

2018  
3/7

JaSST'18東京 E2) 全国JaSST実行委員セッション1

# やってみよう！探索的テスト

～ハイクオリティな妄想の高速ループ～

JaSST Hokkaido 実行委員会  
中嶋 信(TEF道)  
根本 紀之(TEF道)  
小楠 聡美(TEF道)

# 本日のメニュー

- 座学タイム (20分ぐらい)
- ワークタイム (60分ぐらい)
- 事例紹介 (5分ぐらい)
- まとめ (5分ぐらい)

「探索」  
という単語を  
考えてみる

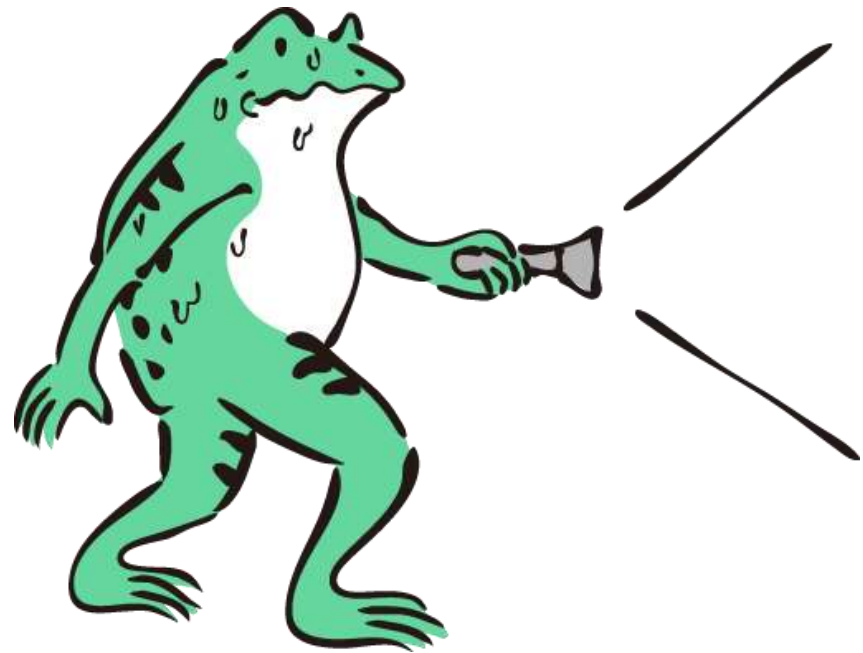
# 探索



未知の事柄などをさぐり調べること。

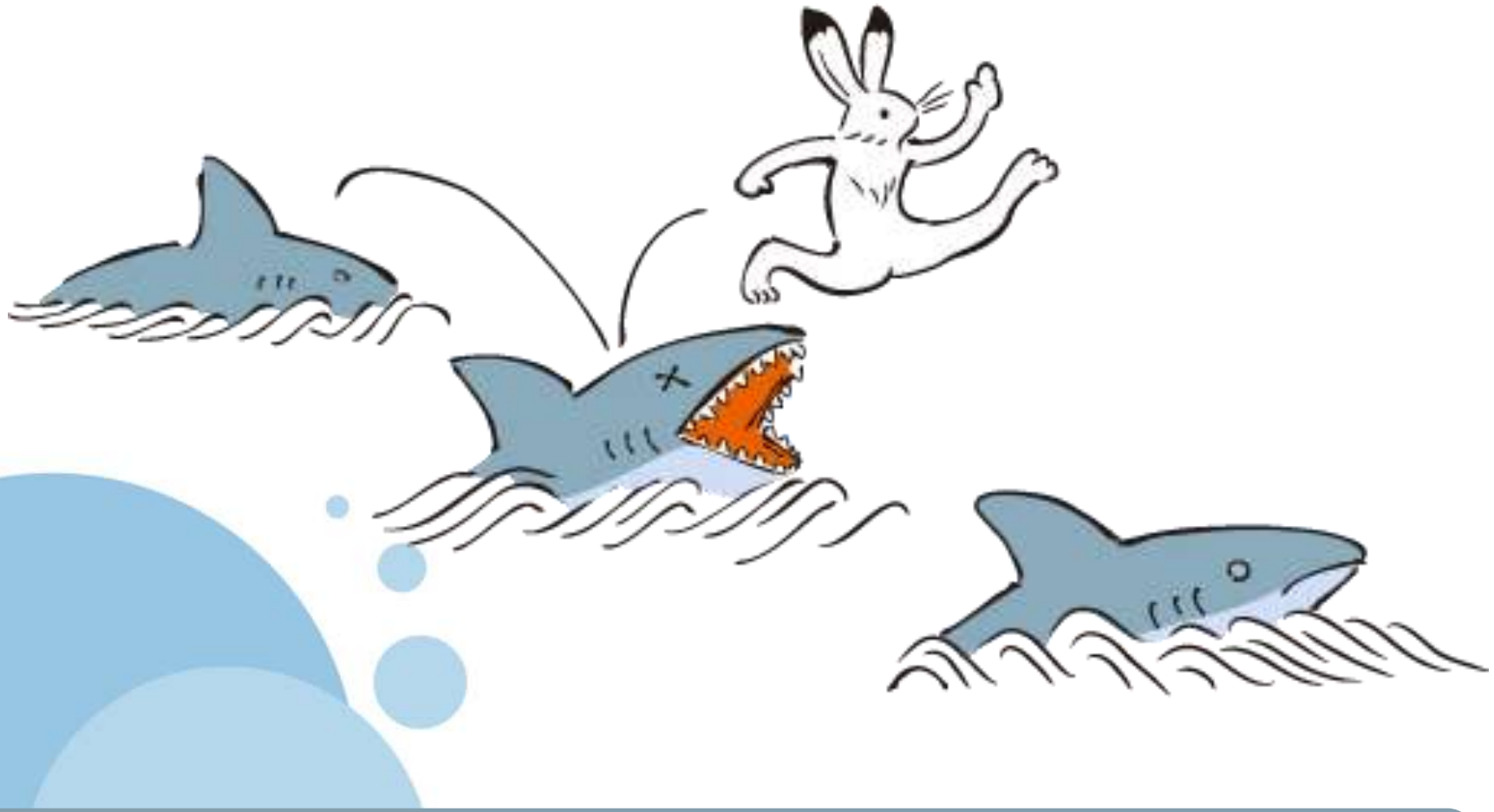
# 「探索」 に似た単語と 比較

# 探検



**危険を冒して未知の地域に入り、実地に調べること。**

# 冒険



成功の見込みの少ないことを無理にすること。

# 散策



これといった**目的もなく**ぶらぶら歩くこと。

# 徘徊



無意識のうちに**目的なく**歩きまわること。

# 用語の違いを整理

		意識が ある？	目的が ある？	危険が ある？	意義
徘徊	無意識のうちに目的なく歩きまわること。	なし	効果がない	なし	なし
散策	これといった目的もなくぶらぶら歩くこと。	あり	効果がない	なし	なし
探索	未知の事柄などをさぐり調べること。	あり	あり	なし	何かを明らかにする
探検	危険を冒して未知の地域に入り、実地に調べること。	あり	ちよつと	あり	何かを明らかにする
冒険	成功の見込みの少ないことを無理にすること。	あり	やりすぎ	あり	やることに意義

ただ何となく  
「徘徊」  
していませんか？



リスクとリターン  
を考慮せず  
「冒険」  
していませんか？



知識を学んじゃおう！

# 座学タイム

# 探索的テストの定義1

## 探索的テスト by Cem Kaner

高橋寿一さんも「知識ゼロから学ぶソフトウェアテスト」で同じ定義を使っています。

- ソフトウェアテストの一つの**スタイル**
- 個人に自由意思を持たせるとともに責任をより明確にする
- 一個人のテスト活動である
- 継続的にテスト活動を洗練させる
- 探索的テストは以下の活動を行う
  - テスト関連の学習 / テスト設計 / テスト実行 / テスト結果報告
- 成熟したテスト活動
- 上記の活動をプロジェクト期間中並行して行う

