■JaSST'09Tokyo

ヘ イ ス ト
HAYST法の概要
・ 直交表を活用したソフトウェアテスト技法

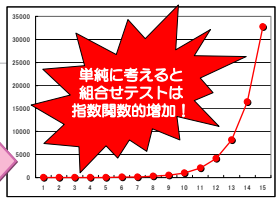
2009年1月29日（木）

富士ゼロックス株式会社
品質本部　秋山 浩一

FUJI xerox

HAYST法（直交表を活用したテスト手法）

- 機能組合せテスト設計の課題
  - 経験則でランダムに組み合わせると網羅率が低下する
  - 機能組合せを網羅的に行うとテスト項目数が爆発する



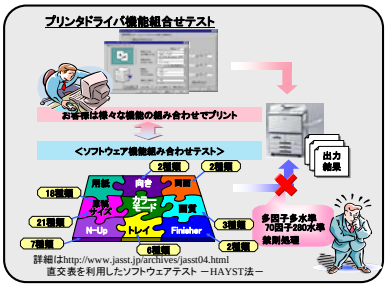
理由

- 対策前の組合せ網羅率
  経験で作りこみ　約30%

対策

直交表の活用
1. 網羅率向上
2. テスト項目数低減

- 対策後の組合せ網羅率
  HAYST法　>80%



FUJI xerox

HAYST法のテストライフサイクル

お客さまの要求分析
↓
仕様化（アーキテクチャ）
↓
テストレベル戦略

ステップ	基本的な考え方	狙い
1.テスト分析	顧客分析・仕様分析により、テストの十分性を確保	効果
2.テスト設計	目的機能の詳細の理解、テスト条件網羅率の最大化	効果・効率
3.テスト実装	ツールによる、テストケース生成の自動化	効率
4.テスト実施	ツールによる、テスト実施とバグ解析の自動化	効率

①テスト分析
- 単体テスト *Unit Test*
- 統合テスト *Sub System Test*
- システムテスト *System Test*
- 受入れテスト *Acceptance Test*

②テスト設計
- 設計仕様書 → FV表（Function Verification Table） → FL表（Factor Level Table）

直交表割付
- 応用：複合、連結、水準集約、ブロック別
- 基礎：直交表の選択、ダミー水準、抽象化、禁則回避、多水準

③テスト実装
- 直交表の特性を活かし低次と高次の列を使い分ける等
- 自動化ツール

④テスト実施
- テスト実行 → 直交性バグ解析ツール（原因追求） → 対策 → 設計へのフィードバック

FUJI xerox

テスト分析

- テストレベル戦略立案
  - テストレベルごとの目的、因子の粒度、観点を変える

開発視点 ↕ 顧客視点

  - 単体テスト：　関数（setBrightness）
  - 統合テスト：　仕様から可能な入力（明度：-100～100）
  - システムテスト：　お客様の観点での入力（明るさ：明るめ、暗らめ）
  - 受入れテスト：　実運用データ（室内と屋外、被写体の材質）
- テストの十分性を確保
  - お客様（6W2H）と仕様の分析/理解　⇒　目的機能の抽出
  - FV表の作成（目的機能でテストの十分性を確保する）
    - 目的機能（F）：　お客様は何をしたいのか？（Why?）
    - 検証（V）：　何を確認したら機能したといえるのか？（What?）
    - テスト技法（T）：どのテスト技法で検証するのか？（How?）
  - FL表の作成（目的機能の検証に必要な因子・水準を洗い出す）
    - 水準は設計（構造やコード）に立ち入って見つける

FUJI xerox

テスト分析

因子を選ぶ視点と抽出方法（6W2H）

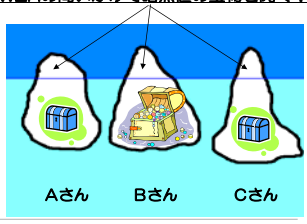
開発の視点
- Why(目的)
- What(仕様)
- How(設計)

お客様の視点
- How much（セールスポイント）
- Where(立脚点)、When、Who

誰のため？お客様のお客様
- Whom

企画・開発・評価・営業が集まってレビューする

6W2Hの問いかけで暗黙値の宝物を開く！

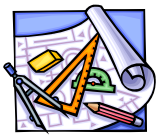


例） Why展開
1. 「下線」という機能(因子)の目的を考える
2. 「目的＝文字を目立つようにする」
3. 「同じ目的を持つ他の機能(因子)」を探す
4. 「大きくする」、「色をつける」、「丸で囲む」、「太字」等々が見つかる
5. 見つかった機能は「下線」と組み合わせることが多いはず！

FUJI xerox

テスト設計

- ラルフチャートを作成する：　目的機能の詳細の理解
  - 目的機能に対して組み合わせるべき因子・水準を設計する
    - 種類：　信号因子、誤差因子（ノイズ）
    - 重要度：　お客様の視点、開発の視点でダミー・抽象化
    - 関係性：　操作順序、禁則
  - 目的機能に対して状態・タイミングのテスト設計を実施する
    - 状態：　内部変数の組合せの意図的設計
    - タイミング：　タイミングにかかわる状態のテスト設計
- 直交表へ割り付ける：　テストの条件網羅率の最大化
  - 因子・水準、禁則、線点図情報の入力と割り付け
  - ツールを用いた、最小マトリクスへの割り付け最適化

