

Software Symposium 2017  
探索的テストにおける不具合  
発見率向上に向けた取り組み

株式会社LIFULL 中野直樹

# 探索的テストの導入の経緯

現在、弊社では自社のウェブサービスを開発・運営しており、社内外で様々な開発プロジェクトが稼働中である。また、開発プロセスはプロジェクト(チーム)で決定する方針である。

QAがテストサポートとして参加するプロジェクトの中で、以下のような課題を持つプロジェクトが複数存在していたため、課題解決に向けて探索的テストの導入を行った。

- ・プロトタイプ開発でテストベースが不足している
- ・短納期でマニュアルテストが十分か確認することが困難

# 探索的テストの様々なスタイル

探索的テストに関する技術的な知見は、様々な形で存在しているが、どれも似て非なるスタイルが定義されていることがわかった。

名前	キーワード	名前	キーワード
Cem Kaner *1	<ul style="list-style-type: none"><li>• テストのスタイル</li><li>• 学習する</li><li>• 設計の定義</li><li>• シナリオ設計</li><li>• 実行</li><li>• 解釈</li></ul>	James Whittaker *3	<ul style="list-style-type: none"><li>• 設計</li><li>• 慎重な観察</li><li>• 多様な考え</li><li>• ツアー</li></ul>
Elisabeth Hendrickson *2	<ul style="list-style-type: none"><li>• テスト設計 &amp; テスト実施</li><li>• 学習</li><li>• ステアリング (舵取り)</li><li>• チャーター</li></ul>	James Bach *4	<ul style="list-style-type: none"><li>• テスト設計</li><li>• 慎重な観察</li><li>• クリティカルシンキング</li><li>• 多様な考え</li><li>• 豊富な資源</li></ul>

\*1 Cem Kaner, J.D., Ph.D.: A Tutorial in Exploratory Testing <http://www.kaner.com/pdfs/QAExploring.pdf>

\*2 Elisabeth Hendrickson Explore It!: Reduce Risk and Increase Confidence with Exploratory Testing ,2013

\*3 James A. Whittaker; Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design, 2009

\*4 James Bach Exploratory Testing Explained <http://www.satisfice.com/articles/et-article.pdf>

# 探索的テスト導入後の課題

- 探索的テストを効率的に進める為に技術や経験が必要
  - テスト担当者間のバグ発見数にばらつきがある
- 探索的テストのバグ発見の精度を向上させたい



# チャーターの活用

- 探索的テストの属人性を減らし、汎用的な技術として扱えるように探索的テストのスタイルの中から効果的なものに絞っていく方針で検討を行った
- チャーター(\*5)を使用することで、テストの観点をあらかじめ整理することができる点や、共有や再利用可能な点からバグ発見数のばらつきを抑える効果があると考えた
- チャーターには、「ユーザーマニュアル」、「テスト観点」、「ユースケース」、「過去プロジェクトの不具合情報」などを利用することがあるが、何を使用するかなどは、テスト開始前にチーム内検討を行うこととした

\*5 Elisabeth Hendrickson 2013 Explorer It!

# チャーターエレメントの作成

チャーターの中でも**過去プロジェクトの不具合情報はチャーターとして有用**であることがわかった。

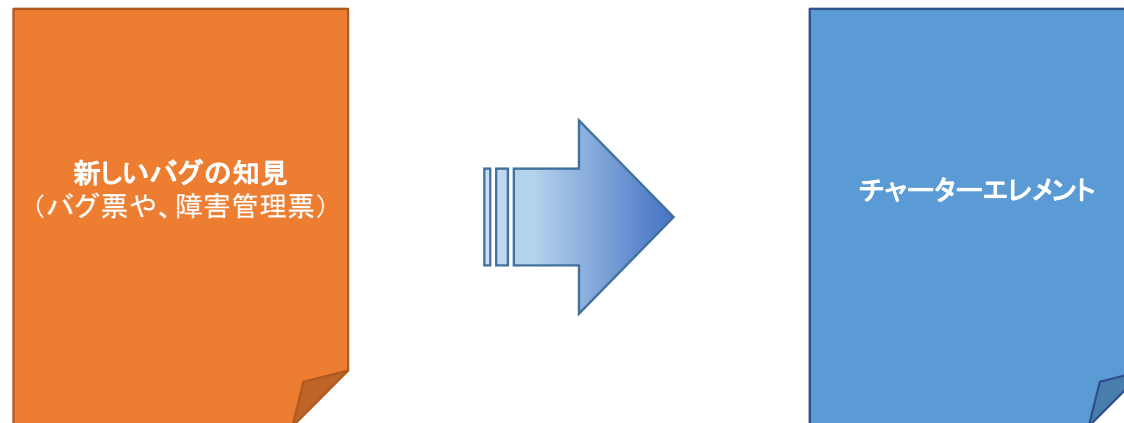
ただし、活用にあたり以下の課題があげられた。

- ・ 情報収集に時間がかかる
- ・ 不具合情報は情報が細かすぎる
- ・ 情報が沢山あった場合に選定する必要がある



# チャーターエレメントの作成

- 「**過去プロジェクトの不具合情報**」をテストプロジェクトのテスト成果物(バグ票)や本番で発生した障害管理票などから切り離して、チャーター作成専用の資料(**チャーターエレメント**)として管理することにした



