

セッション 5 15:50-16:50 (60分)

意図的にバグを混入させたソフトウェアを用いた研修 の実践と効果

NTTデータ 技術革新統括本部 システム技術本部 生産技術部
プロジェクトマネジメント・ソリューションセンタ
熊川 一平

自己紹介

熊川 一平（くまがわ いっぺい）

- 株式会社NTTデータ 技術革新統括本部 システム技術本部 生産技術部
プロジェクトマネジメント・ソリューションセンタ 課長代理

- テスト・品質保証に関する技術支援、研究開発
テスト自動化ツールの適用検討など社内案件の支援に従事

- 執筆・講演歴

- ITPro(日経BP社) 実践！テスト自動化の勘所～実装・実行の自動化
「キーワード駆動テストによるGUIテストの効率化」
<http://itpro.nikkeibp.co.jp/article/COLUMN/20121023/431821/>
- Micro Focus テストソリューション カンファレンス
テストツールを使ったプロセス改善のコツ、NTTデータが教えます
<http://special.nikkeibp.co.jp/ts/article/ac0g/147403/>
- Borlandソリューションカンファレンス
テスト自動化を成功させるには？ ～NTTデータの事例～
<http://special.nikkeibp.co.jp/ts/article/acab/158682/>
- JaSST' 14 Tokyo ★ベストスピーカー賞★
「探索的テストを活用したシステム開発手法」
<http://www.jasst.jp/symposium/jasst14tokyo/pdf/C4-2.pdf>
- SQiPシンポジウム2017 ★SQiP Best Report Effective Award★
「Session Based Test Managementによる探索的テストの実践」
https://www.juse.jp/sqip/symposium/2017/detail/day1/#session_A1-2
- 弊社コラム「イマ旬」への掲載多数
http://www.nttdata.com/jp/ja/insights/trend_keyword/

テスト技法に関する実態

当社の研修

講義と演習それぞれ1日の計2日間

テストの考え方や進め方

テスト設計技法

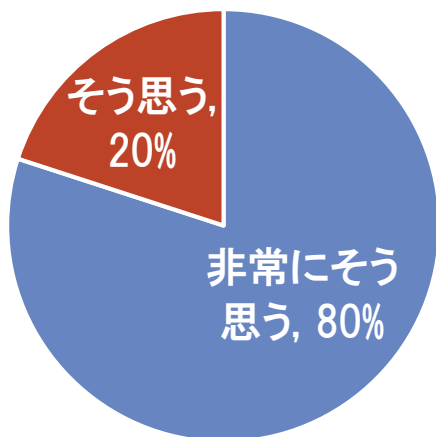
現場の実態

支援先でレビューすると・・・

テスト計画は作って終わり

勘と経験に依存したテスト

Q.この研修は満足いくものでしたか？



研修アンケートの結果は良好なのに
現場では学んだ知識が活用されていない

テスト技法に関する研修の状況

研修のジャンル	代表的なもの
A:技法や方法論の講義を行う施策	ソフトウェア品質セミナー /各社
B:技法や方法論の演習を行う施策	2日でマスターするソフトウェアテストセミナー /日科技連
C:技法や方法論を実践する施策	テスト設計コンテスト /NPO法人ASTER



当社の研修が劣っているとは思えない。
むしろ色々と頑張っているような気もする。

- 先に紹介した研修はAやBに該当
- Cに該当するような実践的なスキル診断も実施(*)

(*)テスト設計スキル評価方法の提案と実践事例
JaSST'14 Tokyo NTTデータ 町田 欣史

どうやって育成を進めればいいんだろう？

自分はどうやって学んできたんだろうか？と振り返ってみた。

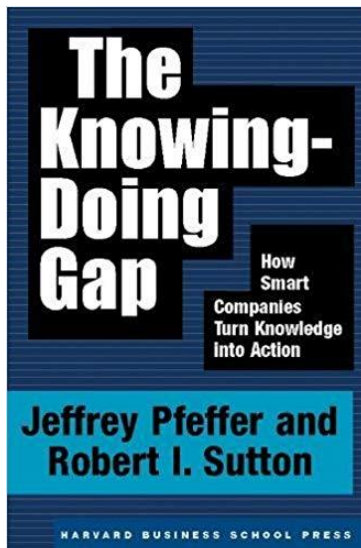


失敗・後悔

からの



学習・挑戦



キーワードは "Knowing-Doing Gap"

