

# テスト技術教育の事例紹介

ーテストエンジニアの知恵を育てるー



---

沼倉靖弘 近美克行 高松保裕 田中一巳  
株式会社シーイーシー  
PROVEQサービス本部

# はじめに

近年、組込み系の機器では搭載機能の拡大に伴い、開発量もテスト量も膨れ上がって来た。一方で開発の短納期化もあり、その必然として効果的かつ効率的なテスト方法を考える必要に迫られた。

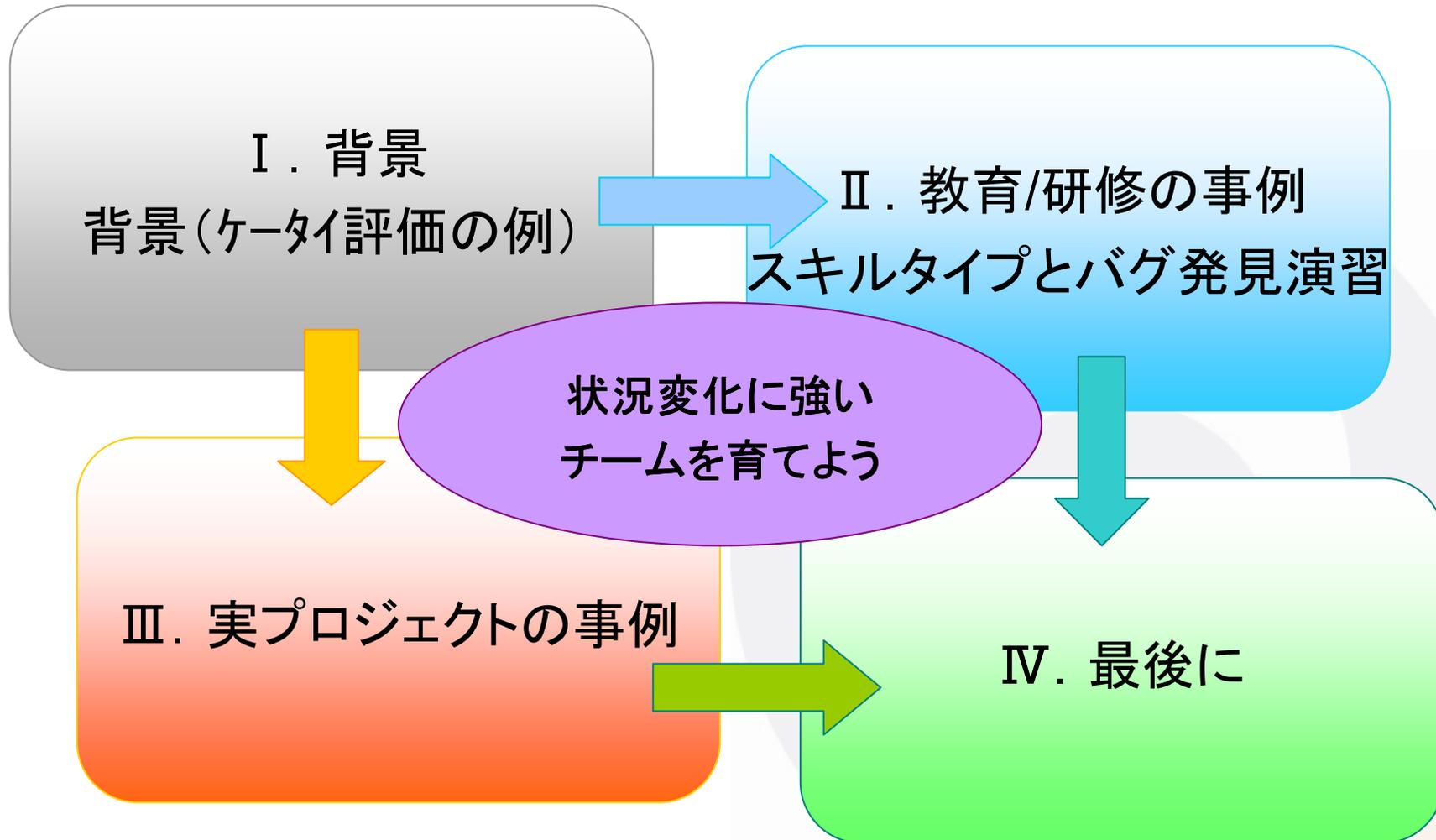
我々は打開策の一つとして探索的テスト手法を活用することにしたが、探索的手法にはテスト実施者へのスキル依存の問題がある。