## 探索的テスト by James Bach

探索的テストは、学習、テスト設計、テスト実行を並行して実施するものである

Exploratory testing is simultaneous learning, test design, and test execution.

# 探索的テストの定義2

## 探索的テスト by Erisabeth Hendrickson

直近の実験から得た“気づき”を次の実験へ活用し、テスト設計と実行を同時に行い、システムについて学習していくこと

Simultaneously designing and executing tests to learn about the system, using your insights from the last experiment to inform the next.

## 探索的テスト by JSTQB用語集

非公式なテスト設計技法の一つ。テストを実施する過程で、テスト担当者がテスト実施情報を活用しながらテスト設計をコントロールし、積極的に質の高い新しいテストケースを設計する。

# 探索的テストとは・・・

「対象を動かしながら、テスト設計～実行～フィードバックを行っていく（ハイクオリティな妄想のループ）対話型のソフトウェアテスト」だと思っています。

医者が患者の診察をするのと同じように、ソフトウェアの動きを見ながら、どこが悪いかを探し当てていくプロセスに近いです。

思いついたテストケースを実施するアドホックテストとは違います。

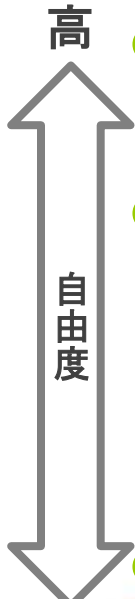
参考) <http://www.itmedia.co.jp/im/articles/1111/07/news203.html>

# どうして探索的テスト？

- 短納期でバグを見つけたい。
- テストの重み付けをその場で変えることができる。
- 実際のソフトウェアを動作させてから確認する。
- 暗黙知を活用しやすい。

# 探索的テストの種類

今回の演習は  
ここをやります

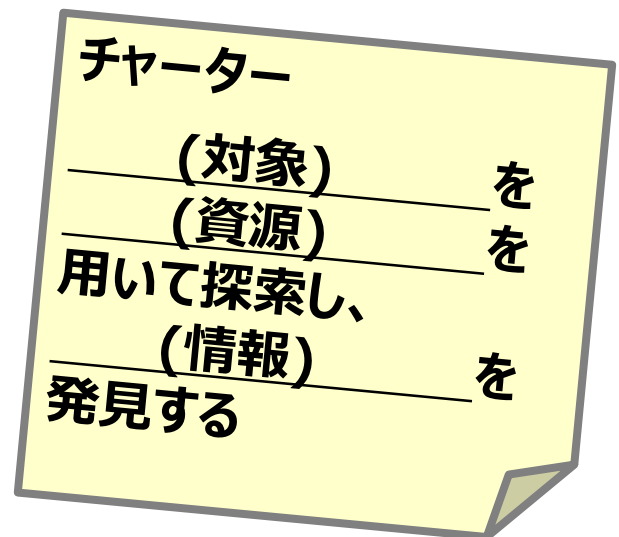
- 
- フリースタイルの探索的テスト
  - テストチャーターを用いる探索的テスト
    - ※テストチャーター：簡単に言うと探索的テストの『道しるべ』となるもの。テストの目的達成のための方針や目印。抽象度の高いテストケースや、機能リスト、リスクリスト等
  - セッションベースドテスト
    - 時間を区切ってセッションという単位で行う。各セッションにはテスト目的であるミッションが含まれる。