知っていてもできない

The Knowing-Doing Gap: How Smart Companies Turn Knowledge into Action
[Jeffrey Pfeffer](#) (著), [Robert I. Sutton](#) (著)

今回の取組み

目的

- 知識を得るだけでなく、実践する力も得られるような教育方法を考える
(knowing – doing Gapを埋められる教育方法)

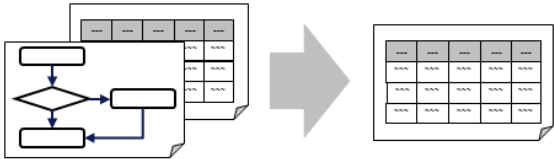



テスト設計技法を学ぶだけではなく、テスト設計を練習してみることでなく
「学び、活用し、失敗し、学びの本質に気が付きが得られる体験型のコンテンツ」



意図的にバグを混入させたソフトウェアを舞台に
テストの一連の流れを体験させる研修を作る

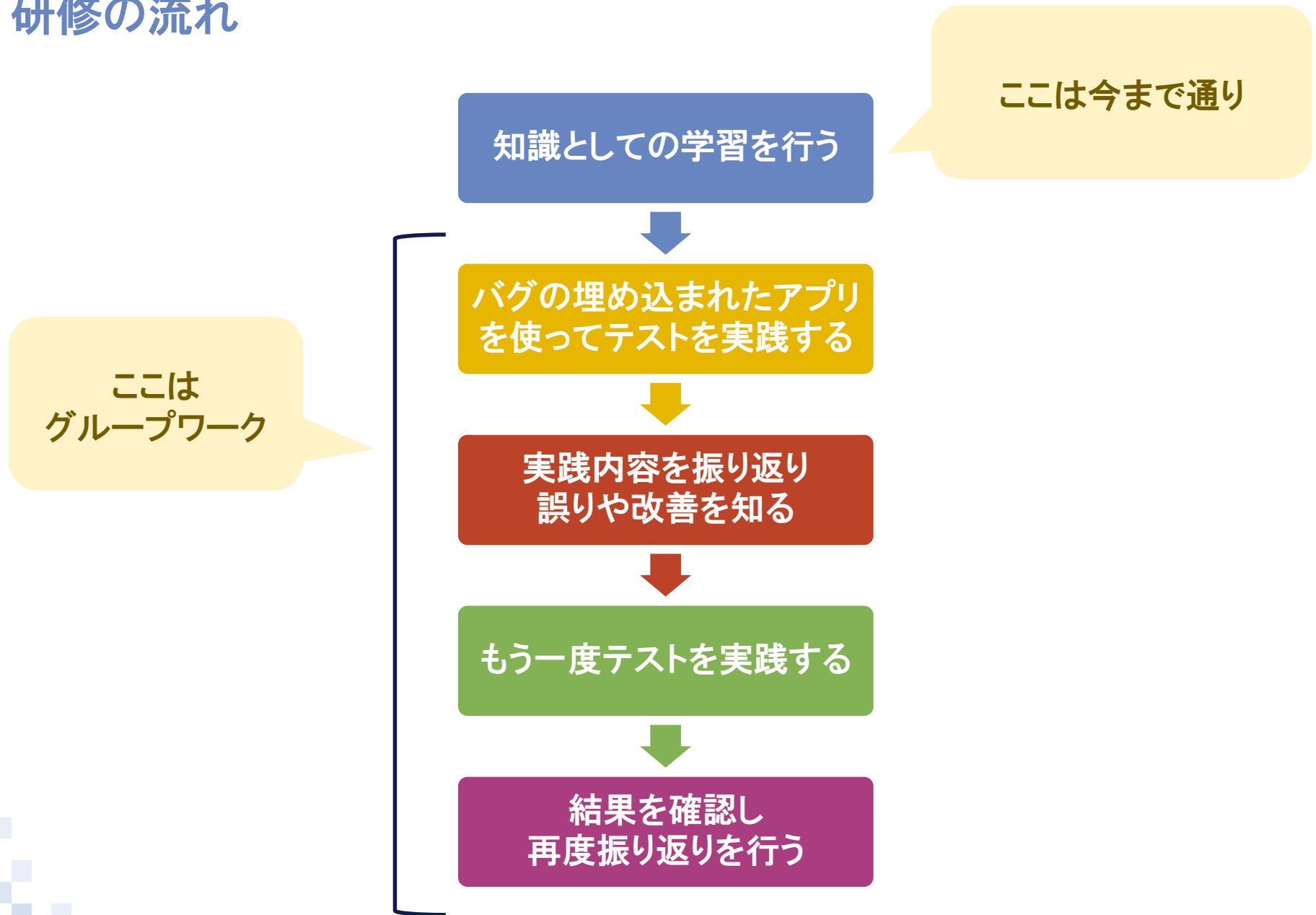
今回のターゲット

スクリプトテスト	アドホックテスト	
 <p>記述されたテストケースやテスト手順書通りにテストする。</p>	 <p>テストケースやテスト手順書を事前に準備せずにテストする。</p>	
	探索的テスト <p>テストをしながら、臨機応変にバグの潜んでいそうな個所をテストをしていく。</p>	モンキーテスト <p>入力や操作をランダムに行う</p>

※セッションベースの探索的テストです。

※セッション(2時間未満)ごとにテストチャーターを使って指示をします。

研修の流れ



バグを混入させたアプリケーション

家計簿 - メニュー

(c) 2018 NTT DATA Corporation

設定

費目の追加 費目の修正・削除

家計簿

明細の追加 明細の修正・削除 分析

保存 保存終了 保存せずに終了

家計簿 - 明細の修正・削除

年	月	日	収支	費目	金額	明細	店舗	理由
2018	1	1	収入	給与	¥300,000	残業なし	—	—
2018	2	1	収入	給与	¥305,000	残業あり	—	—
2018	3	1	収入	給与	¥300,000	残業なし	—	—
2018	4	1	収入	給与	¥30,000	残業なし	—	—
2018	5	1	収入	給与	¥300,000	残業なし	—	—
