JaSST'20 Hokkaido 基調講演

NTT DATA
Trusted Global Innovator

「なぜ大規模SIerで探索的テストを推進しているのか？ ～NTTデータが目指すソフトウェアテストの世界～」

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 ソフトウェア技術センタ
熊川 一平

注意事項

本セッションでの発言、およびスライドの内容には
発表者の主観的な理解や意見が多数含まれています。
多くの場合、それらはロクに検証もされていません。
ご利用にあたっては十分にご注意ください。
「なんか一生懸命しゃべってんなこのオッサン」ぐらいの気持ちで見ておくことをお勧めします。

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 ソフトウェア技術センタ
テクニカルグレード イノベータ(ソフトウェアプロセス)

- ソフトウェアテスト/ソフトウェア品質保証を中心としたR&D活動と、現場適用支援に従事

【主な講演歴】

- JaSST'14 Tokyo ベストスピーカー賞
- JaSST'19 Tokyo ベストスピーカー賞
- ソフトウェア品質シンポジウム2017 SQiP Best Report Effective Award
- ソフトウェア品質シンポジウム2019 SQiP Best Paper Effective Award

略歴・趣味など

略歴：

- 大阪府 交野市 生まれ （関東の人はだいたい読めない）
- 東海大付属 仰星高等学校 （中等部から。元MLB 上原選手の出身。ラグビーも有名）
- 岡山県立大学 情報システム工学部 (ボギー＝+1で卒業)
- （株）NTTデータ 入社

趣味：

- 野球観戦 （目指せパ・リーグ3連覇）
- 料理 （テレワークが増えて料理が楽しい。最近は子供のお弁当作りに注力。）

NTTデータ

■ 名　称

株式会社エヌ・ティ・ティ・データ
NTT DATA CORPORATION

■ 本社所在地

〒135-6033 東京都江東区豊洲3-3-3
豊洲センタービル

■ 交　通

東京メトロ有楽町線 豊洲駅 3番または1b出口徒歩1分

■ 設立年月日

1988（昭和63）年5月23日

■ 資　本　金

1,425億2,000万円（2019年3月31日現在）

■ 事業内容

- 電気通信事業
- データ通信システムの開発および保守の受託、販売ならびに賃貸
- データ通信システムに係るソフトウエアまたは装置の開発および保守の受託、販売ならびに賃貸
- データ通信システムに係る建設工事ならびにその他の建築工事および設備工事の請負
- インターネット、ケーブルテレビ、通信衛星等のネットワークを利用した情報処理、情報仲介および情報提供業務ならびに商取引および決済処理業務
- マルチメディア関連の音声、映像、データ等のコンテンツの制作および販売
- 経理事務、給与計算、各種保険手続等企業の各種事務処理の代行
- 著作権、工業所有権、ノウハウその他の知的財産権の取得、利用方法の開発、使用許諾、管理および譲渡ならびにこれらの仲介
- 広告宣伝に係る広告媒体の開発および販売ならびに広告代理店業
- 不動産の賃貸、仲介、保有および管理
- 労働者派遣事業
- 損害保険代理業および生命保険の募集に関する業務
- 前各号に関する企画、調査、研究、研修およびコンサルティングの受託
- その他商業全般
- その他前各号に関する一切の業務

■事業セグメント別業績（2019年3月期）

NTTデータグループ

総資産
2兆4,760億円

売上高
2兆1,636億円

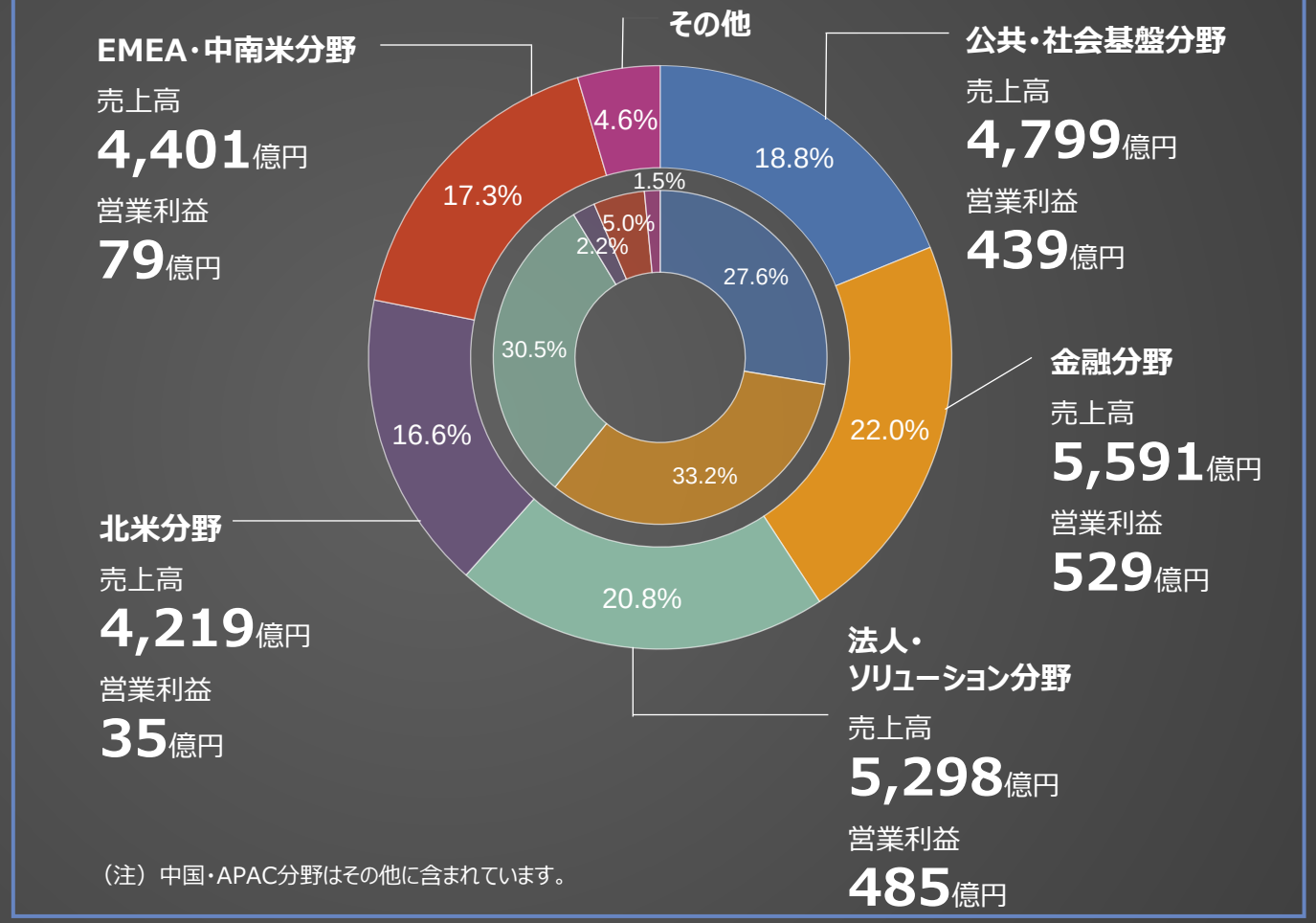
営業利益
1,477億円

従業員数
123,884人（連結）

連結子会社
307社

事業の多様化を推進するビジネスポートフォリオ

売上高(外円)　営業利益(内円)



（注）中国・APAC分野はその他に含まれています。

（注）2019年3月期より国際財務報告基準（IFRS）を適用しています。

NTT DATA

経歴

1. 大規模金融機関向けの勘定系/周辺系システム開発

経歴

1. 大規模金融機関向けの勘定系/周辺系システム開発

↓　社内異動制度を活用して自分で異動

2. テスト自動化を推進する全社組織

経歴

1. 大規模金融機関向けの勘定系/周辺系システム開発
2. テスト自動化を推進する全社組織

↓

3. 探索的テストの推進

経歴

1. 大規模金融機関向けの勘定系/周辺系システム開発
2. テスト自動化を推進する全社組織
3. 探索的テストの推進
   ↓　↑
4. テスト・品質に関するプロジェクト支援