# チャーターの話

- 指針を決めるものである。
- 粒度は様々である。
- 細かな決まりはない。

手順では  
ないよ！

一例として以下のようなものがある。(参考：Explore It!)  
(対象)を(資源)を用いて探索し、(情報)を見つける



# チャーターの例①

細かめに  
出してます

どこ(何) をテストしますか	どうやってテストしますか (how, with resources, condition, etc)	どんな不具合を発見する事をねらいますか？
リモートクライアント	キーボードのみを使って操作する	Windowsアプリケーション特有の操作に関連する不正動作がないかどうか
パラメータ画面	権限の異なるユーザーで編集を繰り返す	見えてはいけないパラメータが履歴、別画面などから見えてないか
エクスポート全般	エクスポート途中の言語切り替え	エクスポートされるものの言語が変わらないか。 途中でおかしな動作をしないか。
エクスポート全般	エクスポートの中断	ファイルが間違ってエクスポートされていないか。
ユーザー管理機能	権限の継承設定や個別設定への変更	権限がないフォルダやアプリケーションを参照できないかどうか。 権限がとびとびのフォルダになっている場合に、間違っ て参照できるようになっていないか。
全般	文字列の統一感はある問題ないか (日/英)	各画面によって使用する文字列が違ってないか。
リモートクライアント	色合いは問題ないか	装置は黒背景、リモートクライアントはグレー背景 なので、見え方に違いが出ることを認識しているか

# チャーターの例②

## ねらい

リンクのさせ方でエラーになるケースを探す。

循環リンク

ドメインを跨いだリンク

## ムーブ機能

上位リンクがあるときはリンクをきるかどうかの確認をされるので、上位リンクがあるケースで操作してみる。

ムーブと言いつつ改名だけして移動はさせない。

アーカイブ→リストアで、以前と同じように動作するかどうか

リモートクライアントと装置で表示・動きが一緒かどうか

別のシミュレータとつないでみる

Win/Linuxで日本語/英語切り替えてみる

リスクリスト  
抽象度の高い  
テストケース

やりすぎると単なる  
テスト仕様書になるの  
で注意！！

# 探索的テストの歴史

## 1.3 探索的テストの歴史的経緯

---

- ソフトウェア開発初期
  - 探索的テストは一般的なテストアプローチ。ソフトウェア開発が始まった初期から実施されてきた
- 1980年代～
  - 手順書に依存せずに分析やフィードバックを活用するアプローチとして、探索的テストという名前で概念を明確化する
    - 1983年にCem Kanerが命名
  - それまで探索的なアプローチとアドホックテストは区別されておらず。  
知的なアプローチ(=探索的テスト)とそれ以外のアプローチを区別するため、探索的テストという考え方を確立した
- 1990年代後半～
  - 専門の文献が整備され、テスト手法の一分野として確立する
    - Ex)「Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design」
    - 「How to Break Software: A Practical Guide to Testing」
      - (いずれもJames A. Whittaker)

# 探索的テスト vs スクリプトテスト

## 1.1 探索的テストとは(2つのパラダイム)

	●探索的テスト	●スクリプトテスト
テスト手順の管理	・テスト手順のドキュメント化を行わない	・テスト設計を行い、事前にテスト手順をドキュメント化する
テスト実施	・事前分析情報、実施者の能力、動かして得られたフィードバックなどを活用して、テストの構築と実施を並行的に行う	・テスト手順にしたがってテストを実施する

# 探索的テストのメリット

## 1.6 探索的テストのメリット・デメリット スクリプトテストに対するメリット

- 軽快
  - テスト設計・実装コストを削減
  - 必要なところをピンポイントでテスト
- 柔軟性が高い
  - テスト手順の変更コストを削減
  - 実物に合わせながらテストを構築できる
  - 人間の知能を使って次のようなものにも柔軟に対応できる
    - 仕様やテスト設計の穴
    - 明文化しにくい操作や動作
- 人間の能力を活用しやすい
  - 暗黙知や明文化しにくいノウハウを活用できる

欠陥検出での  
高い生産性

テストの保守コ  
スト削減

検出困難な欠  
陥検出の実現

# 探索的テストのデメリット

## 1.7 探索的テストのメリット・デメリット スクリプトテストに対するデメリット

---

- 属人性が高い
  - 人によって効果が変動する
    - 能力のない人間が担当するとアドホックテストと変わらなくなる
- テスト設計の品質がわかりにくい
  - 漏れ・だぶり・網羅率がわかりにくい
  - テストの品質を保証しにくい
- 記録が残りにくい
  - テストのエビデンスや監査証跡を残しにくい
  - 再現性を確保しにくい
- テスト設計段階でのバグの予防効果が低い場合がある
  - テスト分析・設計による問題検出の前倒しがない

