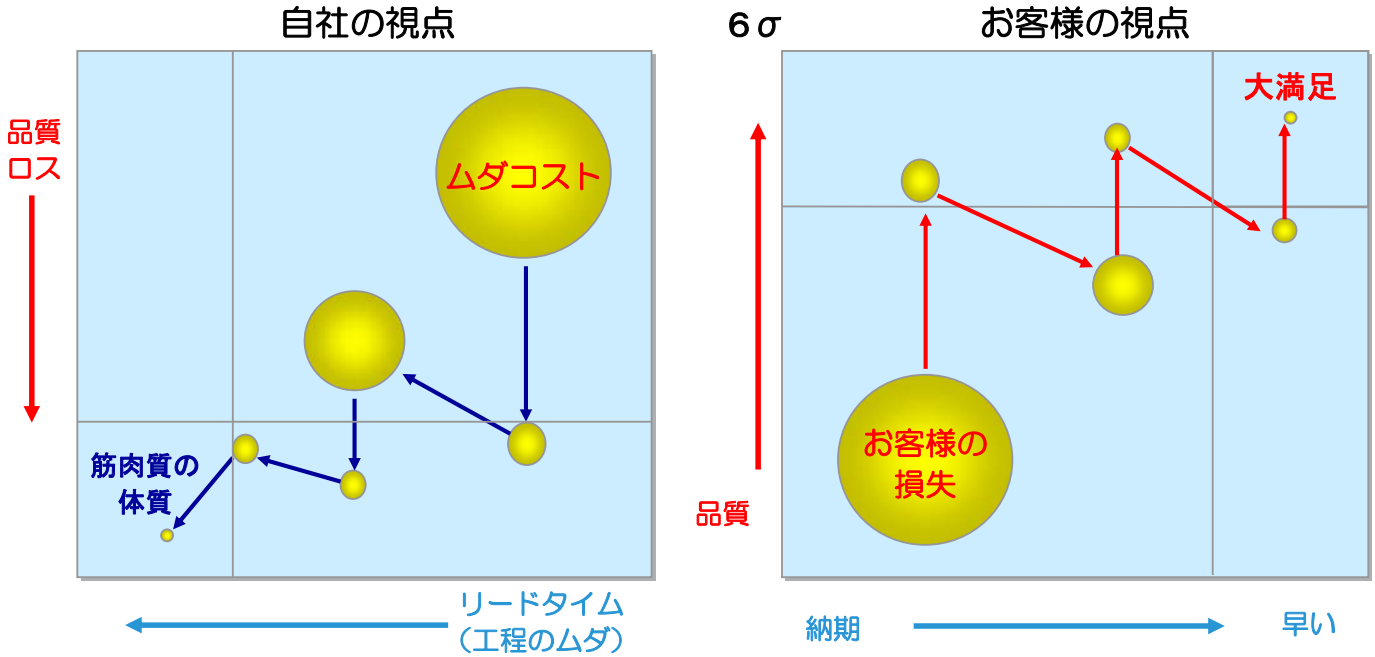
- 計画したテストが全て完了していること
- 重大バグが全て直っていること
- 連続稼動が続いていること
- 最終バージョンで以下が確認できていること
  - 重大バグが出ていないこと
  - 基本的なテストセット（少数）が通っていること
    （この時、パフォーマンスも確認する）
  - ハードウェアに改変があった場合、動作確認できていること
- テスト担当者が「感覚的にOK」といっていること
- テスト中の最終バージョンと、リリースするSWが同じであること
  - 開発者がこっそりチェックインしていないこと

FUJI xerox

改善の戦略的方向

まず、品質を改善する
次にスピードを上げる

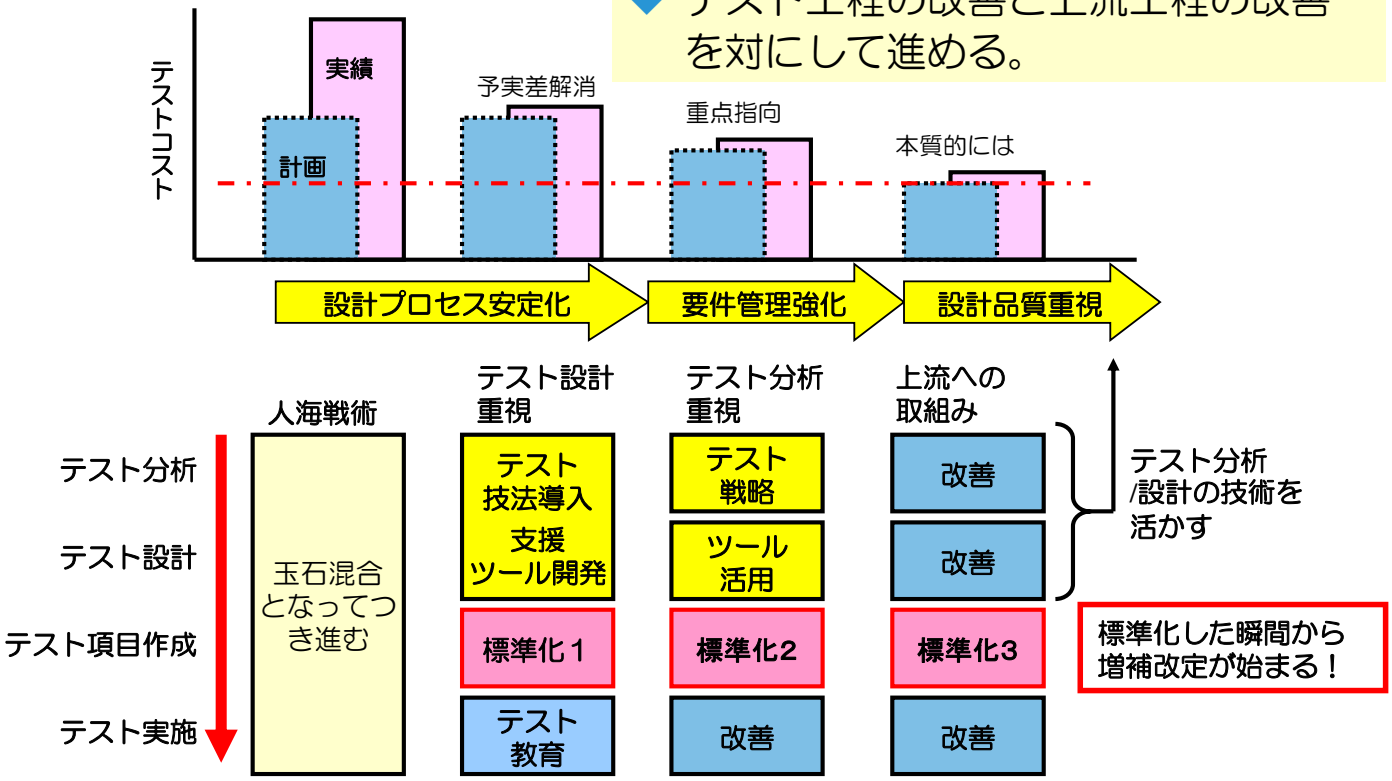
品質問題が発生するまで
スピードを改善する



FUJI xerox

段階を踏んだ改善アプローチ（PDCA）

◆ テスト工程の改善と上流工程の改善を対にして進める。



FUJI Xerox

ご清聴ありがとうございました。

FUJI Xerox