北は北海道。南は福岡まで現地で支援！
札幌には1年ぐらい居ました。スープカレー食べまくってました。

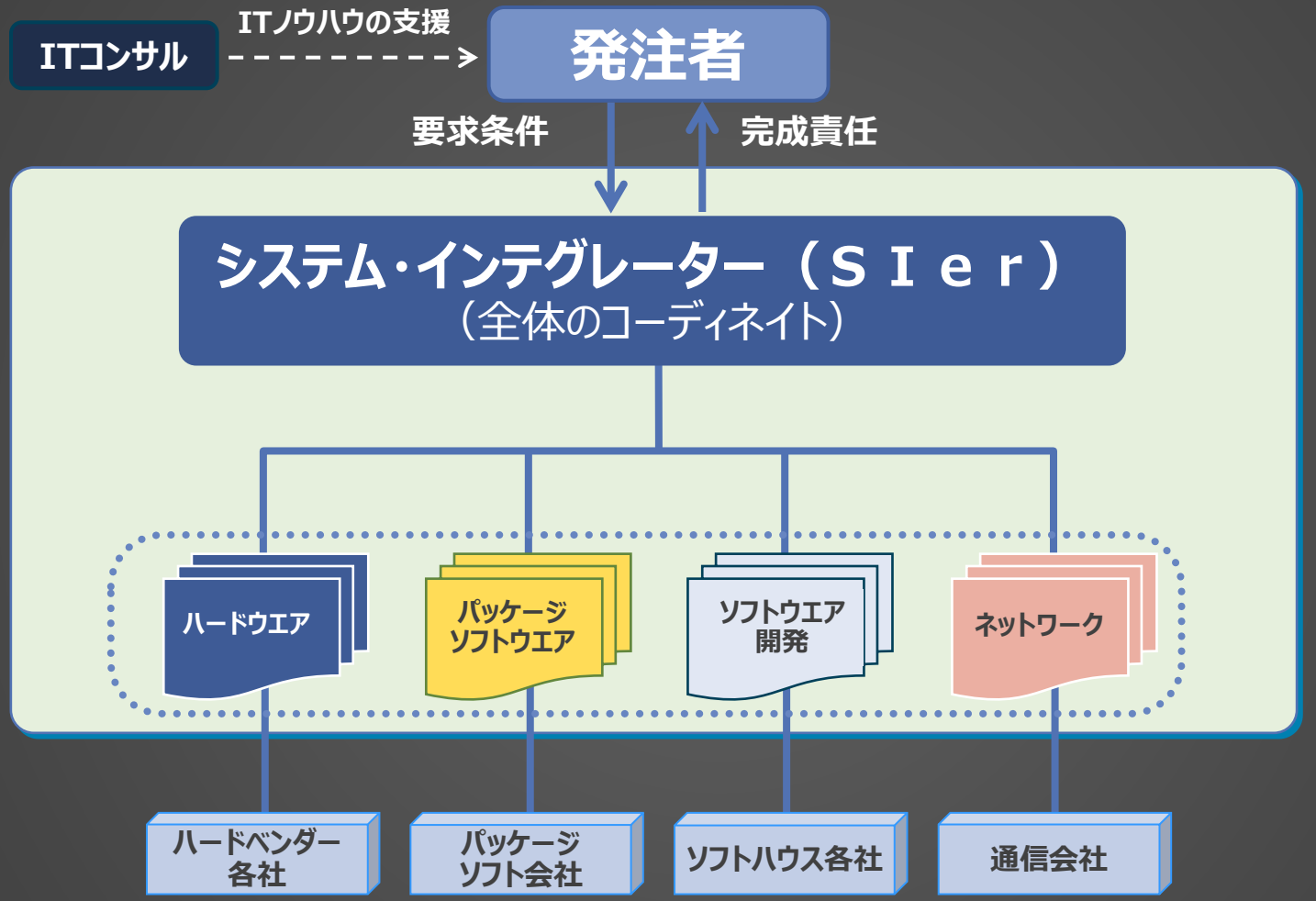
経歴

1. 大規模金融機関向けの勘定系/周辺系システム開発
2. テスト自動化を推進する全社組織
3. 探索的テストの推進
4. テスト・品質に関するプロジェクト支援

↓

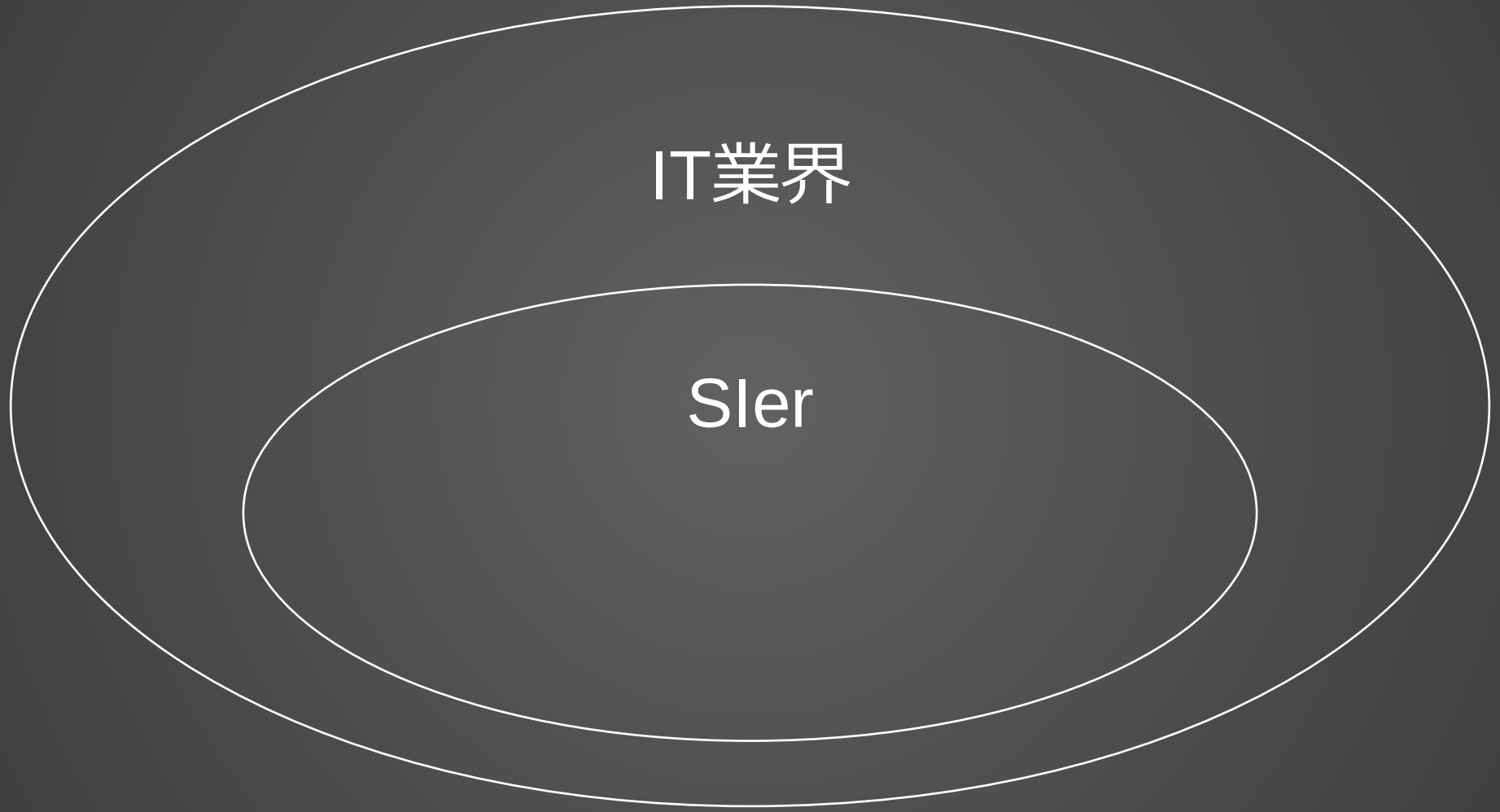
5. そして・・・？

NTTデータ



NTT DATA

IT業界 ≠ SIer



IT業界 ≠ SIer

	利用者	責任	期間
SI	他人	大きい	長い
自社開発	自分/他人	大きい	長い
日曜プログラマ	自分	小さい	短い

SIerのテスト

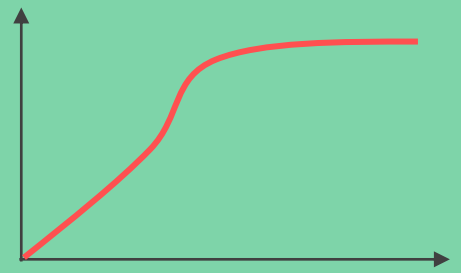
SI事業者のテストでは
テストに可監査性・客観性が求められる

テスト/バグ密度

上限

下限

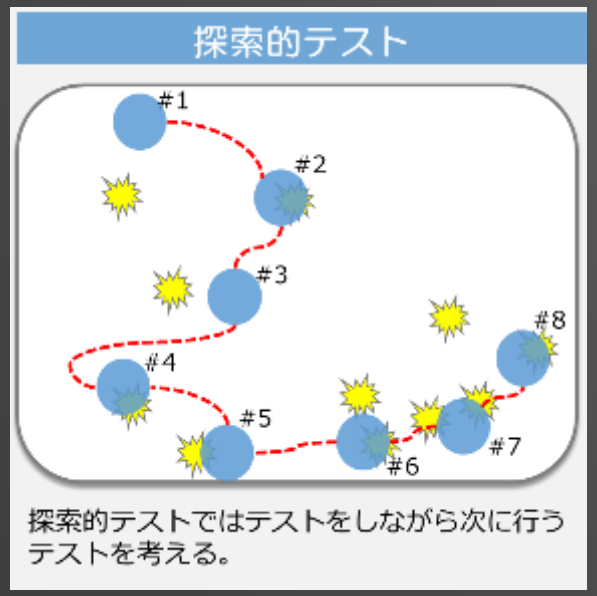
信頼度成長曲線

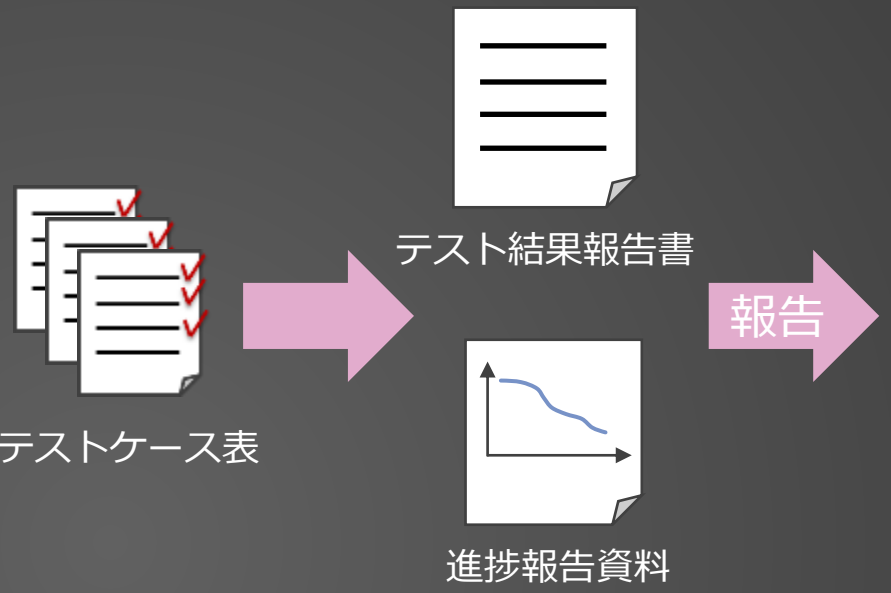


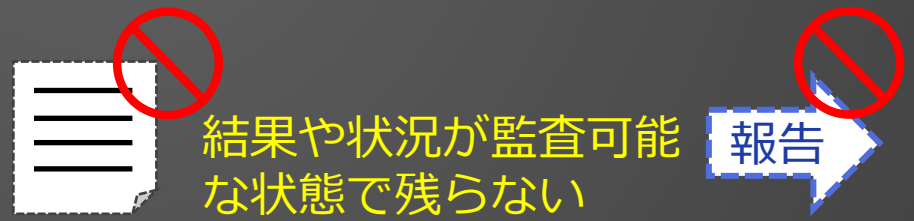
テストケース表

NTT DATA

探索的テスト







NTT DATA

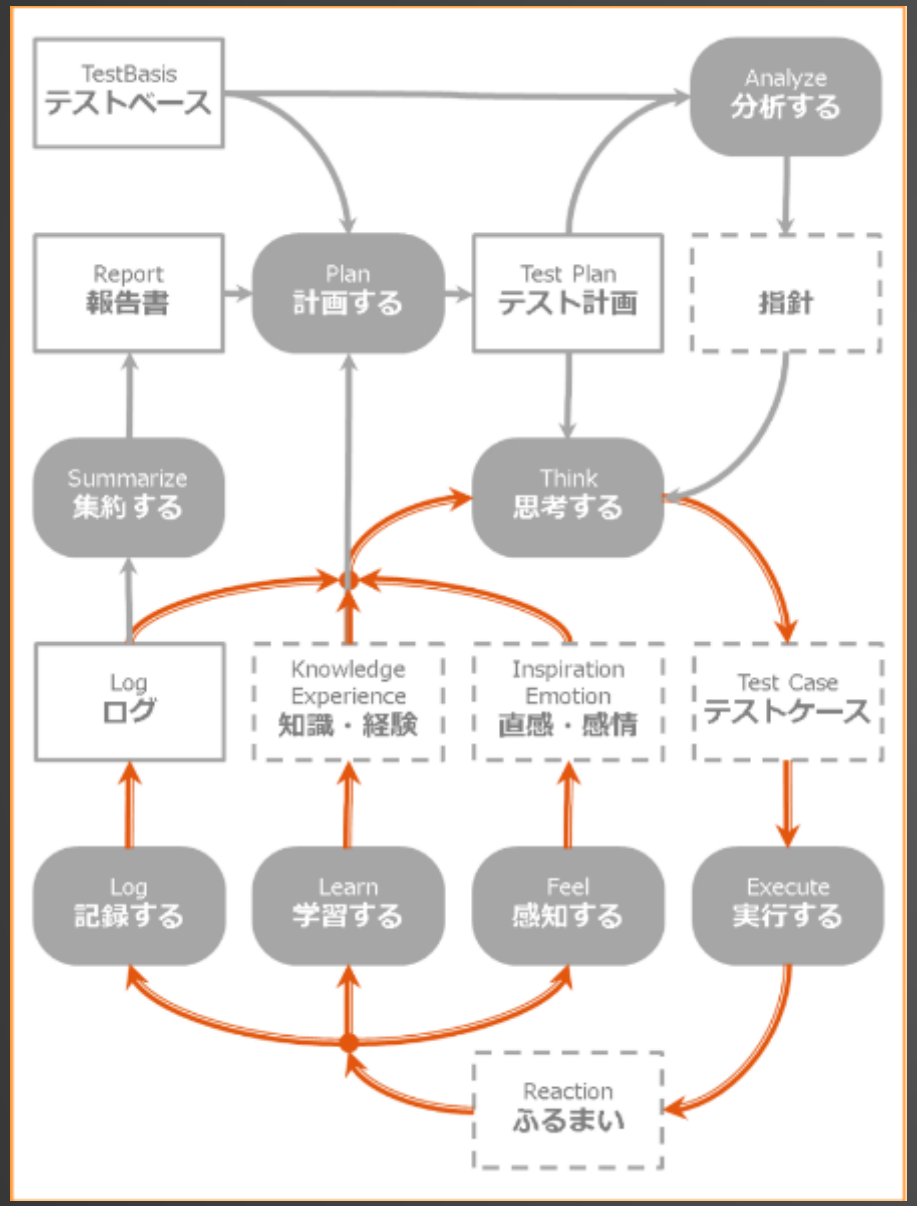
よく似た言葉たち

◆ Ad-hoc Testing
- アドホック（*ad hoc*）は、「特定の目的のための」「限定目的の」などといった意味のラテン語の語句である。*(wikipediaより)*
- その場で思い付いた項目を実行するだけで、テスト計画なしに行われるソフトウェアテストのこと。
- 経験あるテスト技術者によるアドホックテストは、計画的なテストが行き詰っている場合に有益なテストとみなされることがある。これは、伝統的にはエラー推測、近年では探索的テストとも呼ばれる。(某Webメディアより)

◆ Monkey Testing
- プログラムのテスト方法の一つで、猿に実機を渡して無茶苦茶にイベントを発生させ、装置が問題なく動くかどうかを確認するストレステスト。アドホックテストともいう。(某Webメディアより)

◆ Exploratory Testing
- テスト対象として与えられたソフトウェアにおいて起こりそうなバグを推測して、それを検出するテストケースを設計すること。経験ベースのテスト技法に分類される。(某Webメディアより)