# 探索的テストと スクリプトテストの併用パターン

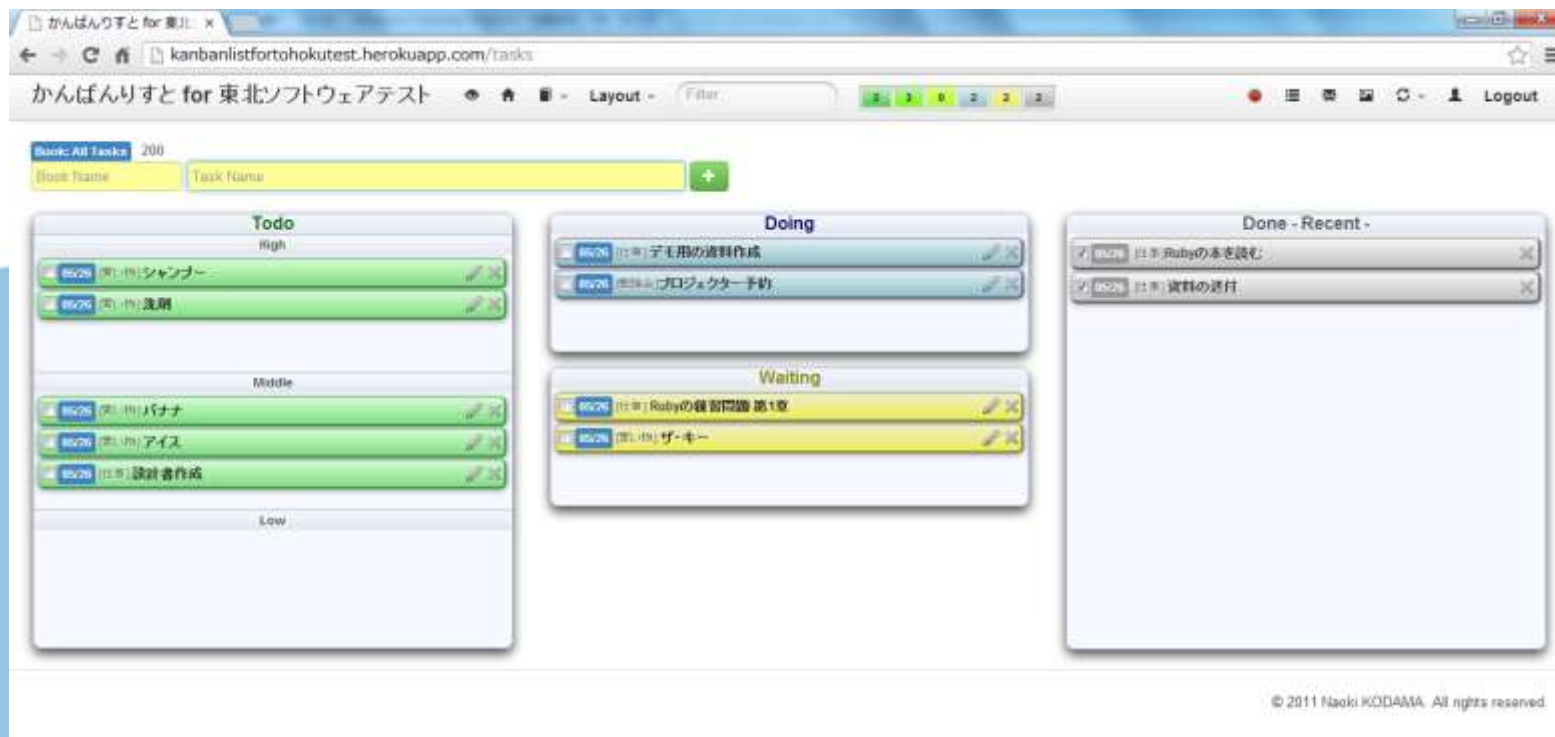
	説明	図
A	初めにスクリプトテストを実施し、弱い箇所を見当つける。その箇所に絞って、探索的テストを実施する。	
B	初めにスクリプトテストを実施する。その後、探索的テストを実施するがスクリプトテストの結果を考慮しない。	
C	初めに探索的テストを実施し、弱い箇所を見当つける。その箇所に絞ってテスト設計を行い、スクリプトテストを作成し、実行する。	
D	探索的テストとスクリプトテストを同時に行う。	