2018	6	1	収入	給与	¥300,000	残業なし	—	—
2018	7	1	収入	給与	¥300,000	残業なし	—	—
2018	8	1	収入	給与	¥300,000	残業なし	—	—
2018	8	29	支出	居住費	¥50,000	マンションSLDK	—	事故物件
2018	8	29	支出	交際費	¥4,000	送別会	居酒屋	—
2018	8	29	支出	交際費	¥1,800	歓迎会	—	ランチ形式
2018	8	29	支出	食費	¥800	朝食	—	—
2018	8	29	支出	食費	¥555	朝食	—	—
2018	8	29	支出	水道光熱費	¥8,000	ガス	—	多め
2018	8	29	支出	水道光熱費	¥8,000	電気	—	暑いのでクーラーフル稼
2018	8	29	支出	水道光熱費	¥3,000	上下水道	—	シャワーのみ
2018	8	29	支出	通信費	¥5,000	自宅光有線	—	固定
2018	8	29	支出	通信費	¥8,000	スマホ	—	ほうだい

修正 削除 明細のソート 月から費目の項目でソートします メニューに戻る

家計簿 - 明細の追加

2018 年 9 月 10 日

収支

☐ 収入 ☒ 支出

費目 太字は必須項目です

金額 半角数字で1~10,000,000の範囲で入力してください

以降は、全角文字で入力してください。前後のスペースは取り除かれます。

明細

店舗

理由

追加 メニューに戻る

参照

年	月	日	収支	費目
2018	1	1	収入	給与
2018	2	1	収入	給与
2018	3	1	収入	給与
2018	4	1	収入	給与
2018	5	1	収入	給与
2018	6	1	収入	給与
2018	7	1	収入	給与
2018	8	1	収入	給与
2018	8	29	支出	居住費
2018	8	29	支出	交際費
2018	8	29	支出	交際費
2018	8	29	支出	食費
2018	8	29	支出	食費
2018	8	29	支出	水道光熱費
2018	8	29	支出	水道光熱費
2018	8	29	支出	水道光熱費
2018	8	29	支出	通信費

実際の開発を想起させるためのドキュメントや議事録なども用意。

プロジェクト計画書
レビュー記録 など...