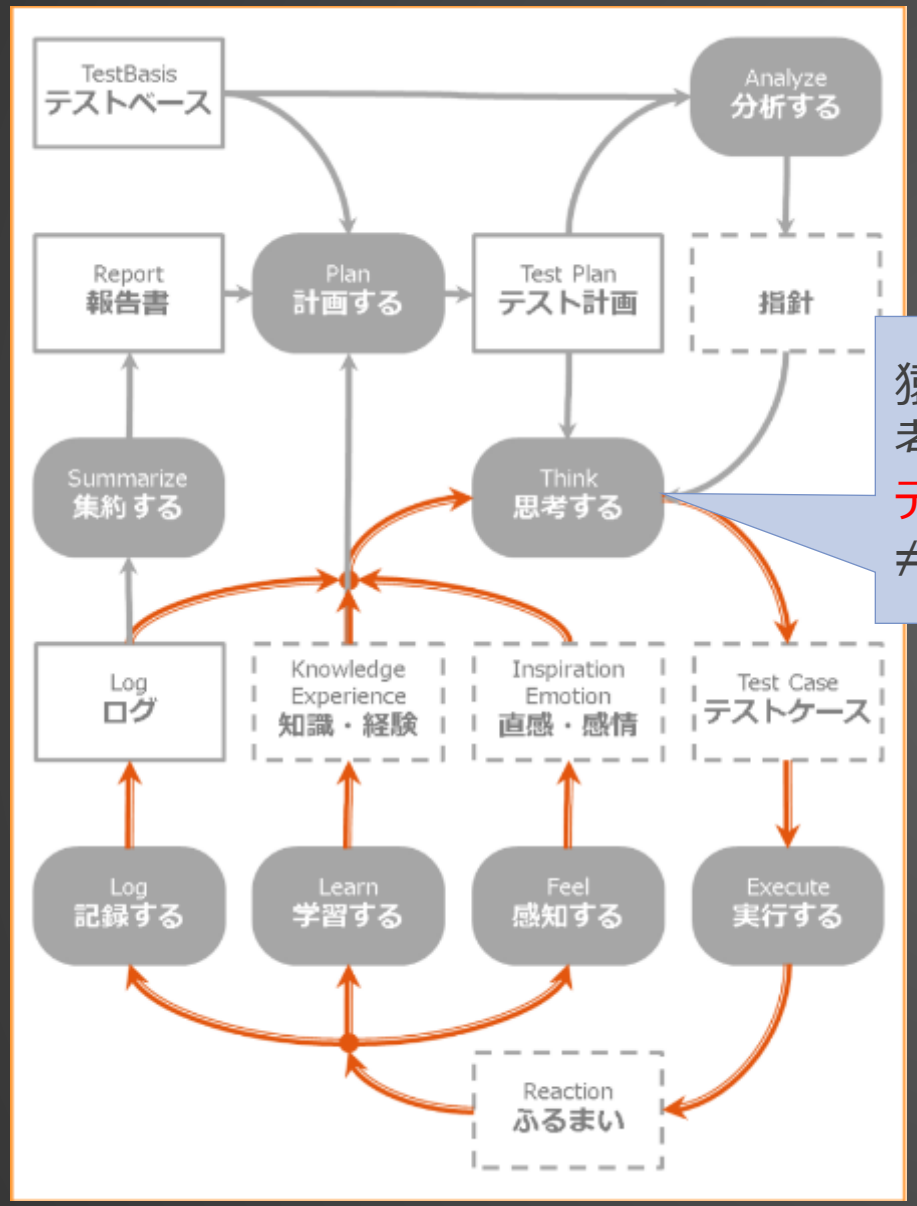
バグを「探索」するということ



https://sites.google.com/site/exploratorytestingjapan/
出典：探索的テストのモデル（探索的テスト研究会）

NTT DATA

バグを「探索」するということ

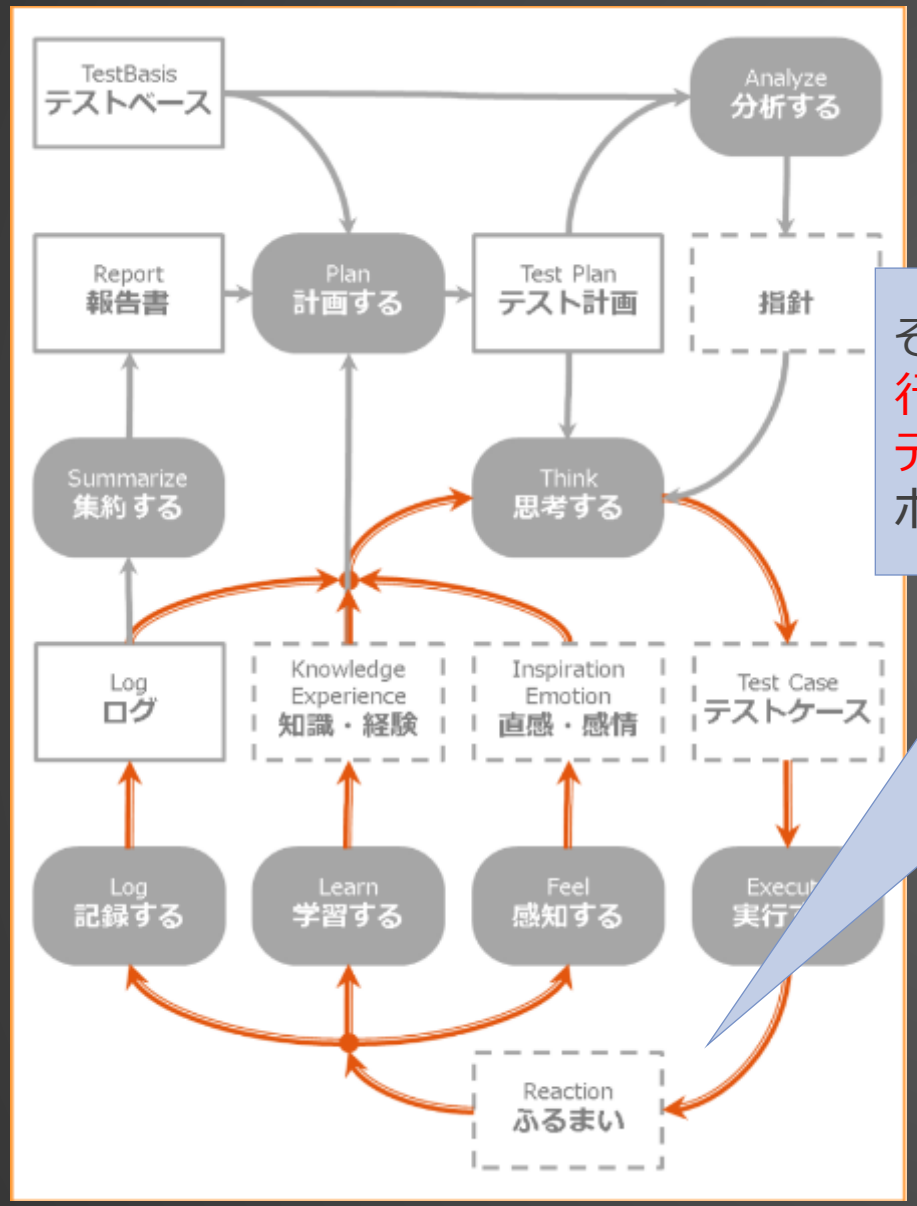


猿のように何も考えず打鍵するのではない。
考えて、頭の中でテストケースを設計して
テストを行うのが探索的テストだ。
≠モンキーテスト

https://sites.google.com/site/exploratorytestingjapan/
出典：探索的テストのモデル（探索的テスト研究会）

NTT DATA

バグを「探索」するということ

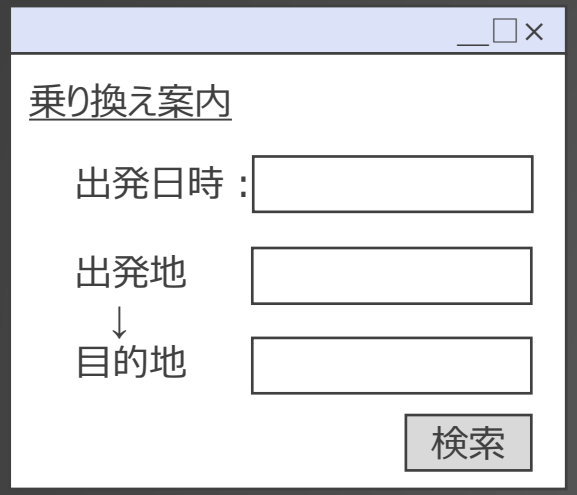


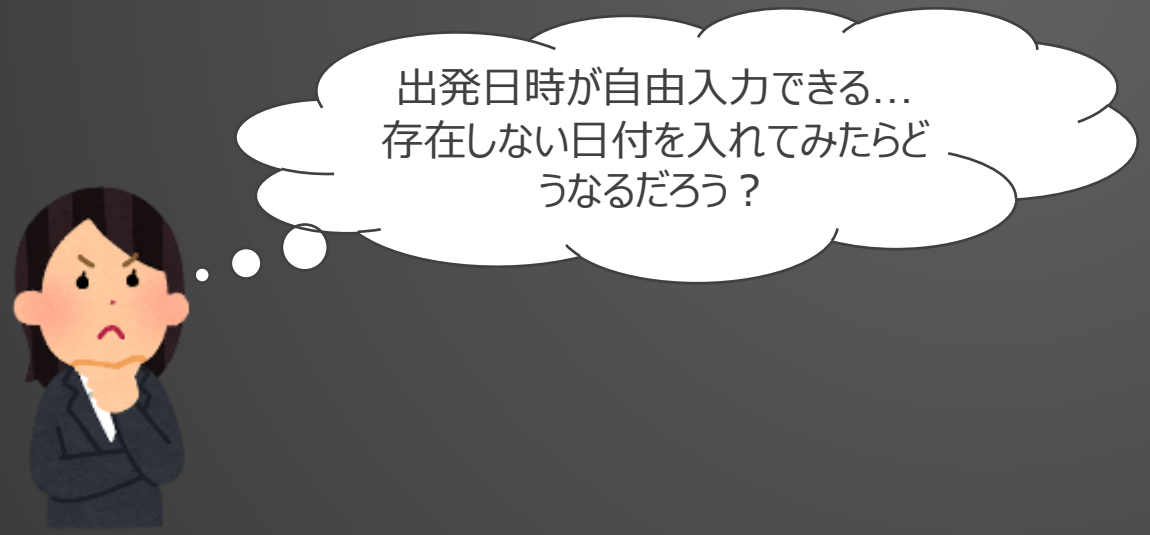
その場で思いついたことをやるだけではない。実行した結果を「観察」して、さらに思考を深めてテストを実行していくのが探索的テストだ。≠アドホックテスト