ガチで探索的テストを  
実行してもらいます

# ワークタイム

# 本日のお題：かんばんりすと

- 実際にユーザーが存在するWebサイト
- 勉強会用にソースコードをもらい、本家のサイトとは別に立ち上げている。

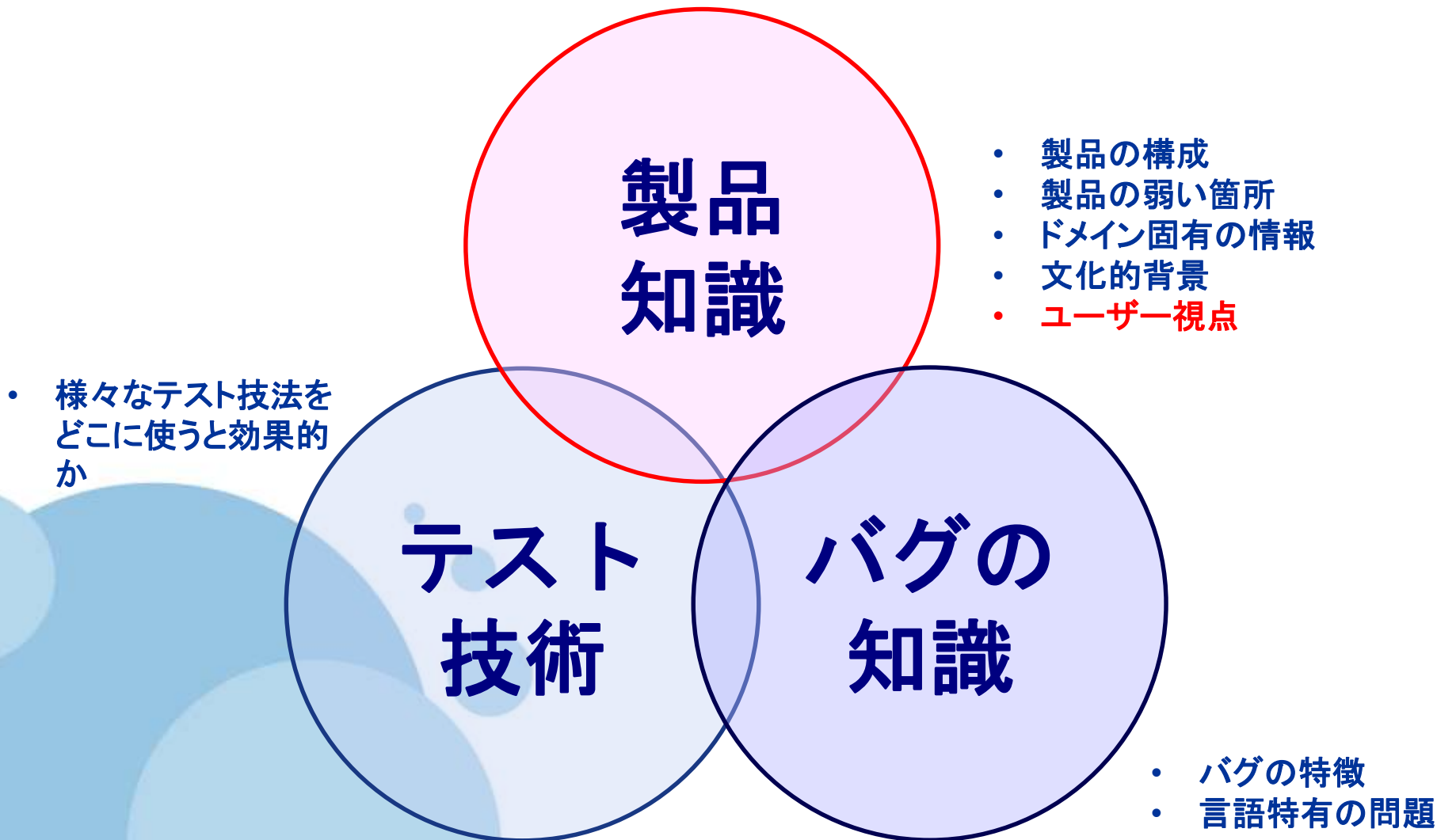


# 準備の時間 [10min]

- サイトにアクセスしてください。  
<http://kanbanlistfortohokutest.herokuapp.com/>
- アカウント登録してください。
- タスクを以下で登録してください。
  - BookName: テスト勉強会
  - Task: 探索的テストの実行
- タスクをDoingに移動させてください。
- とりあえず操作してバグを発見してみてください。