埋め込んだバグ

ポイント

学習対象のテクニックに精通した人が実践した内容からバグを作り出すと効果的！
Mutation Analysisも検討したが、学ばせたいテクニックと関係ないバグになってしまう。

高

発見難易度

低

発生条件が複数の処理にまたがっていて、探索的テストの考え方を理解して、ソフトウェアの挙動をよく観察し、仕様を推測しなければ見つけれないバグ。

複数のデータを編集する処理を組み合わせ、矛盾したデータを作成することで異常終了が誘発されるバグなど。

単一のデータを編集する処理内で、様々な編集パターンを試行することで発生するバグ。

特定の値を入力した場合のみ、誤った計算結果を出力するバグなど。

単一の処理内で代表的な利用方法を試行することで確認できるバグ。

画面レイアウトや使いにくいユーザインタフェース、ユーザマニュアルとの矛盾など。

学習を効果的にさせるポイント

知識としての学習を行う



バグの埋め込まれたアプリ
を使ってテストを実践する



実践内容を振り返り
誤りや改善を知る



もう一度テストを実践する



結果を確認し
再度振り返りを行う

1度振り返りの機会を
作ってあげる
(Gapに気付く)

振り返りの後にも
実践をさせる
(Gapを埋める)

答え合わせは最後に。
(極力自らの体験に)

振り返りのポイント

答えを教えるのではなく、身につけさせたい技術の考え方や陥りがちなアンチパターンをチェックリストで確認させる。

(例)

- 身につけさせたい技術の考え方
他のテスターの報告内容を聞いて、バグの見つけ方のコツや、探索の勘所に気が付きが得られましたか？
- 陥りがちなアンチパターン
テストする前から確認しようと思っていた箇所だけでなく、テスト対象の振る舞いを見て動的にテストの内容を検討できていましたか？

答え合わせのポイント

ただ答えを提供するのではなく、どうあれば良かったのか？という情報を提供。

※バグは偶然見つかることも多い。

バグ事象の解説

家計簿で使われている費目は削除ができない。が、修正はできる。
家計簿で使われている費目を、いったん別の費目名称に修正することで、削除できてしまう。

家計簿 - 費目の修正・削除		
No.	収支	費目
1	支出	居住費
2	支出	食費
3	支出	水道光熱費
4	支出	被服費
5	支出	交際費
6	収入	給与
7	支出	通信費

家計簿 - 費目の修正・削除		
No.	収支	費目
1	支出	居住費
2	支出	食費
3	支出	水道光熱費
4	支出	被服費
5	支出	交際費
6	収入	給与
7	支出	通信費2

修正

削除

エラーにならない

どうやったら見つかったか？

明細が存在する費目を削除するとどうなるだろう？と着想できれば、費目削除時のチェック条件に明細の有無があることを確認できる。同様に修正も確認すれば、削除と修正でチェック条件が異なることに気付くことができる。あとは、チェック条件の違いから不正な操作ができるか検討できると、このバグを見つけることが出来るかもしれない。
また、セッションシートの検討時に、単体テスト実施報告書の議事録で示唆されていた入力チェック条件に関するリスクを考慮できていると見つけやすい。