https://sites.google.com/site/exploratorytestingjapan/
出典：探索的テストのモデル（探索的テスト研究会）

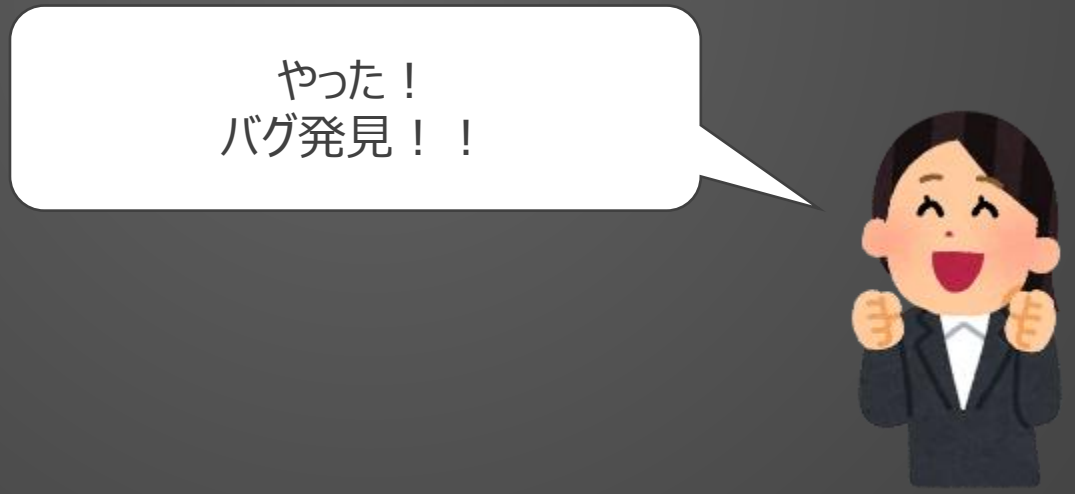
NTT DATA

「探索する」ということ

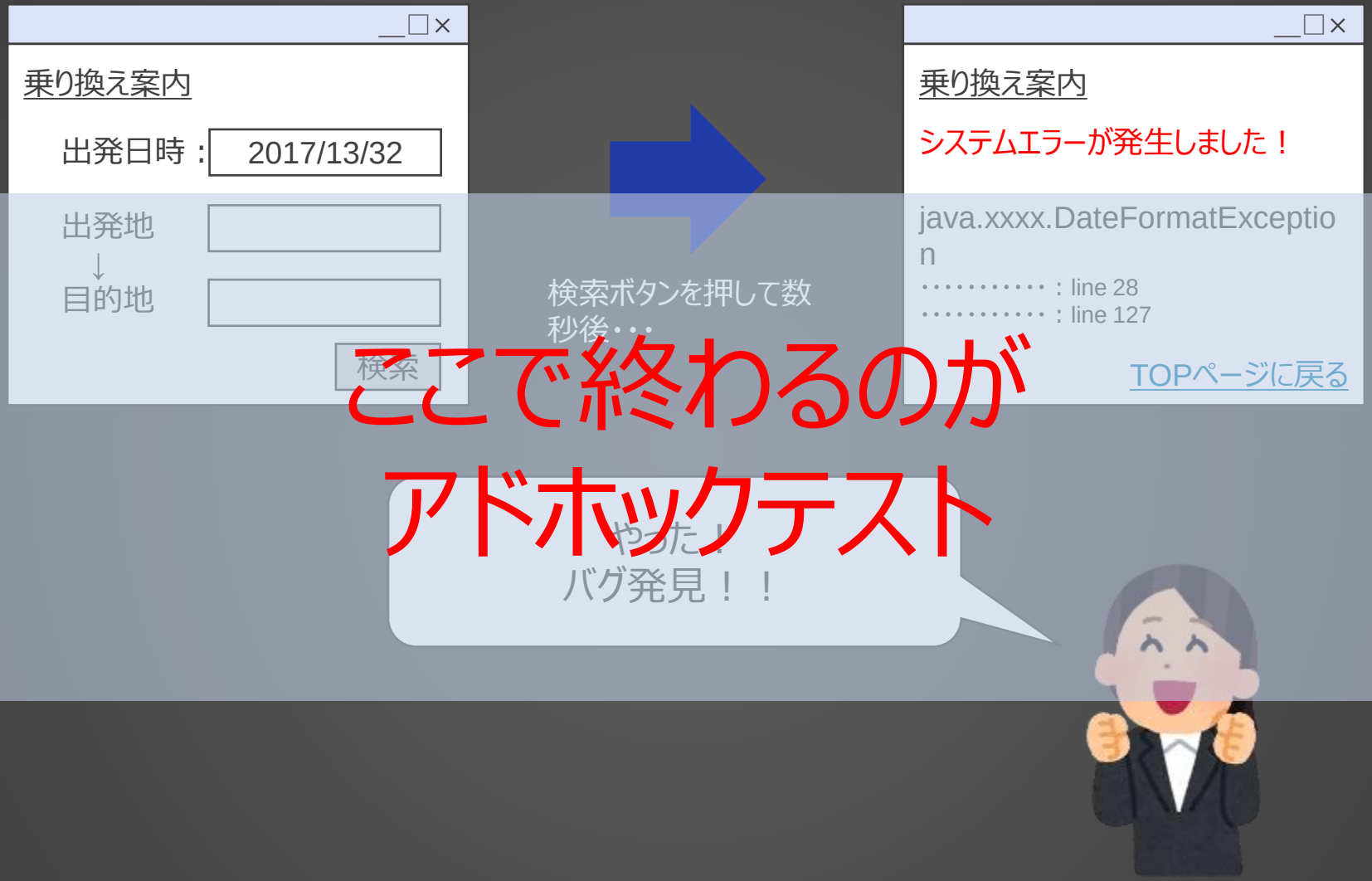




「探索する」ということ



「探索する」ということ

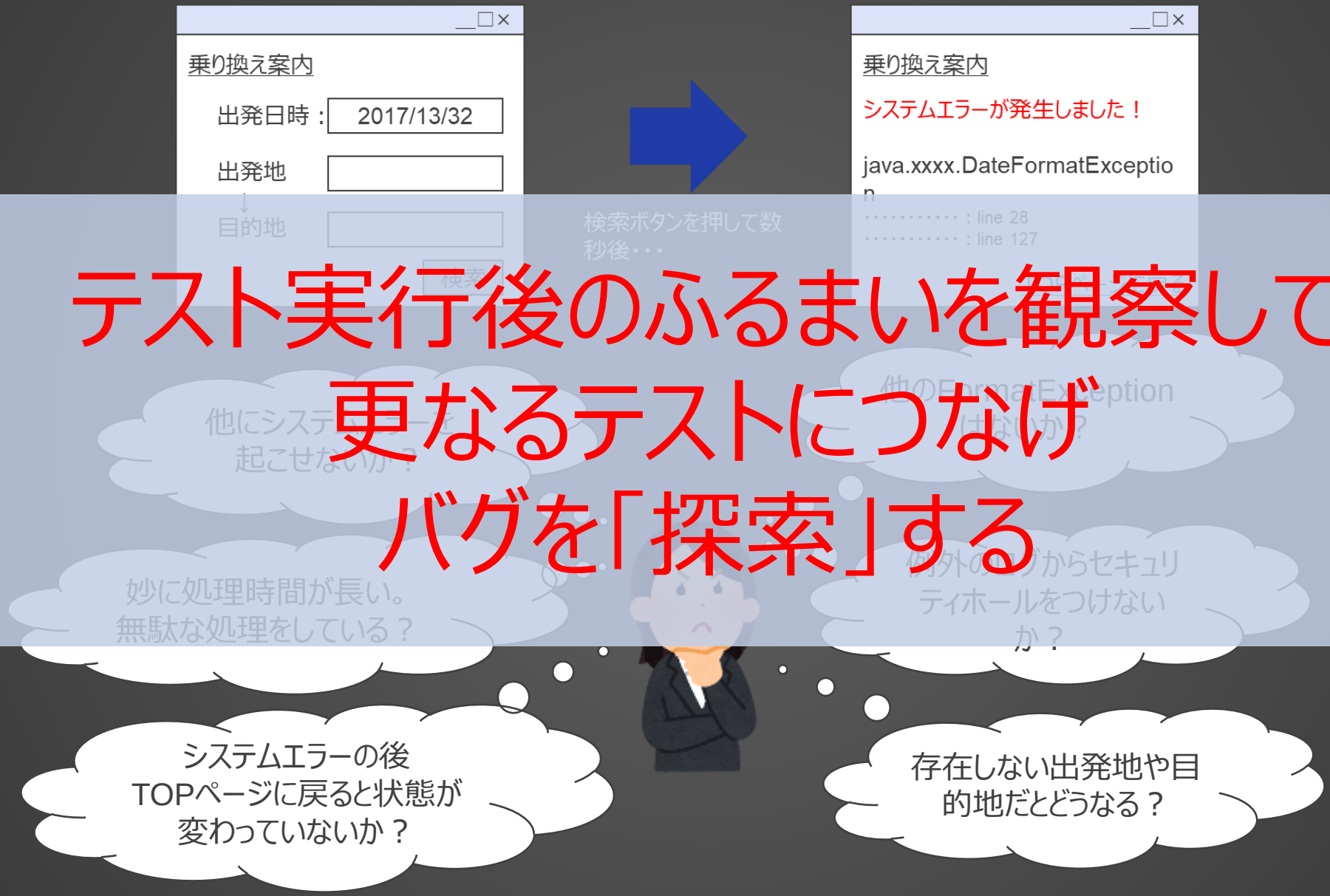


NTT DATA

「探索する」ということ

NTT DATA

「探索する」ということ



テスト実行後のふるまいを観察して
更なるテストにつなげ
バグを「探索」する

NTT DATA

観察する。

観察するのは画面だけじゃない。

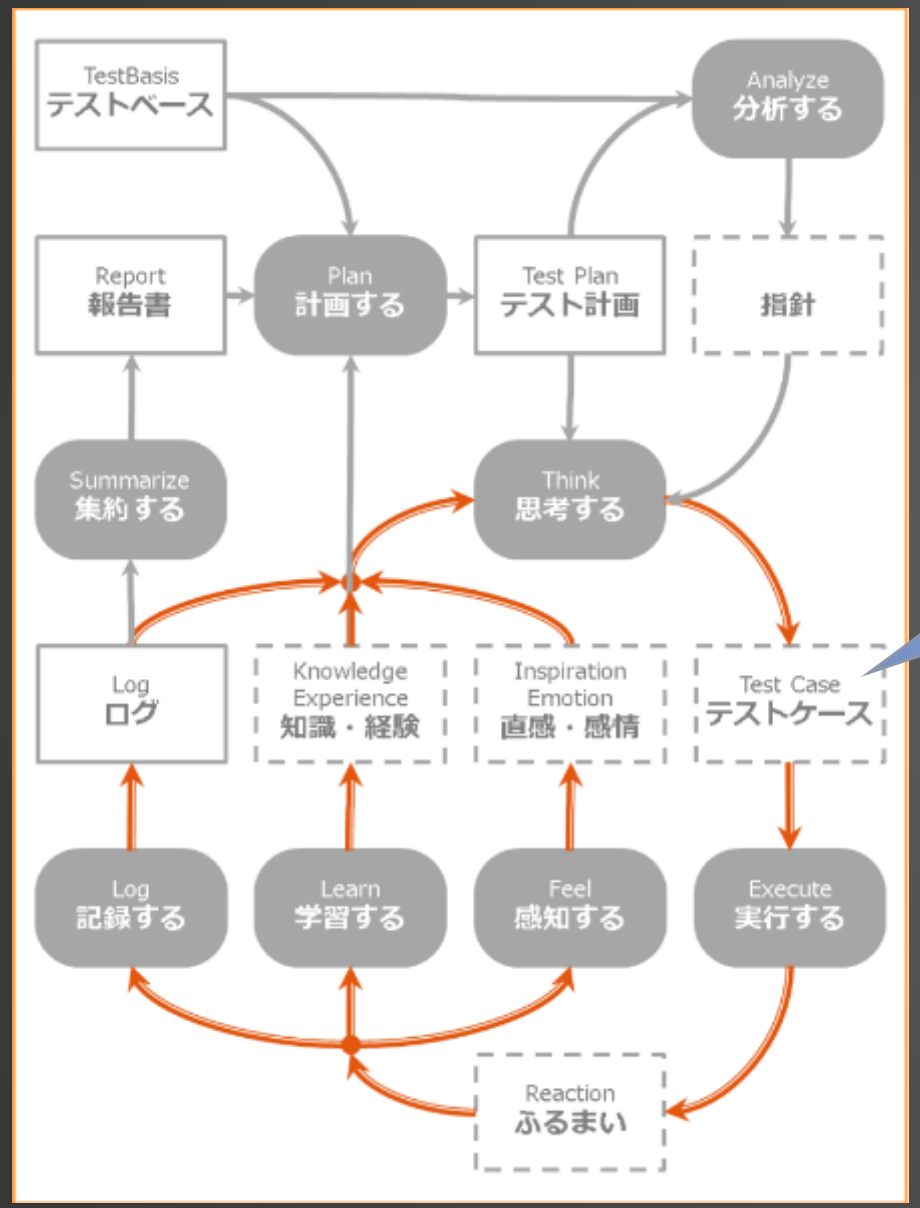
- 帳票
- 各種ログ（デバッグログ含む）
- DBレコード
- 出力ファイル
- 体感的な応答速度　・・・たくさんあるはず！

（こんなことからバグを見つけた人もいる　※特殊な例）

もう操作は終わったのに、ルーターがやけにチカチカと光っている…

操作するとストレージが轟音を立て始める・・・

探索的テスト

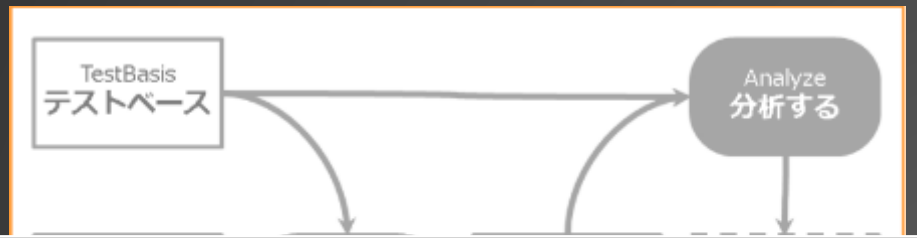


テストケースは無形の成果物

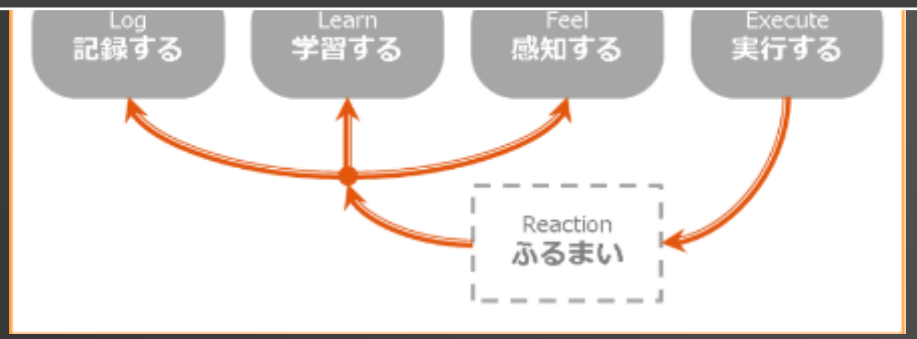
https://sites.google.com/site/exploratorytestingjapan/
出典：探索的テストのモデル（探索的テスト研究会）

NTT DATA

探索的テスト



ん？SIerのテストには客観性が求められるんじゃなかったの？
なんでそんなの推進してんの？



https://sites.google.com/site/exploratorytestingjapan/
出典：探索的テストのモデル（探索的テスト研究会）

NTT DATA

なぜ、探索的テストを始めたのか？

NTT DATA

経歴

1. 大規模金融機関向けの勘定系/周辺系システム開発
2. テスト自動化を推進する全社組織
3. テスト・品質に関するプロジェクト支援
4. 探索的テストの推進
5. そして・・・？

背景その１

頼れる先輩たち

NTT DATA

頼れる先輩たちがモゴモゴするとき・・・。

NTT DATA

長年考え続けている

考え続けて、あきらめに近い感情があった。

- 品質が良い/悪い、品質に問題がないというのを断言するのは「神にも等しい行為」
- テストを十分にやった。と客観的に証明することは不可能
- ましてや「すべてのテスト」をやりました。と言うのは完全なウソだ

品質が良いとか、十分にテストをした。なんて
誰にも証明できないことなのではないか・・・・・・？

背景その２

某金融機関のシステムトラブルに関する報道

XX銀行システムトラブル！！ XX万人に影響！！

※実話

NTT DATA

某金融機関のシステムトラブルに関する報道

XX銀行システムトラブル！！ XX万人に影響！！

※ 実話

コロンブスの卵

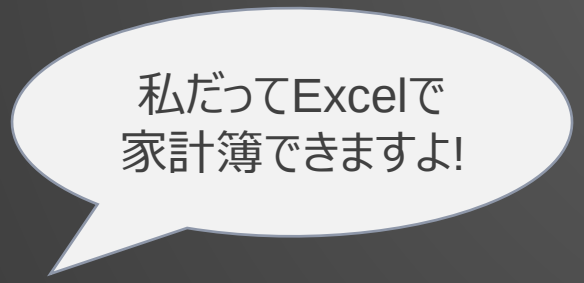
物事が為されたあとは、誰でもその方法を知っている。

こんなこともテスト
してないんですか？

バグが見つかった後は、誰でもその見つけ方を知っている。

銀行の勘定系システムはとても難しい

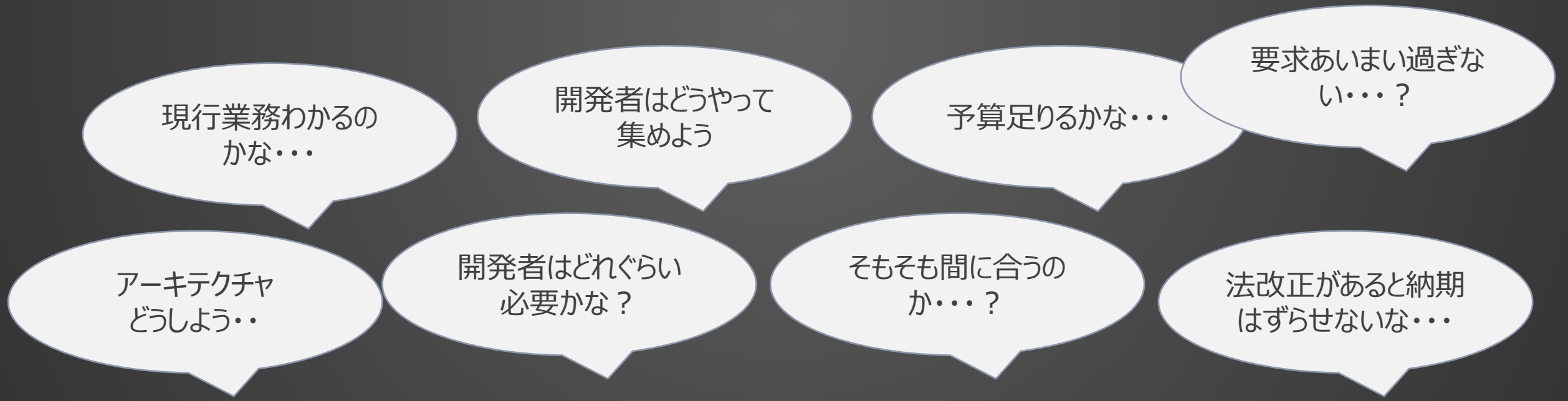
- 24H365D落ちることが許されない超ミッションクリティカルなシステム
- ソフトウェアのバグだけでなく、ハードウェア・ネットワークの故障時も落ちることは許されない
- 当然ながら個人だけでなく法人口座もあり、取り扱う口座の数分、莫大なデータ量が存在する
- 個人の利用（引出、預入など）だけでなく、法人の利用（給与振り込みなど）もあり、一時的なエラーやトラフィック増があったとしても、全てのトランザクションを矛盾なく取り扱う必要がある
- そのために2重・3重・4重にトランザクションを保証する仕組みを運用している。
- 「ATMで引き出しが行われている最中に、公共料金の引き落とし処理が行われていて、順序を守らないと残高がマイナスになる。しかも引出しの処理中にネットワーク故障が発生。」
- などなど