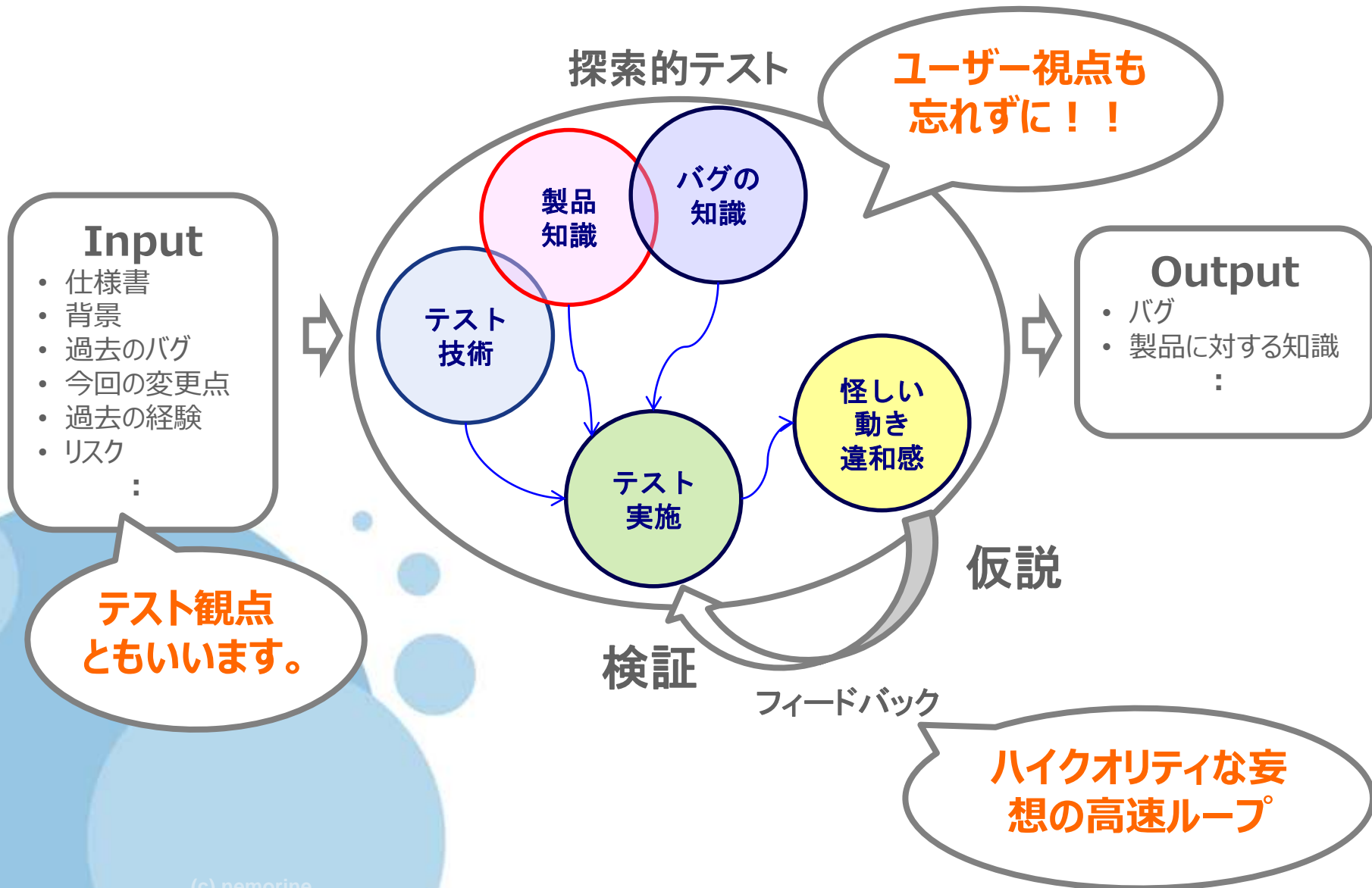


# 探索的テストに必要な3要素



# 探索的テストのイメージ

## 探索的テスト



# バグを上手く検出するコツ

- 仮説～検証を考えながら、テストを実行する
  - コードや設計はどうなっているのか想像する
- テストの技術（@nemorine のオススメ順）
  1. バリエーションに着目する
  2. データのCRUDに着目する
  3. 状態遷移に着目する

Explore It! には  
他にも探索的テストの方法  
が書かれています。

# 1. バリエーションに着目する

- 常に複数のバリエーションを考える
- 操作 / 値 / 順序 ← 最初はこれで十分

同値分割、境界値分析  
もこれに含まれる。

例) “文字入力をする”のバリエーション

- キーボードで打つ
- コピー＆ペーストする
- オートコンプリート
- 別の画面からの入力

## 2. CRUDに着目する

- データのCRUDに着目して、バグが出そうなタイミングで操作する。
  - C : Create
  - R : Read
  - U : Update
  - D : Delete
- Initialize（初期化）も着目すると良い  
→ 2回目の操作 など

# 3. 状態遷移に着目する

- 状態遷移に着目する
  - 状態がフラグ（Bool値）の集合である場合は、フラグの落とし忘れなどがあるかもしれない。
  - 自動で起きる状態遷移は気をつける（タイムアウトなど）
  - 状態遷移の間に時間がかかる処理があるときは、そのタイミングを狙う。

例) 有名なバグでは、  
チビファイヤーマリオ



# ワーク

## マニュアルを用いた探索的テスト [15min]

以下のチャーターに沿って探索的テストを行ってください。

バグを出した場合は書き留めておいてください。

### チャーター

(対象) かんぱんりすと を  
(資源) マニュアル を  
用いて探索し、  
(情報) マニュアルとの不一致 を  
発見する

どんなバグがみつかりましたか？

# グループでちゃんと確認 [15min]

- 他のはどのようなバグを見つけましたか？
- それは**どう考えて**、検知したのでしょうか？
- 下記の点に着目してバグを見つけるまでのプロセスを共有してください。
  - 怪しい動作
  - 仮説
  - 検証