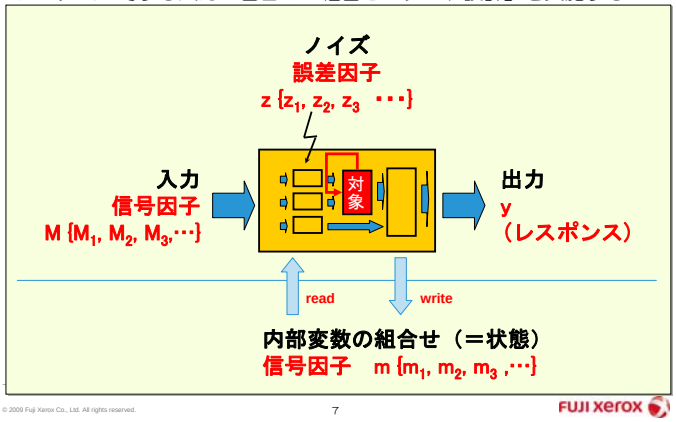


FUJI xerox

テスト設計

ラルフチャート（HAYST法のテストモデル）

システムに対する入力に着目し「組合せのテスト設計」を実施する



FUJI xerox

テスト設計

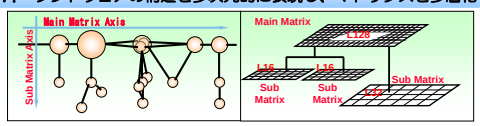
ラルフチャートの具体例： 割り勘システム

FUJI xerox

テスト設計

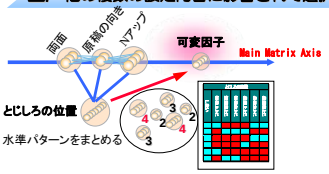
禁則処理問題への対応

1. ソフトウェアの構造を多次元的に表現し、マトリクスを多層化

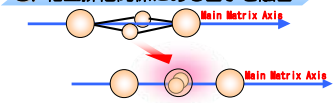


禁則処理問題の克服と
L128の中のリソースの節約
(=多因子多水準への対応)を同
時に解決

2. 他の複数の設定内容に影響されて選択肢自体が変化する場合に対応



3. 相互排他関係にある因子を融合



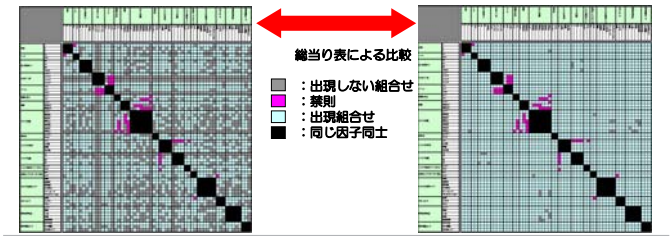
FUJI xerox

テスト実装

- テスト条件 ⇒ テストスクリプト
  - 事前条件、期待結果、事後条件の追加
- テストケース生成の自動化と検証
  - 網羅度による検証と「効果・効率の調整」

従来のKnowHowに基づいた方法
網羅率30%～40%程度

HAYST法による組合せ生成
網羅率70%～90%（テスト項目数は1/3）



FUJI xerox

テスト実施

- データ駆動スクリプトによるテスト実施
  - 直交表を1行読み込み共通の自動実行スクリプトで自動テストする
    - 市販のツールとの相性もよい
    - ただし期待結果との比較を作成する部分は難しい
  - 手動でテストを実施する場合
    - 1行のテスト方法を確実に伝えることによって、直交表の大きさ分のテストを正しく実施できる

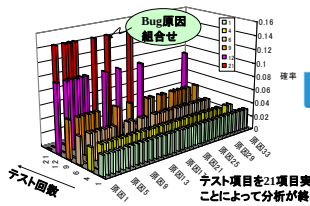
- 直交性を活用したバグ解析の自動化
  - わずか10個の因子でも2つの因子の組合せは$_{10}C_2$=45通り
    - ⇒ バグ原因の組合せを発見するために最悪45回のテストが必要
    - ⇒ バグの解析に時間がかかる
  - HAYST法では、直交表の性質（繰り返し、同じ組合せが現れている）を利用し、テストを続けながらバグを解析する
    - ⇒ バグ解析のための追加工数が少なくて済む

FUJI xerox

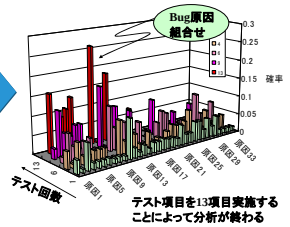
テスト実施

バグ解析の収束を示すデータ

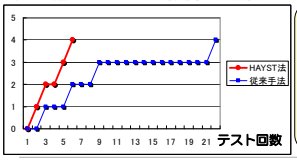
従来手法による分析



HAYST法による分析



4個の組合せバグを発見する速度の比較



テストケースの直交性を活用した分析により、

1. バグの絞込み速度: 従来手法の約2倍（21→13）
2. バグの発見速度: 従来手法の3倍以上（20→6）

※ 4件の組合せバグを、6回のテストで検出し、13回のテストで、それ以外に無いことを確認できた

FUJI xerox