今回の事例発表は、探索的テスト手法実施に必要なパターン推測と「テストを組み立てる」という考え方の教育を例にご紹介する。

また、探索的テスト手法を実際のソフトウェア評価プロジェクトでどのように取り込み、同時にOJTとして活用しているかという点についてもご紹介したい。

キーワード: 技術教育、探索的テスト手法、動的テスト戦略、マインドマップ、スキル特性、バグパターン、目的指向、テストを「組み立てる」という発想法

# 目次



# 背景(ケータイ評価の例)

---

- ケータイ電話評価を取り巻く状況が8年で激変
- 機能の増加と評価戦略の変遷

非公開情報に付き、一部削除

# 現在の試験戦略(1)

## 【短期間化】

商品サイクルが短期化したためソフトウェア評価に使える期間が減少した

## 【仕様確定とテスト開始時期の問題】

コンカレント開発/差分開発が主流で仕様情報が確定していない

## 【品質問題の社会現象化】

手順が多少複雑でもネット掲示板等の書き込みであつという間に広まってしまう



- ・全部はテストできないため、リスクベースの考え方
- ・あらかじめの試験表作成に頼らない方法
- ・バグパターンやテストパターンの蓄積を武器にする
- ・自動評価機が実用レベルになった
- ・ユーザクレームの動向に敏感になった

## 目的

第一は『ユーザクレーム防止』

第二に『新規開発箇所の品質向上』

第三に『既存機能に影響のないことの確認』

# 現在の試験戦略(2)

- あらかじめの試験表作成に頼らない方法
- 試験工数全体の効率化(試験設計+準備+実施+分析)

## ⇒探索的テスト手法を取り入れる

### ◆デメリットに注意が必要

試験漏れのリスク、テスト実施担当者のスキル依存性など  
(参照:レックスブラック著「ソフトウェアテスト実践ワークブック」)

### ◆解決策

- **複数のアプローチを組合せたテスト戦略を構築**  
(要件ベース+組合せ試験+探索的テスト手法+自動評価機)  
つまり・・・自動化できることは機械に任せる、人は人の特性を最大限利用してテストをしよう。  
(適材適所)
- **テストエンジニア教育を実施**

# 探索的テストとは？

---

ところで探索的テスト手法ってなに？

- 書籍にみる探索的テスト手法の説明

「システムを実際に操作しながら、テストの設計と実行を同時並行的に進める」と定義しています。(J. Bach)  
探索的テスト・プロセスは、説明するのは簡単ですが、実行するのは大変です。

(参照:レックスブラック著「ソフトウェアテスト実践ワークブック」)

- スキルってなに？経験ってなに？なぜ経験が必要なのか？
- どこでやめるのか？(次の探索テーマへ移るのはいつか？)
- 探索的手法のメリットとデメリット

# 知識の構造、運用能力と知恵(発想力)

## ②記憶力 関連性の思考力

繋がりを考えられる

スキル不足の人が陥るパターン  
一部分に囚われてしまう  
視野が狭い

## ①理解力 暗記力

個々の内容に関心を持つ

## ③抽象化 関連性の思考力

分類や組合せの見方ができ、決まった流れを見つけることができる

# スキルのタイプ

## ①決められたことに従うタイプ

理解力、暗記力

MCPC系のテストから利用

目の前のことに意識が向く

## ②知っていることから他に あてはめられるタイプ

記憶力、関連性の思考力

JSTQB系のテストから利用

想像することができる

## ③基本構成を見つけ出して、 他にあてはめられるように 抽象化できるタイプ

抽象化、関連性の思考力

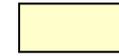
テスト技術系のテストを利用

先読みができる



# 参考データ(過去の社内スキル調査から)

非公開情報に付き、  
一部削除



短期経験者(3年未満)



長期経験者(3年以上)

理解力、暗記力

MCPC系のテストを利用

知識

記憶力、関連性の思考力

JSTQB系のテストを利用

知識+経験

抽象化、関連性の思考力

不具合事例分析と対応策検討力を評価

知恵+経験

# テストエンジニアのレベル別研修



- ソフトウェア検証の目的と社会的意義
- マインドマップ演習(自己紹介、機能マップを作成)
- バグパターンの基礎(ケータイの例)
- 機能試験設計演習(仕様書から試験作成)
- 基本動作の訓練  
(不具合表の書き方、報告・コミュニケーション、セキュリティ研修)