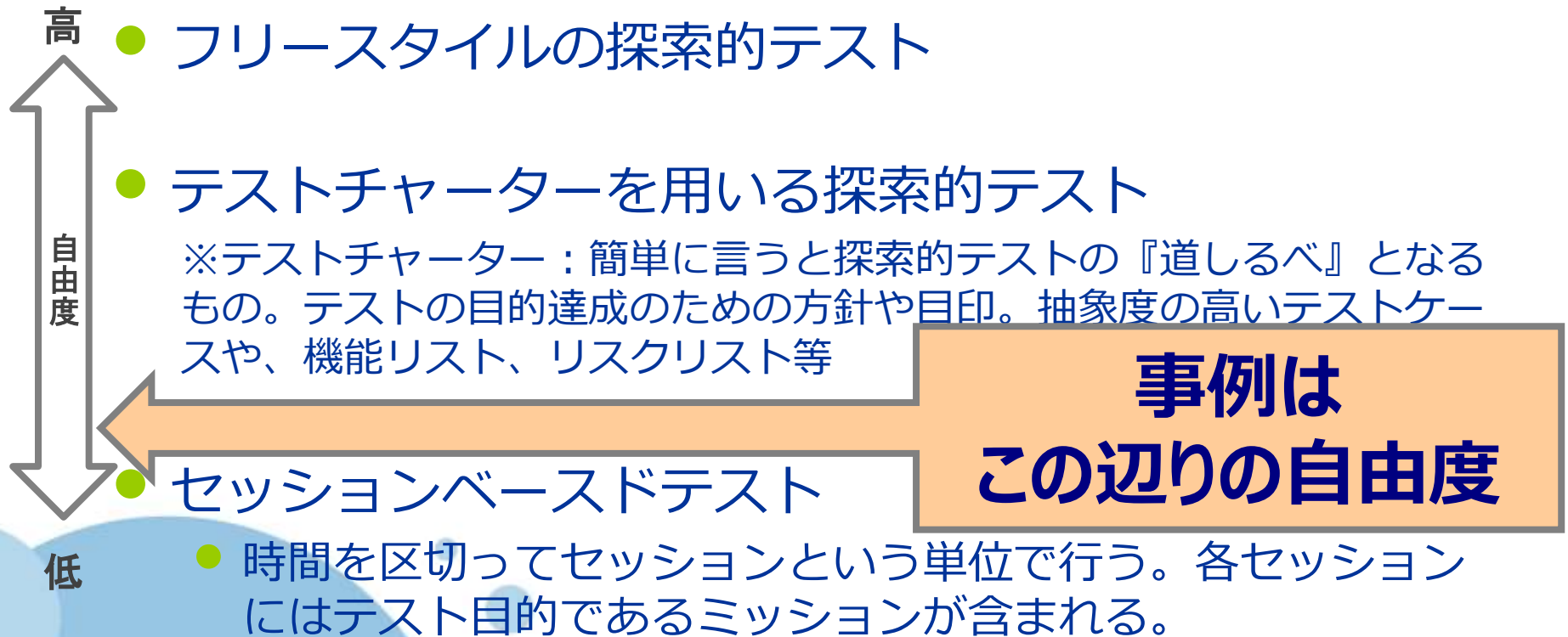
# Demo

- ・最大文字数の謎
- ・隠れた仕様
- ・状態遷移の穴

**イメージできましたか？  
「仮説～検証の繰り返し」**

# 事例紹介

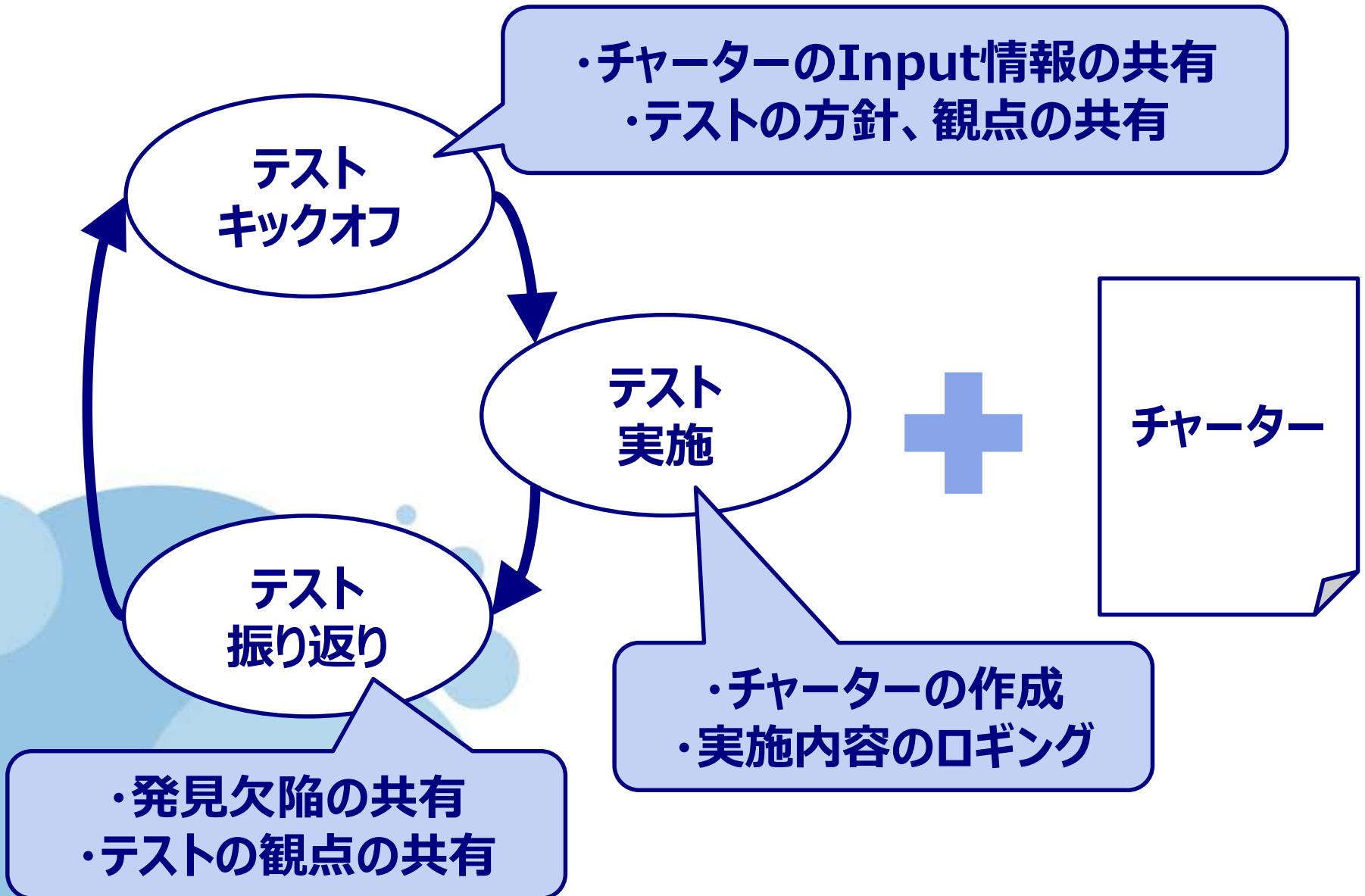
# 探索的テストの種類



# とりくみ

- チャータの利用
  - 漠然とテストしないようにする
  - 担当をはっきりさせる
- 対象案件の共有
  - 狙いの絞り込む
  - 開発者の心配ごとを引き出す
- テスト結果の共有
  - 検出欠陥からノウハウを得る
  - 実施内容からノウハウを得る

# 施策とプロセスの関係図



まとめ

# 探索的テストでよく聞く悩み

- 長時間やってしまう
  - タイムボックスで切るのが一般的。（JSTQBより）
  - キチンと考えたのがセッションベースドテスト
- エビデンスが残らない
  - 残したければスクリプトテストにする。あくまでバグを見つけることを第一とする。
  - また正式な記録を強制することで、思考が止まるという可能性もある。
- ユーザーにとって、あまり重要ではないところを探索してしまう。
  - 狙いどころを絞るチャーターを用意したり、リスクベースで実施する

# 探索的テストの注意点

- 探索的テストには100%頼らない
  - 自動化テストやスクリプトテストと組み合わせて考える必要がある。
- 探索的テストはトレーニングを十分受けたテスト担当者によってなされる
- ユーザビリティを除く非機能テスト（負荷テストなど）を探索的テストではアプローチしないかも・・・
- 見つけた不具合がどうして仕込まれたのかを分析できる必要がある

# 探索的テストのヒントワード

- 書籍「Explore It !」より (@nemorineの訳)

- 抽象
- 決して / いつも
- はじめ、途中、さいご
- 全てを集中させる
- モデルを変える
- CRUD
- 全てを発散させる
- データに沿う
- 大きすぎる / 小さすぎる / ちょうど良い
- 妨害する
- 逆にする
- いくつか / ない / 全て
- 不足させる
- 多量すぎる / 少量すぎる
- 有用な近似
- 異常データや異常フォーマット
- 0(ゼロ)
- 0、1、たくさん
- ズーム
- 後ろ、前、履歴
- しおりを入れる

# まとめ

- 仮説～検証を意識しながら探索的テストをしてみよう！
- スクリプトテストと上手く併用しよう！
- 「どう考えて出したのか」は重要な情報なので、チームで共有していこう！

# 參考資料

# 探索的テストの参考資料

## ◆探索的テスト

- 高橋 寿一  
知識ゼロから学ぶソフトウェアテスト 【改訂版】 4章  
<http://www.amazon.co.jp/dp/4798130605>
- 高橋 寿一  
探索的テストってなんですか？ - JaSSTソフトウェアテストシンポジウム PDF  
<http://www.jasst.jp/symposium/jasst14kyushu/pdf/S3.pdf>