非常に難しいシステムを開発している。
が、ユーザから見ると、一見して簡単そうに見えてしまう。

そもそもシステム開発自体が難しすぎる

あなたはXX省〇〇システム更改PJのプロジェクトマネージャです。
〇〇システムはサービス開始から１０年が経過し、その間に実施されたバグの修正や、機能追加によって
複雑な構造になっており、維持費用が高額になっていました。
本プロジェクトは、〇〇システムを更改することにより、維持費用を低減させることを目的とします。
また、ユーザに優しいインタフェースに変更し、業務効率を高めるとともに、20XX年に予定される法改正に対応
することも目指します。開発期間は１年。予算はＸ億円です。



NTT DATA

そもそもシステム開発自体が難しすぎる

そもそもシステム開発自体が難しすぎる

NTT DATA

その割に、断言する人が多い

プロジェクトの成否を決めるのは ずばり「アーキテクチャ」だね！

勝ちに不思議の勝ちあり
負けに不思議の負けなし

By 野村克也さん
From 松浦静山の剣術書『剣談』

- みんな、自分の成功体験から簡単に結論を出しすぎている。
- 多くの場合、問題はそれほど簡単なものではない。
- 質が悪いのは、プロジェクトの成否にかかわる要因があまりに多岐にわたるため、おおよそ思いつきで言われたことであっても、大体は当てはまってしまう。
- その結果、難しい問題がまるで簡単な問題のようにとらえられてしまう。
- 事情や背景、知識や経験がない人ほど、この傾向は強い。
- 多少賢い人は、要因の重みづけを考え始め、決定因子を特定しようとする。
- が、あまりにも条件が多岐にわたるため決定因子が特定できない。

本来、とても難しい問題を
簡単な問題に置き換えてしまうことで
論理的に解決した気になっている。

本来、とても難しい問題を
簡単な問題に置き換えてしまうことで
論理的に解決した気になっている。

論理的に解かないという選択肢はないのか？

なぜ、探索的テストを始めたのか？

NTT DATA

背景１＋背景２

品質が良いとか、十分にテストをした。なんて
誰にも証明できないことなのではないか・・・・・・？

論理的に解かないという選択肢はないのか？

もっと現実的な効果を積み上げていくほうが
いいのでは？

テストを現実的に考えてみる

テストをすることによって得られる
もっともわかりやすい効果はなんだろう？

↓

品質がよくなる

テストを現実的に考えてみる

テスト
もっと
やすい効果はな
ろう？

NTT DATA

テストを現実的に考えてみる

テストをすることによって得られる
もっともわかりやすい効果はなんだろう？

↓

バグが見つかること

仕様です

テストを現実的に考えてみる

テストをすることによって得られる
もっともわかりやすい効果はなんだろう？

↓

バグが見つかること
悪い仕様も見つけること

探索的テストの効果

分類	欠陥摘出能力(件/時間)
平均値	1.214
中央値	0.925

開発分類	探索的テスト(件/時間)	スクリプトテスト(件/時間)
スマホアプリ	1.726	0.137
Webシステム	1.112	0.085
組み込み機器	0.717	0.072

ID	プロジェクト概要	欠陥摘出能力(件/時間)
A	Androidスマートフォンアプリの開発	0.696
B	金融機関向けスマートフォンアプリの開発	1.129
C	通信企業向けスマートフォンアプリの開発	3.389
D	音声認識ロボットのソフトウェア開発	0.545
E	スマートホーム機器のソフトウェア開発	1.187
F	公共団体向けWebシステムの開発	1.605
G	企業人事向けWebシステムの開発	0.440
H	金融機関向けWebシステムの開発	0.455

探索的テストの効果

分類	仕様改善提案能力(件/時間)
平均値	0.880
中央値	1.111

開発分類	探索的テスト(件/時間)	スクリプトテスト(件/時間)
スマホアプリ	1.043	0.018
Webシステム	0.723	0.009
組み込み機器	1.817	0.011

ID	プロジェクト概要	仕様改善提案能力(件/時間)
A	Androidスマートフォンアプリの開発	0.625
B	金融機関向けスマートフォンアプリの開発	1.944
C	通信企業向けスマートフォンアプリの開発	0.574
D	音声認識ロボットのソフトウェア開発	1.931
E	スマートホーム機器のソフトウェア開発	1.500
F	公共団体向けWebシステムの開発	0.296
G	企業人事向けWebシステムの開発	1.391
H	金融機関向けWebシステムの開発	0.500

１０倍近いバグの摘出能力
１００倍近い仕様改善提案能力

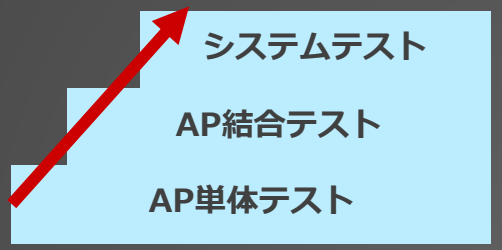
↓

なんて現実的な効果

どんなテストをしたほうがいいのか。
どうやれば品質の良さ・悪さを証明できる？

解けないパズルを解こうとするよりも
現実的に効果が出るソリューションを求めたい。
探索的テストを使っていこう。

公式なテスト



- 事前にテストケースを作成し、要件を順番通りに検証する
- 再現性／客観性／可監査性に優れる
- テスト1回あたりに必要な工数が大きい

非公式なテスト

こっち

- テストケースは事前に作成せず、検証対象を特に定めない
- 再現性／客観性／可監査性に乏しい
- テスト1回あたりに必要な工数が小さい

NTT DATA

探索的テストの活用ポイント

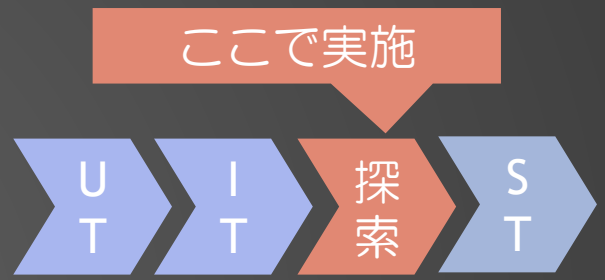
工数削減したい

- 結合テストの前に探索的テストでバグを出すことで、後工程のテストで見つけるよりも**少ない工数でバグを修正**できる。
- 運用対処やUI改善が実施でき、**顧客満足度が向上**する。
- UT以降継続して実施しても良い。

商用バグのすり抜けを減らしたい

- STまで**すり抜けてしまったバグを探索的テストで抽出**できる。
- 設計書からテストを実施しないため、設計書からテストケースを作成する**従来のテストとは違う観点のテストができる**。

効果的な受入テストがしたい



- 受入テストの時点で、**後工程で見つかるバグが摘出**できる。
- 請負ベンダと違う観点で**効果的な受入テストができる**。

※上記はITまで委託した場合

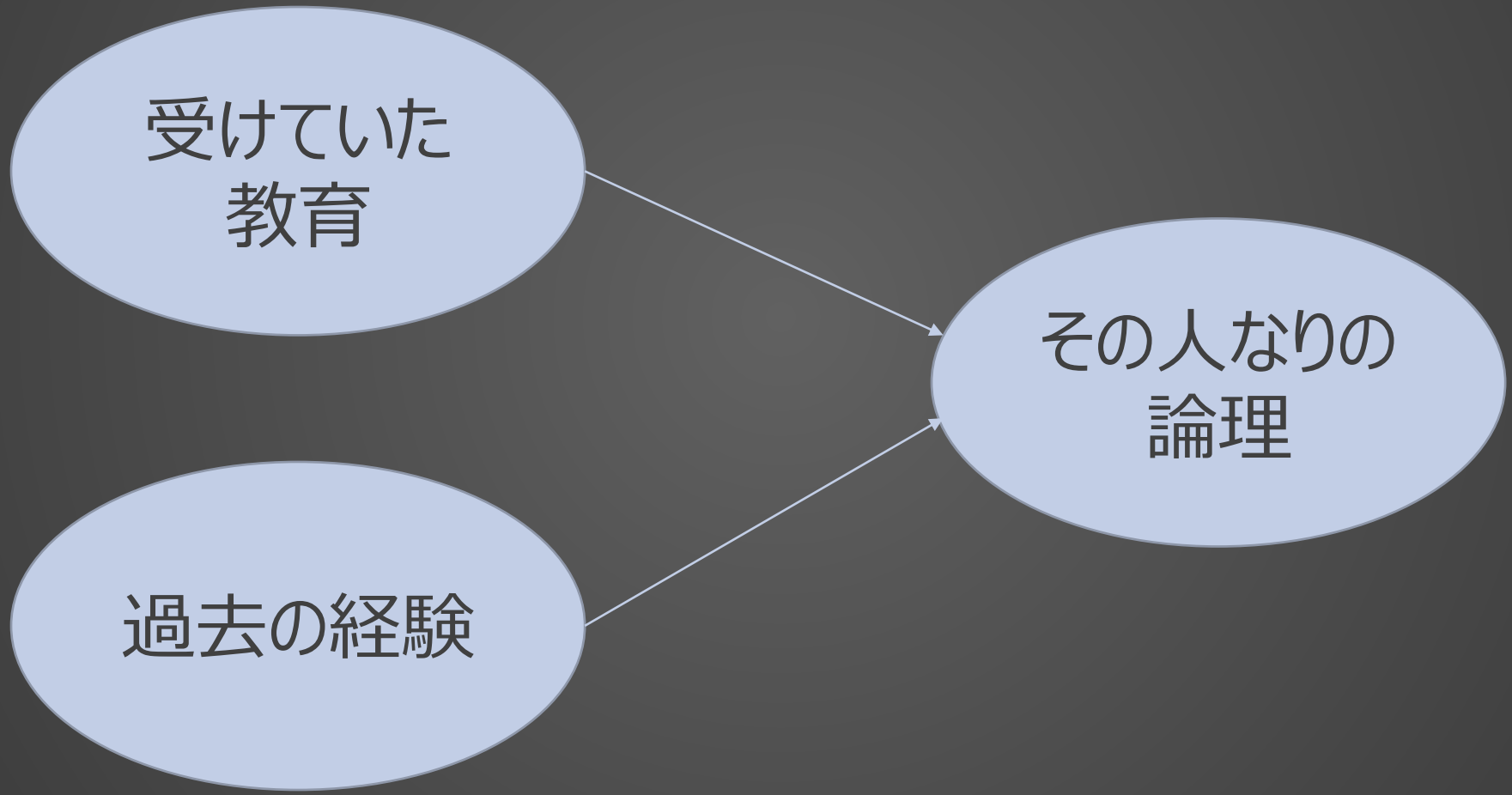
NTT DATA

社内での最初の評価

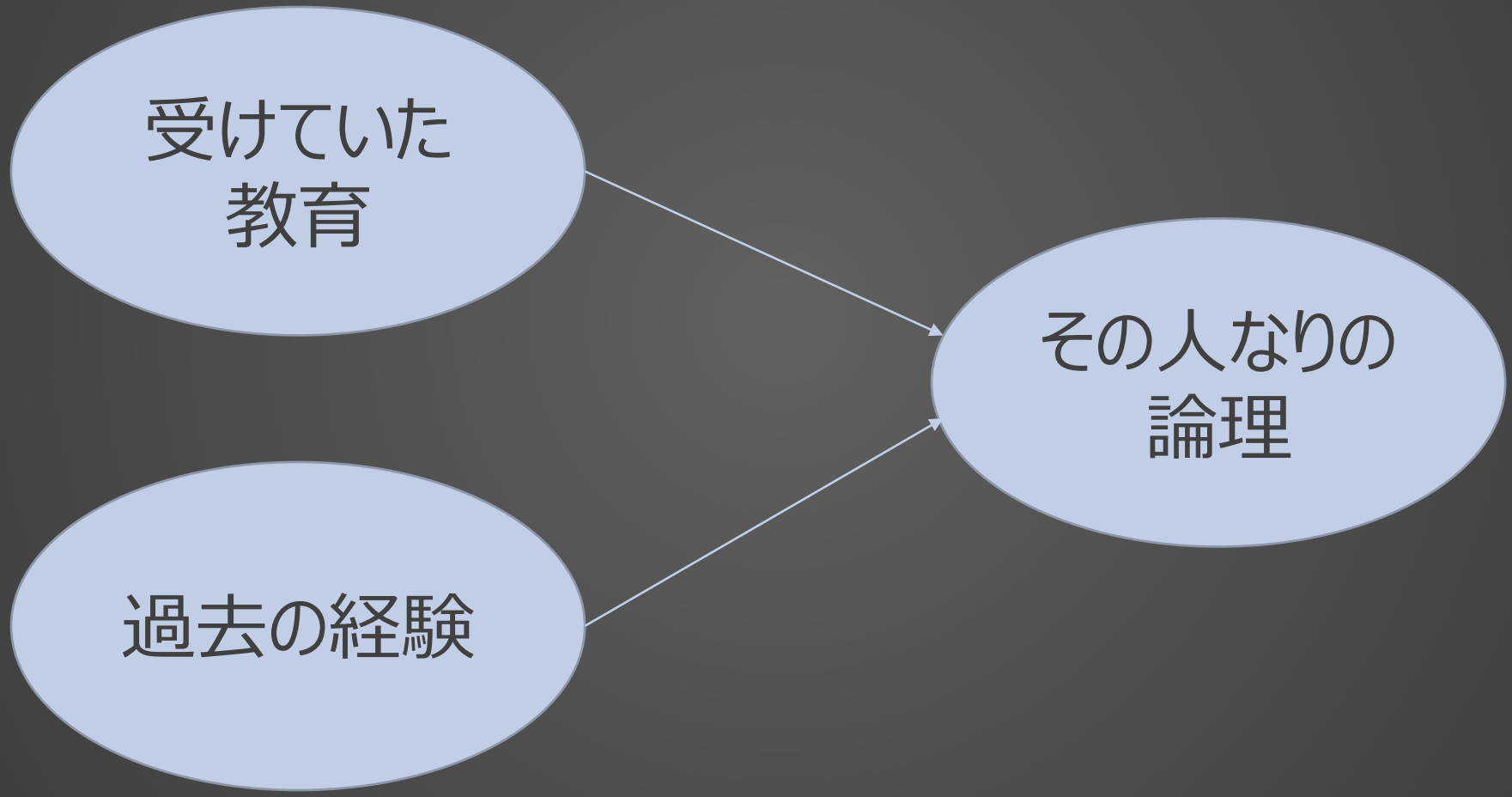
- 本当にバグの摘出効率は高いのか？
- 結局公式なテストをすることになるのであれば、無駄な投資になるのではないか？
- 多忙なプロジェクトの中で、非公式なテストに対して投資できる工数がない。