- **バグ発見演習**
- バグ事例の検討
- 観点ツリーの構成
- グループ演習

・テスト技法一般  
・ドメイン知識  
・組合せ試験の設計法  
は個別の研修及び  
資格試験向けの勉強会

- プロジェクトマネジメント講習  
(一般の開発プロジェクトと変わらない)
- 評価対象のリスク検討(技術リスクとは分けて検討)

プロセスが確定的なもの  
情報が纏まっているもの

# バグ発見演習(一例)

## 【事前準備】

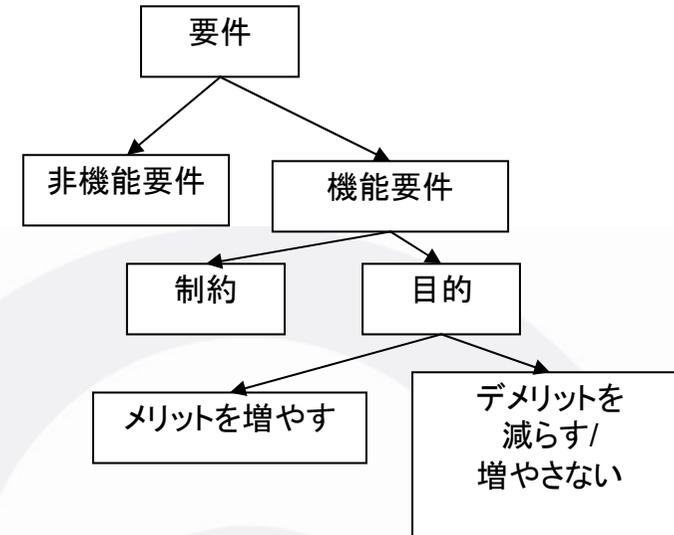
- VBAで作成した万年カレンダーソフト(不具合を意図的に混入したもの)を利用
  - 実施者にはバグパターン(ケータイの例)は予め講習しておく
  - マインドマップにより機能洗い出し研修は実施済み
- ※事前準備ではバグパターンとマインドマップは別々に教える。

## 【実施方法】

1. 機能一覧の作成などで情報収集(マインドマップ等をつかって機能マップを作成しても良い)
  2. 機能一覧から、「バグパターン」に当てはまると思う箇所や「複雑」だと思ふ箇所をテストさせる
- ◆大抵この段階では不具合発見できず
3. ヒントを出す。バグパターンの中からキーワード(「限界値」「境界値」「特徴的な設定は？」等)を告げたり、「複雑＝マインドマップなら枝が多い」等をアドバイスする。
  4. マインドマップからキーワードを埋め込ませる(枝の先に追記)
- ※「機能」とバグパターンからの「観点」という異質なものを結びつけさせる所が指導のポイント
- ◆1件でも不具合が検出
5. 類似不具合を探させる(情報の関連性を意識せるポイント)
- ※自分、または他の人が発見した不具合の情報も活用する。
- ⇒「バグパターン」を暗記させるのではない点に注意！！
- ⇒「バグパターン」は抽象化させたバグの情報で、複数のバグの「相違点」「類似点」を抽出したものの。
6. 発見したバグを自分達なりに分類し、その分類をした理由を報告させる。

# バグ発見演習(一例)

データの個数 / 不具合内容	難易度A:易B:普C:難			
バグ分類	A	B	C	総計
文字列・文言		2		2
改善			1	1
状態遷移			1	1
動作異常	1		3	5
動作異常(うるう年)			1	1
境界値	1			1
入力チェック	1		1	2
入力チェック(境界値)	1		2	3
入力チェック(不正値)	1		2	4
総計	7	8	5	20



- ・バグを組み込んだテスト練習用のおもちゃプログラム
- ・典型的なものを難易度別に20件組み込む
- ・初級エンジニアで4件程度、典型的には7件から10件
- ・類件がまだあることを伝えて、検討させると追加で2件～5件程度の追加発見がある。
- ⇒「類似の不具合」の存在を意識させ、それを推定する訓練
- ・もっと進めると仕様書がない(記載がない)状況での「要件」「機能目的」の推定にたどり着く