- 井芹 洋輝 (@goyoki)  
探索的テスト入門 (SlideShare)  
<http://www.slideshare.net/goyoki/ss-34292539>
- James Bach  
General Functionality and Stability Test Procedure  
<http://www.satisfice.com/tools/procedure.pdf>
- James Bach  
Exploratory Testing Explained(v1.3 4/16/03) PDF  
<http://www.satisfice.com/articles/et-article.pdf>
- James A. Whittaker  
How to Break Software: A Practical Guide to Testing (英語) ペーパーバック - 2002/5/9  
<http://www.amazon.co.jp/dp/0201796198/>

# 探索的テストの参考資料

- James A. Whittaker  
Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design (英語) ペーパーバック – 2009/8/25  
<http://www.amazon.co.jp/dp/0321636414>
- ※一部抜き出したバージョンがmicrosoftのサイトにあります。  
[https://msdn.microsoft.com/ja-jp/library/jj620911\(v=vs.120\).aspx](https://msdn.microsoft.com/ja-jp/library/jj620911(v=vs.120).aspx)
- Elisabeth Hendrickson  
Explore It!: Reduce Risk and Increase Confidence With Exploratory Testing (英語) ペーパーバック – 2013/2/28  
<http://www.amazon.co.jp/dp/1937785025/>
- Cem Kaner  
A Tutorial in Exploratory Testing  
<http://www.kaner.com/pdfs/QAIEExploring.pdf>

## ◆セッションベースドテスト

- James Bach  
Session-Based Test Management  
<http://www.satisfice.com/sbtm/>
- Jonathan Bach  
Session-Based Test Management  
<http://www.satisfice.com/articles/sbtm.pdf>

# お願い

- 本資料やサイトを使って、自社でワークをする場合は下記まで連絡いただくと助かります。

tefdoblog@gmail.com