強い抵抗を持つ人＝論理的

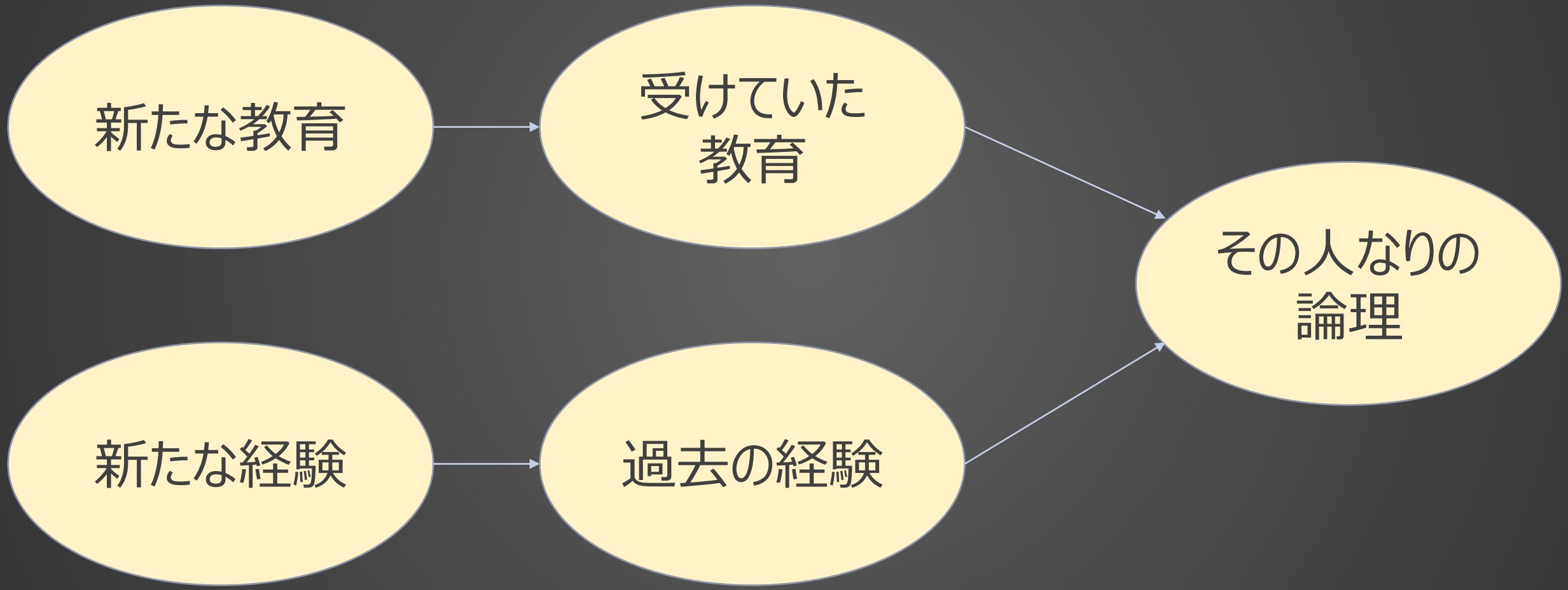
ただし、その論理はその人は受けていた教育や、過去の経験に依存したものである。



「論理」にどうやって立ち向かうのか？

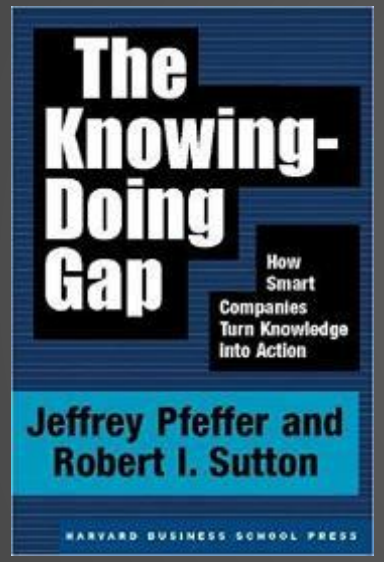


新しい「論理」を作ってゲームチェンジする



NTT DATA

知識と経験を充足するような研修を作る

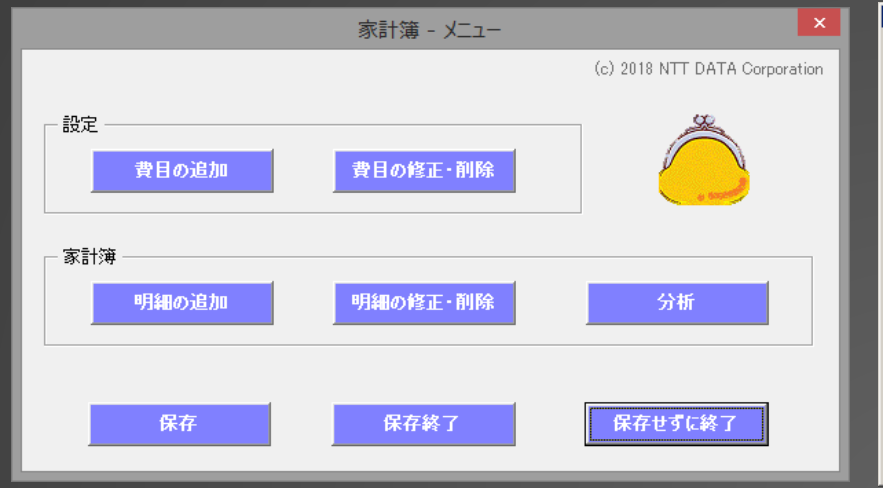


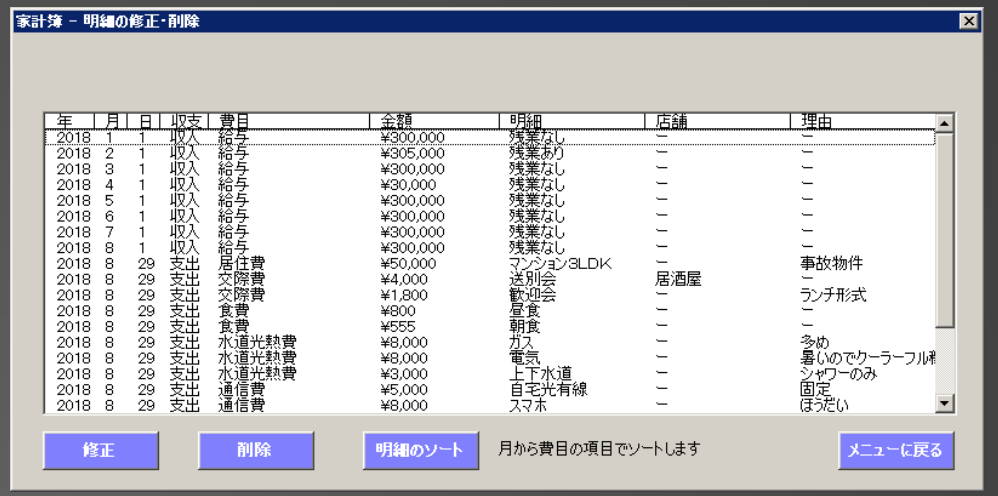
キーワードは ”Knowing-Doing Gap”