# 実際の評価プロジェクトでの適応

- 探索的テスト手法を工数の半分程度組み込んだ、テストプロジェクトを実施
- 探索的テスト手法では単位工数当たりの不具合発見数は、スクリプトテストに比べ高くなった
- 探索的テスト手法を用いることでバグ発見効率が高くなっただけでなく、担当者のテスト対象やテスト設計時にも適用できる観点に対する理解が進んだ

※バグの情報を元にした観点分析やテスト手法が分かりやすいのでバグ情報を中心に扱うが「バグ出し」指導が最終目的ではない



# 実際の評価プロジェクトでの適応

---

非公開情報に付き、削除





# 実際の評価プロジェクトでの適応

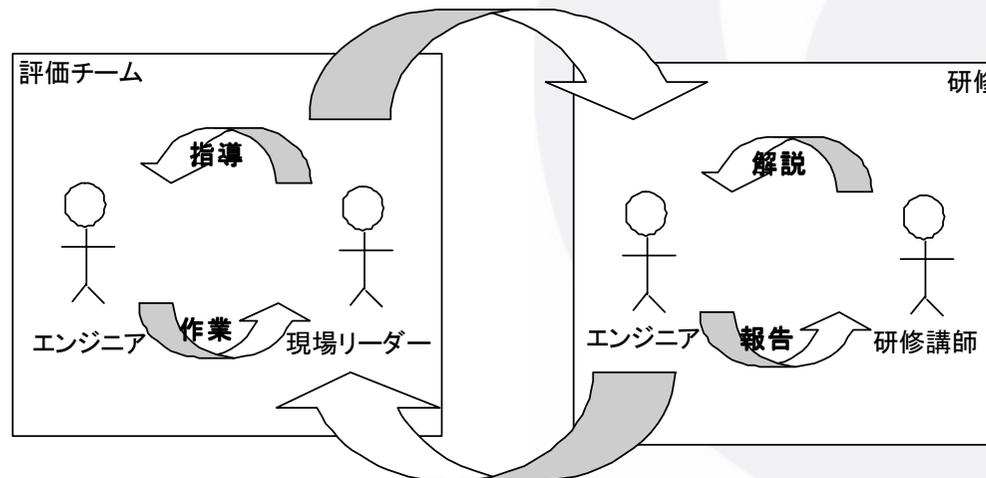
---

非公開情報に付き、削除



# 教育を実プロジェクトとリンクさせる

- 講義、演習、OJTの連携が大切
- 新手法導入のための先行教育⇒現場で実践⇒現場で指導を受ける(OJT)
  - ◆この段階では適切な指導を受ければ(バグ探索等)出来る様になっている
- 現場で指導を受ける(OJT) ⇒現場で指導を受けた内容やアドバイスの理由について説明を受ける(本人達による事例研究)
  - ◆これを繰り返すことで自立を加速させる



# 最後に

- 「変化に強いチームを育てよう」という目標
  - ⇒ 探索的テストを中心的アプローチとするテスト戦略
- エンジニアの探索的テストスキル向上
  - ⇒ 演習やOJTとを組み合わせる
- 探索的テストの運用実績からの発見
  - ⇒ チーム内のテストスキル向上が促進された
- 成功要因
  - ⇒ 探索的テストは加点評価テストに知的な楽しみを与える。  
(スクリプトテストは減点評価)
- 知識は「共通」と「一致」が求められるが、知恵は「バリエーション」が勝負(できるだけ多くのアイデアが欲しい)。
  - ⇒ 知恵によりテストの幅が広がり、チームの戦略的な状況適応能力が高くなる。

# ご清聴ありがとうございました。

沼倉 靖弘 近美 克行 高松 保裕 田中 一巳

株式会社シーイーシー PROVEQサービス本部 東日本サービス部  
事業推進部

〒216-0033 川崎市宮前区宮崎2-9-16（宮崎台システムラボラトリ）  
E-mail: { y\_numakura, k\_chikami, y\_taka , kazumi}@cec-ltd.co.jp



<http://www.cec-ltd.co.jp/service/kensyo/proveq01.html>

担当部署:コーポレートサービス本部 総務部 広報室

フリーダイヤル 0120-057-232

TEL:03-5789-2442