カークパトリックの4段階モデルを使って評価を実施

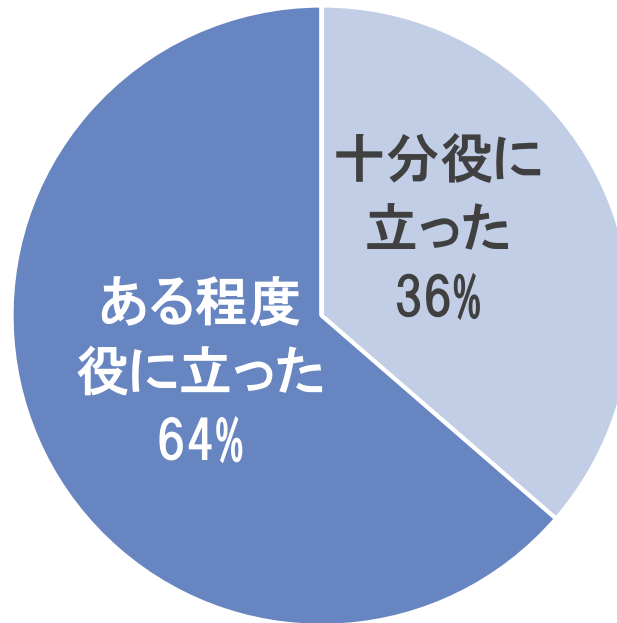
カークパトリックの4段階モデル	提案手法の効果の仮説	評価方法
1：反応 (研修満足度)	受講者は学習自体に満足している.	アンケート
2：学習 (学習到達度)	受講者の探索的テストに関する理解度が向上している.	アンケート
3：行動 (行動変容度)	受講者は探索的テストが実践できている.	モニタリングおよびテストの結果分析
4：結果 (成果達成度)	受講者は探索的テストを活用して成果を挙げている.	実際のソフトウェア開発プロジェクトのデータ分析



いまの研修だと「2:学習」までしか到達していないかも。
「4:結果」まで到達したい。

Lv1:反応 満足度をアンケートで評価

この研修は皆さまの業務の役に立ちましたか？



その他の選択肢は0票
「どちらとも言えない」
「あまり役に立たなかった」
「ほとんど役に立たなかった」

被験者の属性	人数
探索的テストの経験者	2
ソフトウェアテストの専門家	2
開発経験の豊富な技術者	2
プロジェクトマネジメント分野の専門家	2
その他	3
合計	11

システム開発経験年数	被験者数
0～4年	5
5～9年	3
10年以上	3

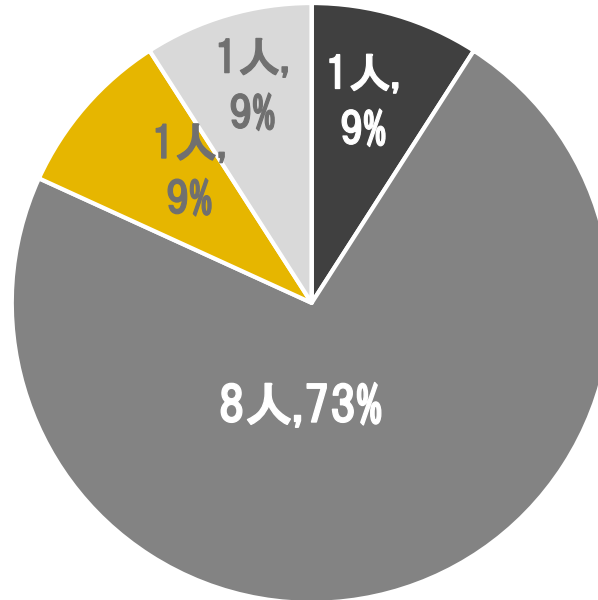
Lv2:学習 習熟度をアンケートで評価(1/2)

質問内容	アンケートの回答(平均値)	
	受講前	受講後
テスターとして、探索的テストの技法を用いてテストを実施できますか？	2.4	3.5
テスト管理者として、探索的テストの技法を用いてテストを管理できますか？	2.2	3.1
テスターとして、当社の探索的テストの進め方やルールに従ってテストを実施できますか？	2.5	3.7
テスト管理者として、当社の探索的テストの進め方やルールに従ってテストを管理できますか？	2.3	3.3

習熟度に関するアンケートの選択肢	点数
全ての作業を独力で遂行でき、後継に指導ができる	5
全ての作業を独力で遂行できる	4
ある程度の作業を独力で遂行できる	3
上位者のサポートがあればある程度の作業を遂行できる	2
上位者のサポートがあっても作業を遂行できる自信はない	1

Lv2:学習 習熟度をアンケートで評価(2/2)