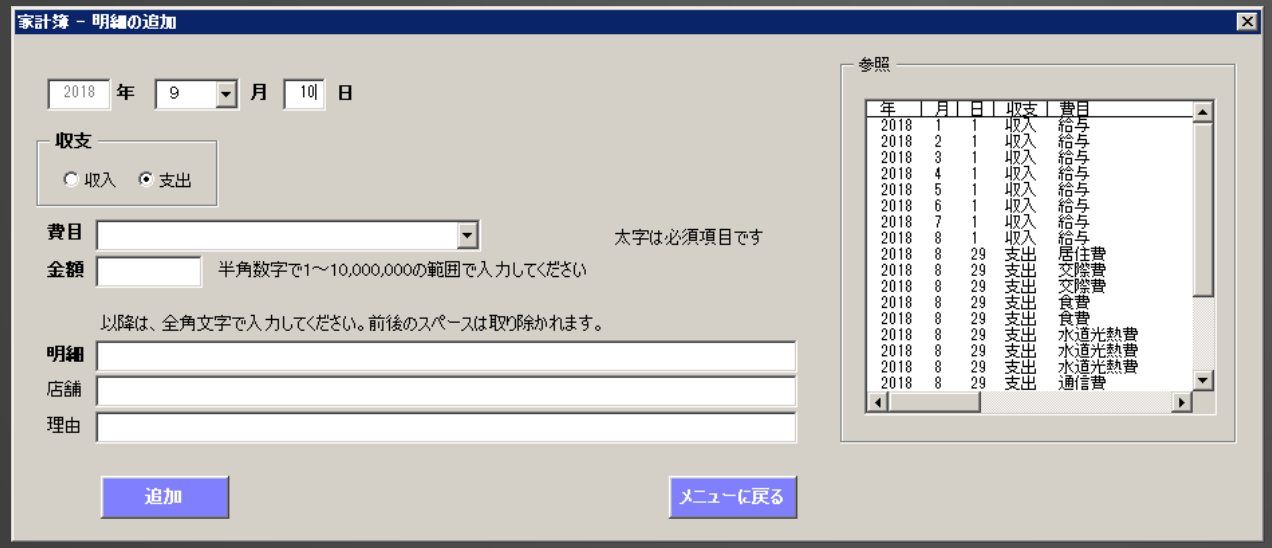
知っていてもできない

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

意図的にバグを混入させたアプリケーション



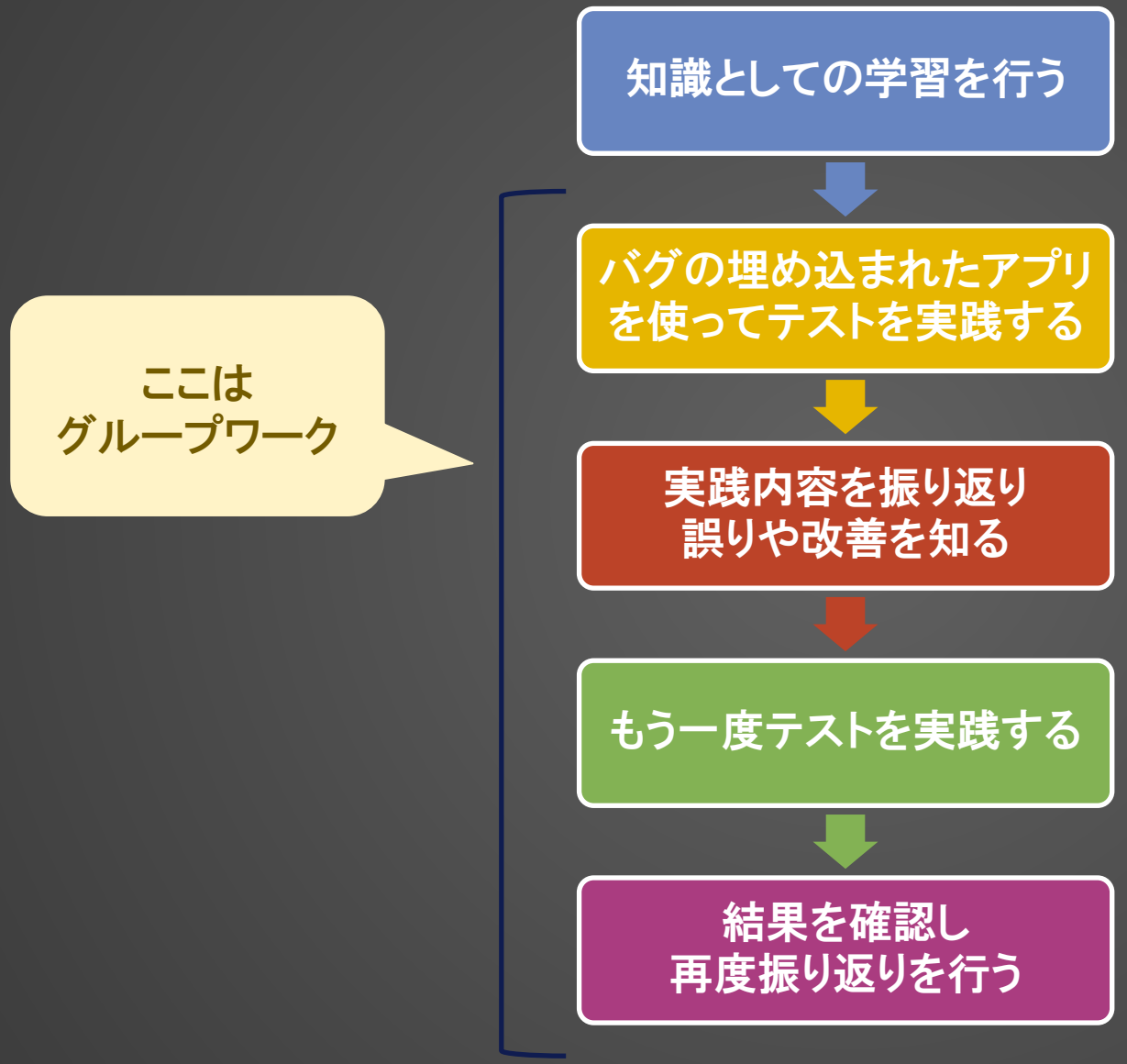




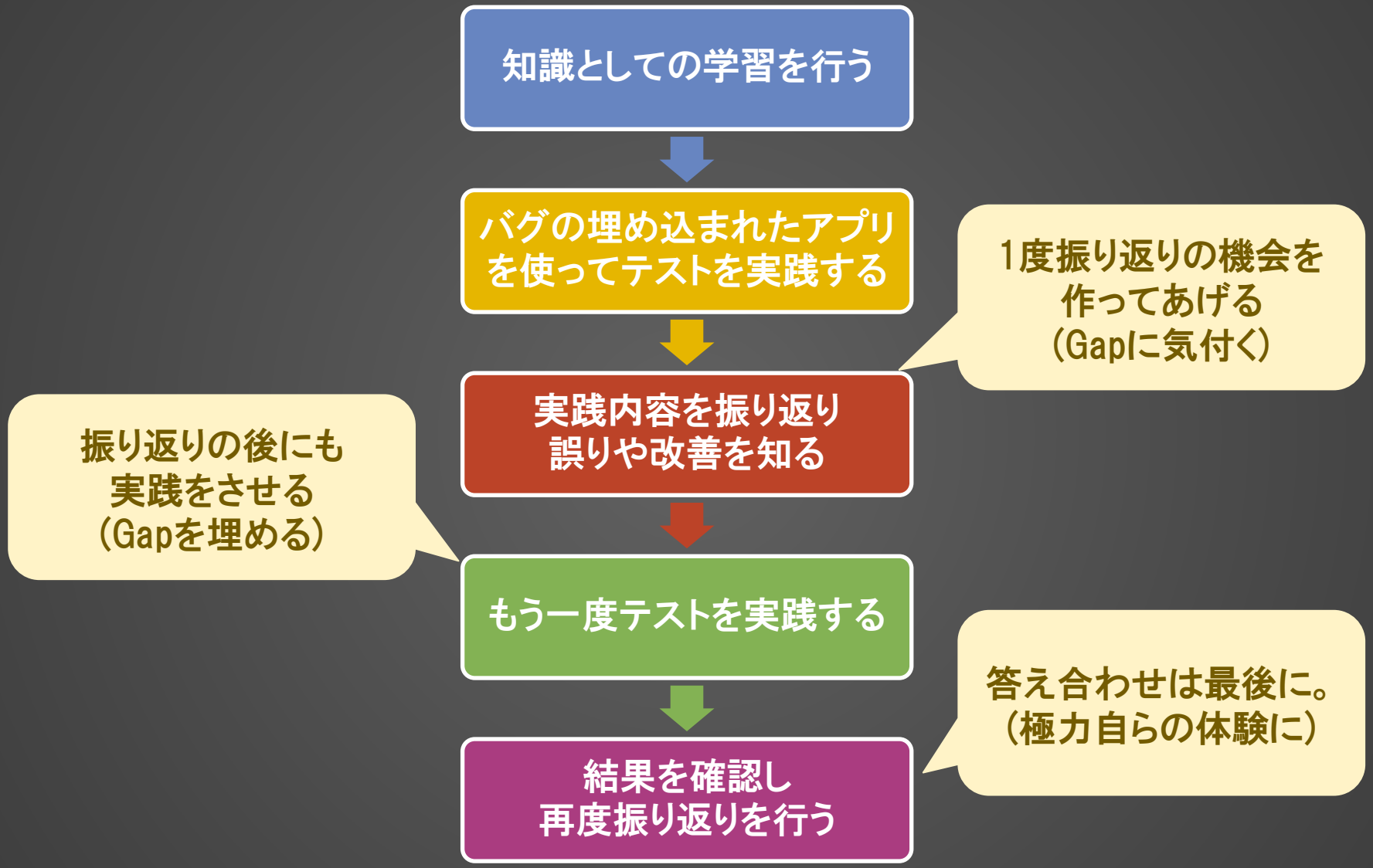
実際の開発を想起させるためのドキュメントや議事録なども用意。

プロジェクト計画書
レビュー記録　など…

知識と経験を充足するような研修を作る



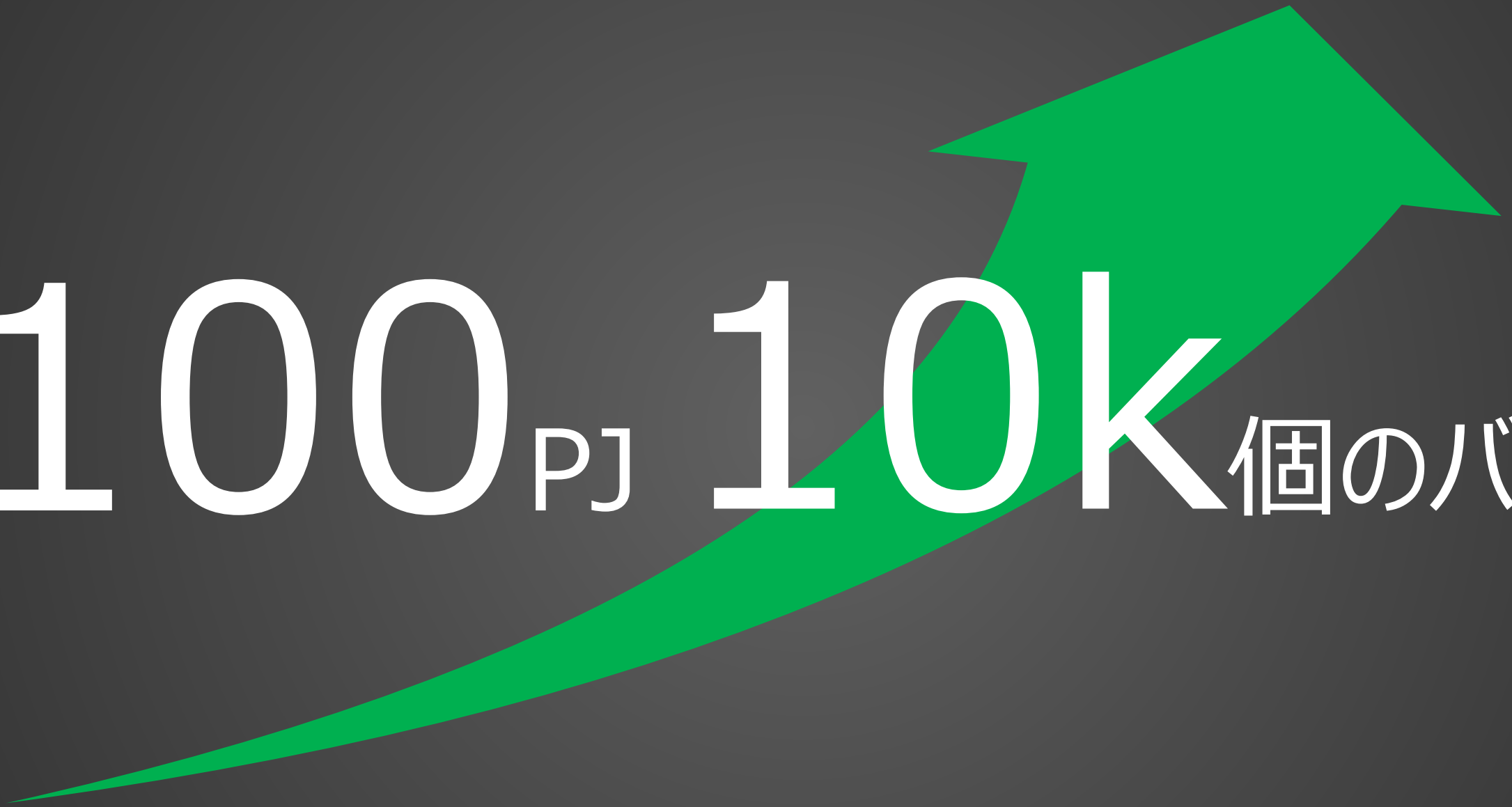
知識と経験を充足するような研修を作る



論理を変えれば、反応も変わる

- 本当にバグの摘出効率は高いのか？
- 結局公式なテストをすることになるのであれば、無駄な投資になるのではないか？
- 多忙なプロジェクトの中で、非公式なテストに対して投資できる工数がない。

- こんな短期間でバグが見つかるならやってみよう
- 先にバグを見つけたほうが、後で見つかるよりいいよね

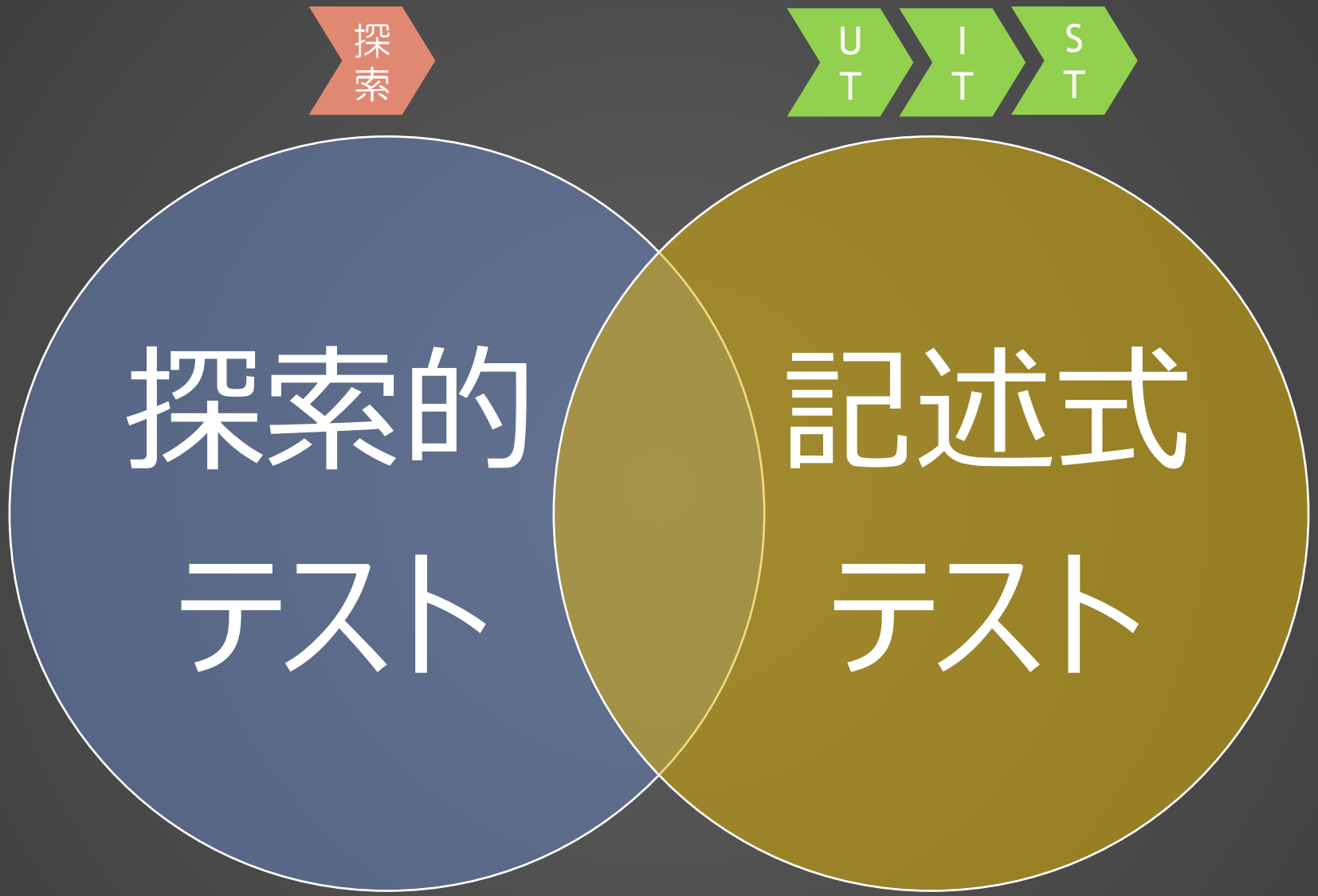


現実的な効果

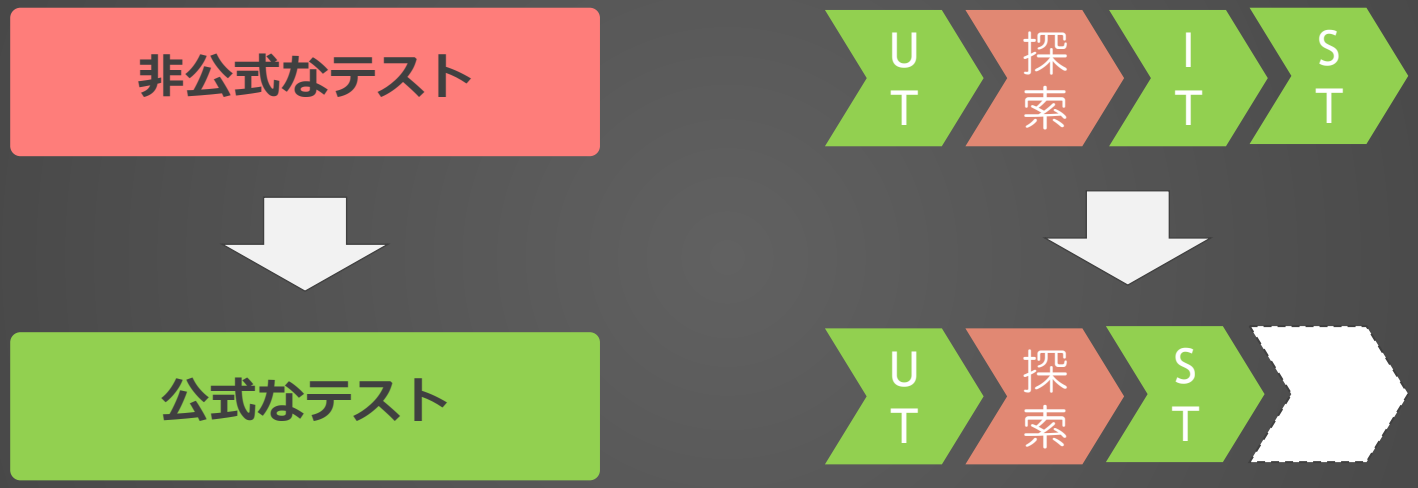
いま、進めていること
その１

経歴

1. 大規模金融機関向けの勘定系/周辺系システム開発
2. テスト自動化を推進する全社組織
3. テスト・品質に関するプロジェクト支援
4. 探索的テストの推進
5. そして・・・？