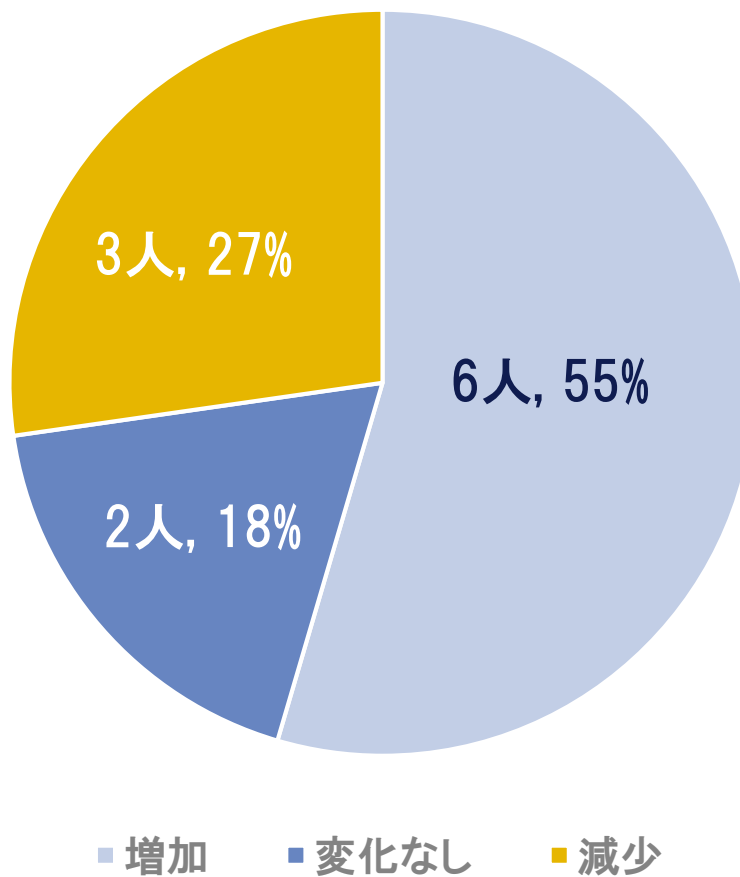
見つけられなかったバグは、もう1度同じような機会があれば見つけられますか？



- 5: そう思う
- 4: ややそう思う
- 3: どちらとも言えない
- 2: あまりそう思わない
- 1: そう思わない

Lv3:行動 テスト結果の分析

発見難易度の高いバグの摘出数に関する変化



研修模様の
モニタリング
もやったぞ



Lv4:結果 実案件での適用結果(1/2)

プロジェクト:Androidアプリケーションの第三者検証

テスター:4名

提案手法による学習	テスター名	テスター経験年数	探索的テストの経験
未実施	テスターA	3年5か月	有
未実施	テスターB	8か月	無
実施	テスターC	2年10か月	有
実施	テスターD	3年6ヶ月	無

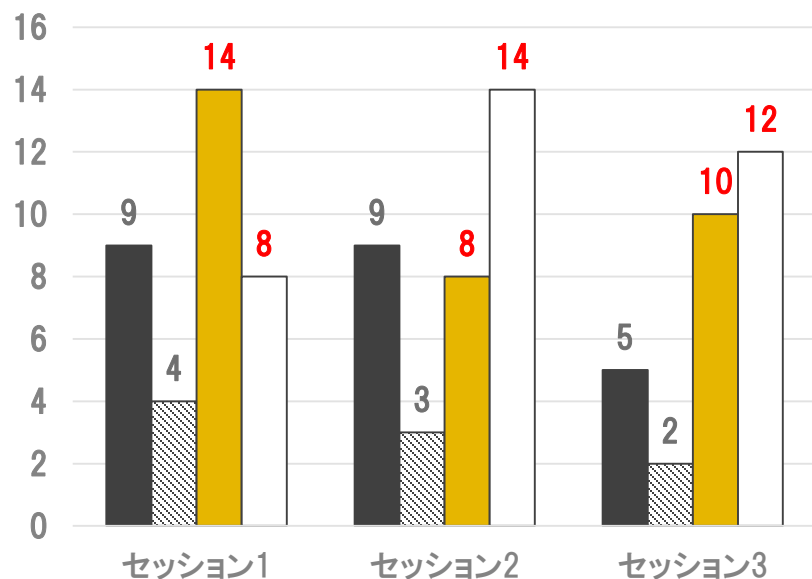
結果(摘出バグ数)

テスター名	実施セッション数	摘出バグ数(総数)	摘出バグ数(1セッションあたり)
テスターA	9	17	1.9
テスターB	8	5	0.6
テスターC	6	22	3.7
テスターD	5	30	6.0

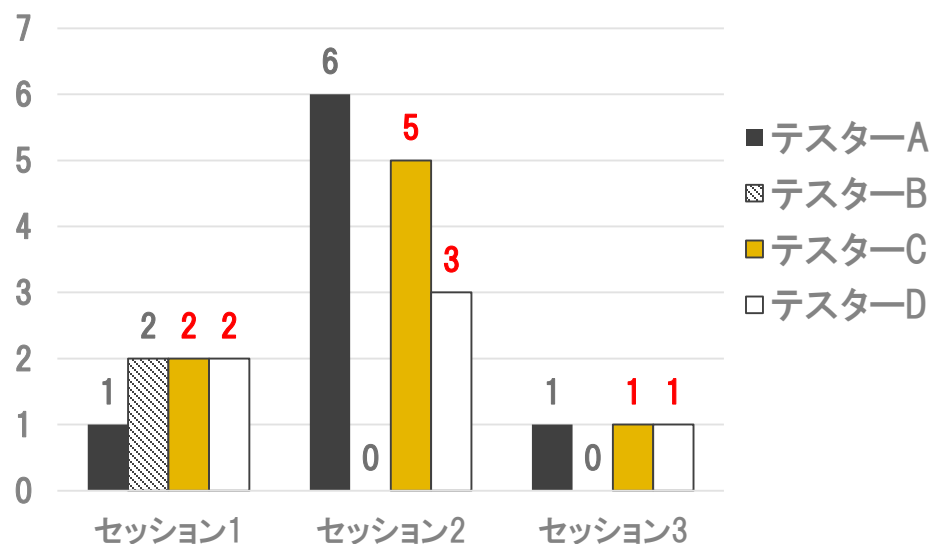
Lv4:結果 実案件での適用結果(2/2)

セッション内容の偏りを考慮し、同じテストチャーターで3セッション実施。

■テストの方向性に対するコメント数



■抽出バグ数



提案手法による学習	テスター名	テスター経験年数	探索的テストの経験
未実施	テスターA	3年5か月	有
未実施	テスターB	8か月	無
実施	テスターC	2年10か月	有
実施	テスターD	3年6ヶ月	無

被験者からの主な意見

実際に体験できる学習コンテンツは有効である。

意図的にバグが埋め込まれたテスト対象にもっとバリエーションがあったほうが良い。

ドキュメントからは学習できない実践の感触を掴めるのが良かった。

テスト実施までの学習コンテンツでは、受講者間でバグの摘出数を競い合うことでモチベーションが上がる。その反面、勝ち負けにこだわってしまい、手段に拘らずバグの摘出を目指した結果、本来の目的である学習がおろそかになる可能性がある。

最初は戸惑いを感じたが、進めていくうちに理解が追い付いた。他の社員にも推奨したい。

実際のアプリケーションを用いてテストすることで、探索的テストを直感的に理解できる。

バグ解答例を見ることでテスト経験の浅いメンバでもテストのポイントが理解できるので、学習前では見つけられなかったようなバグを見つけることができると思う。

結果の考察と今後について

結果は概ね理想に近い。

Knowing-doing gapを埋めることのできる学習コンテンツになったと言えそう。
カークパトリックの学習モデルにおけるLv4:結果の検証については、もう少しデータがほしいところ。

2次的な効果として、技術の普及展開にも役立っている。

(コンテンツさえ用意すれば自主的な学習にさせることもできる)

今後は探索的テスト以外の学ばせたいコンテンツに対して拡大させていきたい。