NTT DATA

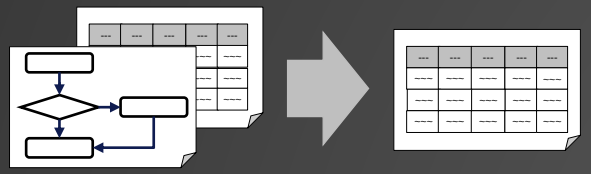


記述式テストと探索的テストの良いとこ取りがしたい

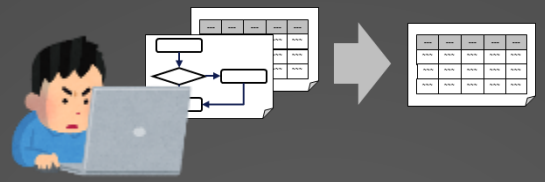
アプローチ	客観性	効率
記述式テスト	○	×
探索的テスト	×	○

アプローチ	客観性	効率
新しい手法	○	○

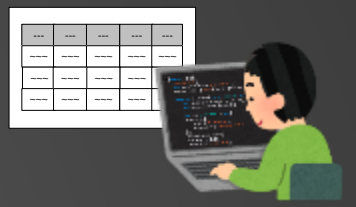
記述式テスト



テストベースからテスト設計する



テスト設計成果物を
レビューして抜け漏れを確認する

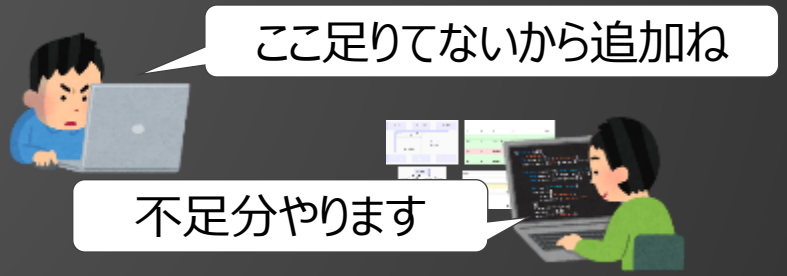


テストする

SONAR Testing

まずテストする

テスト中に取得されたモデルを
レビューして抜け漏れを確認する



抜け漏れを補うようにテストする

NTT DATA

時間があったらデモ

いま、進めていること
その２

経歴

1. 大規模金融機関向けの勘定系/周辺系システム開発
2. テスト自動化を推進する全社組織
3. テスト・品質に関するプロジェクト支援
4. 探索的テストの推進
5. そして・・・？

開発者・技術者への尊敬/敬意

あらゆるモノづくりを担う人たちへの尊敬
当社は、当社とお付き合いのある技術者の皆様によって支えられている

技術者の皆様が

- もっとパワーを発揮できるようにする
- もっと楽しく仕事ができるようにする

今は、これが自分の使命だと言い切れる。

日本のSI業界のゲームチェンジ

- 「品質に問題ない」
- Excel方眼紙
- PowerPoint設計書
- 何図かわからない便利すぎる図
- ソースコードと情報量の変わらないフローチャート
- 画面キャプチャ証跡ぺたぺた

日本のSI業界のゲームチェンジ

「品質に問題ない」

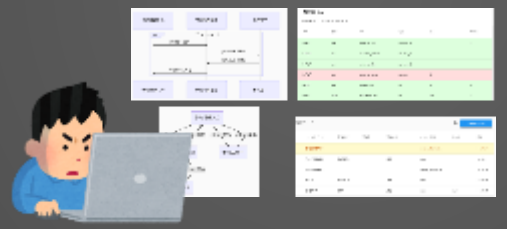
もっと“現実的に”変える

- 画面キャプチャ証跡ぺたぺた

技術者の皆様が

- もっとパワーを発揮できるようにする
- もっと楽しく仕事ができるようにする

今は、これが自分の使命だと言い切れる。



テスト中に取得されたモデルを
レビューして抜け漏れを確認する

技術者の皆様が

- もっとパワーを発揮できるようにする
- もっと楽しく仕事ができるようにする

皆様や、周りの技術者にも

テスト中に取得されたモデルを
レビューして抜け漏れを確認する

NTT DATA

ずっと、考えていること

NTT DATA

経歴

1. 大規模金融機関向けの勘定系/周辺系システム開発
2. テスト自動化を推進する全社組織
3. テスト・品質に関するプロジェクト支援
4. 探索的テストの推進
5. そして・・・？
6. そして、そして・・・？

境界値分析

- 境界にはバグが潜む。
- 等符号の使い方を誤ったりするから。

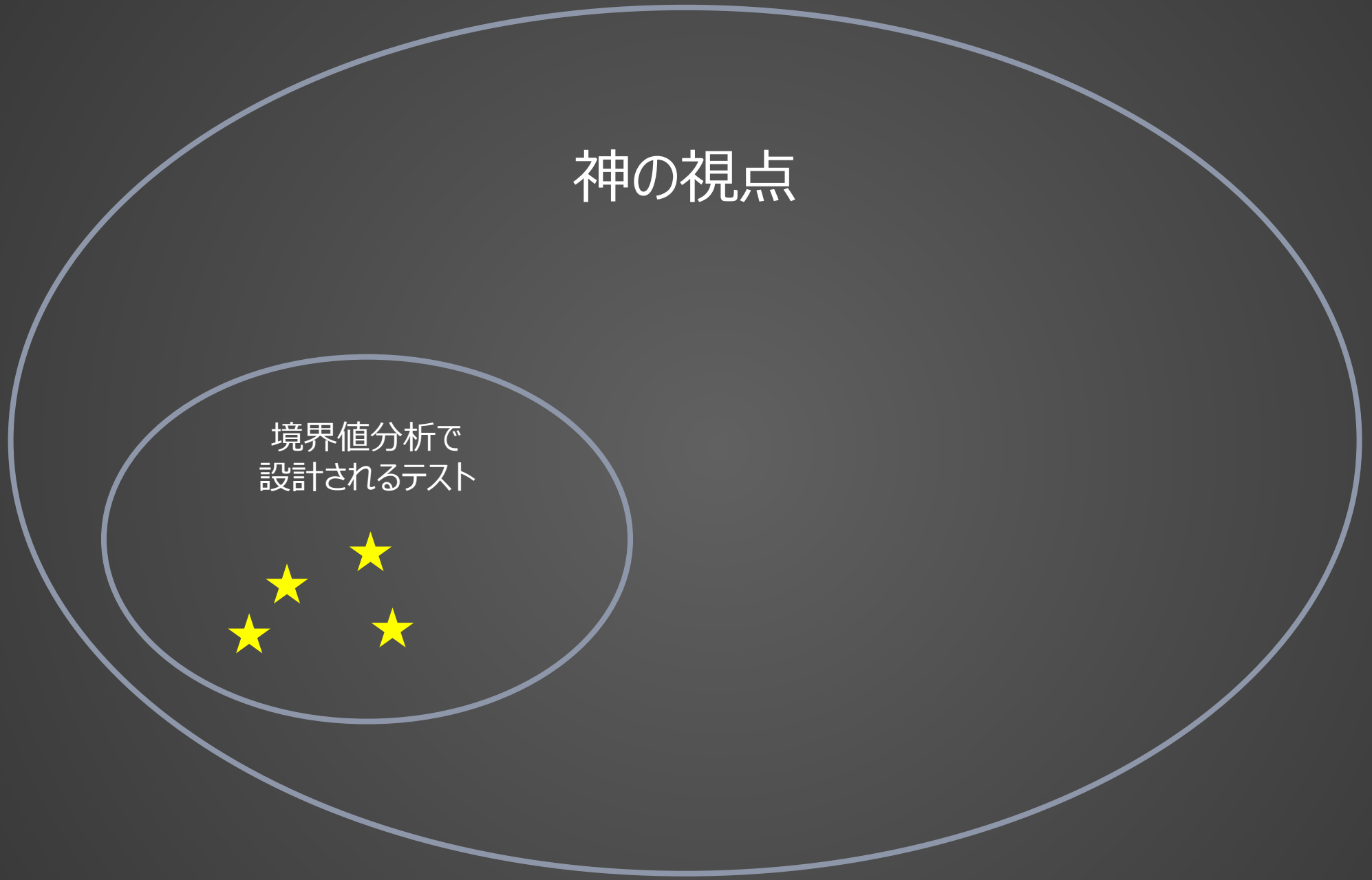
たぶんそう。でも証明できない。

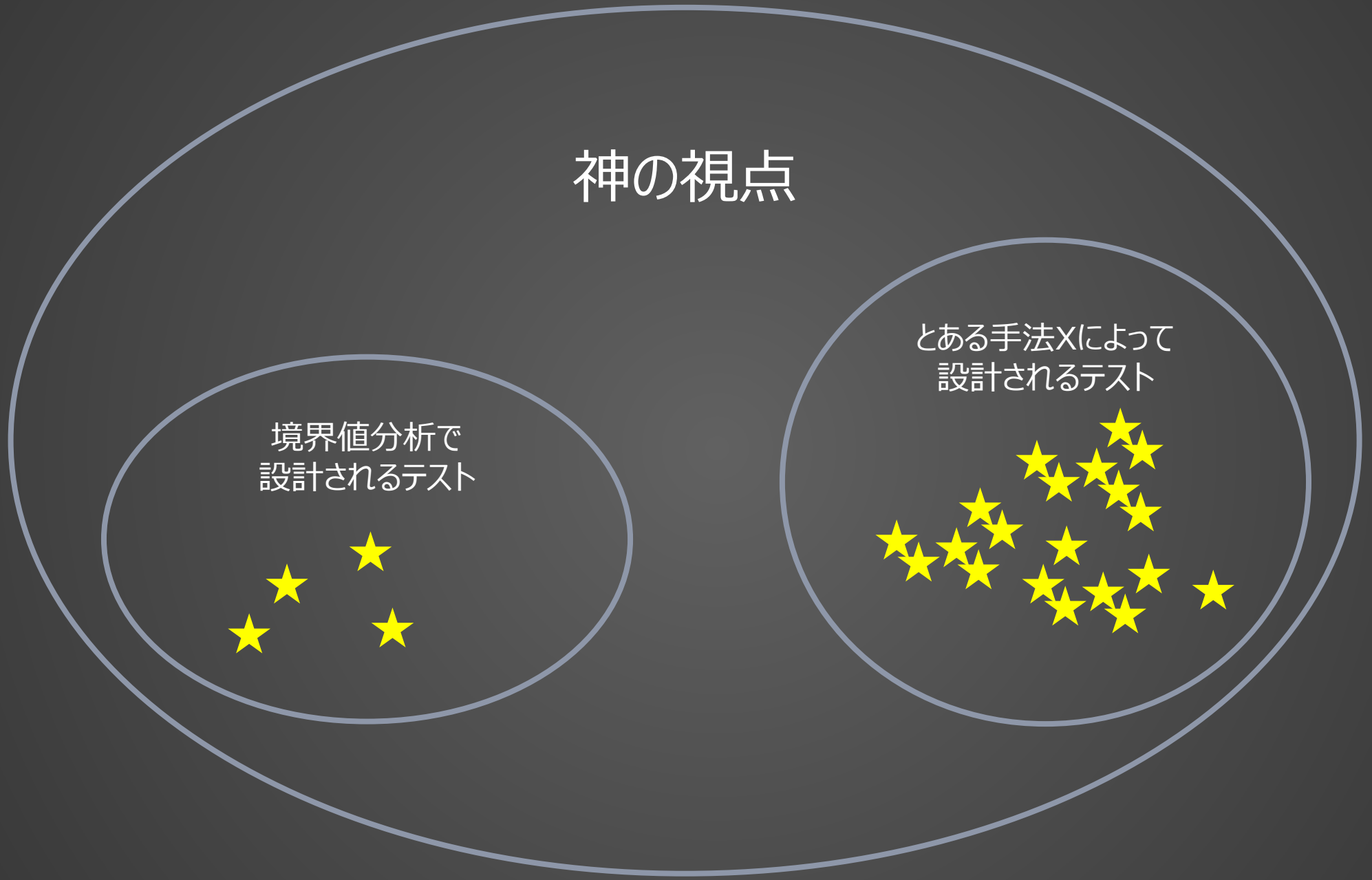
境界値分析

- 境界にはバグが潜む。
- 等符号の使い方を誤ったりするから。

自分の体感

- 言うほど等符号まちがわないよ
- そんなことよりAPIが思ってたのと違うデータ返すことのほうが多いよ
  （その原因は境界値なのかもしれないが・・・）





本来、とても難しい問題を
簡単な問題に置き換えてしまうことで
論理的に解決した気になっている。

本来、とても難しい問題を
簡単な問題に置き換えてしまうことで
論理的に解決した気になっている。

論理的に解かないという選択肢はないのか？

もっと大胆に視野を変える必要がある

- 綺麗なテストアーキテクチャ
- 高度な技術
- 美しいレポート

素晴らしい。
ただ世の中を変えるためには、能力だけでなく、「普及のしやすさ」、「わかりやすさ」も重要である。

もっと大胆に視野を変える必要がある

- 綺麗なテストアーキテクチャ
- 高度な技術
- 美しいレポート

素晴らしい。
ただ世の中を変えるためには、能力だけでなく、「普及のしやすさ」、「わかりやすさ」も重要である。

万人に伝わらない「論理的できれいな手法」よりも
万人に伝わる「現実的で泥臭い手法」のほうがいい時もある

品質が良いとか、十分にテストをした。なんて
誰にも証明できないことなのではないか・・・・・・？

論理的に解かないという選択肢はないのか？

もっと現実的な効果を積み上げていくほうが
いいのでは？

NTT DATA

Trusted Global Innovator