# チャーターエレメントの作成例

バグ・障害管理票

チャーターエレメント

## 【概要】

詳細画面の〇〇欄の金額が四捨五入されておらず、端数が切り捨てられて表示される

## 【手順】

- 1.Top画面から一覧画面へ遷移
2. 一覧画面で検索実行後、詳細へ遷移
3. 詳細画面で金額欄を確認

## 【期待値】

SQLで××テーブルの値を確認するか、一覧画面の〇〇欄の金額でも確認が可能

## 【発生原因】

- ・
- ・

不具合情報からドメイン情報を排除し、再現できる抽象度で抽出

PC・スマートフォンブラウザ

## 【検索】

- ・
- ・
- ・

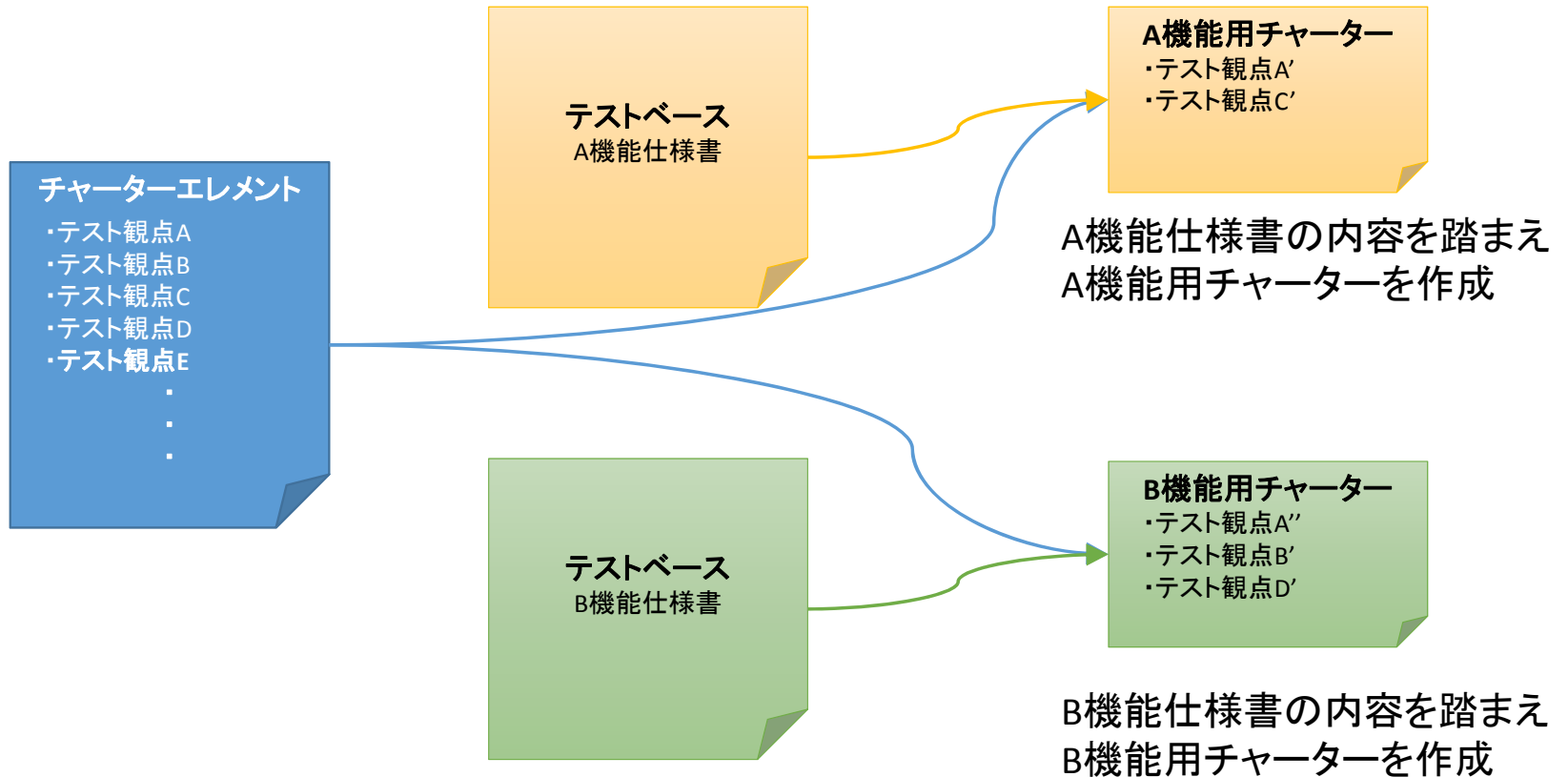
・同一のデータを異なる画面で照会した場合の値の差異

- ・
- ・
- ・

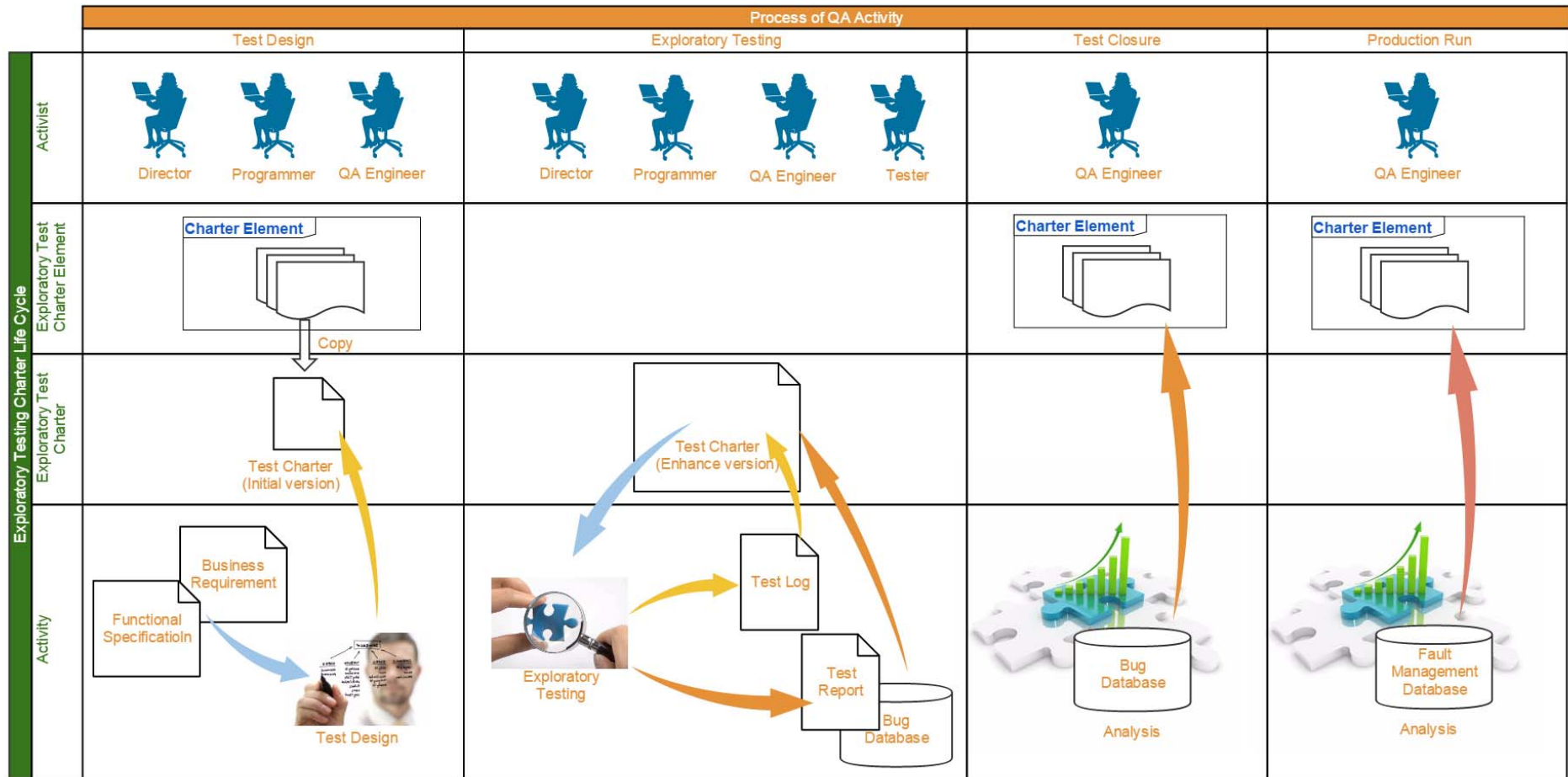
- ・観点B
- ・観点C



# チャーターエレメント使用時のイメージ

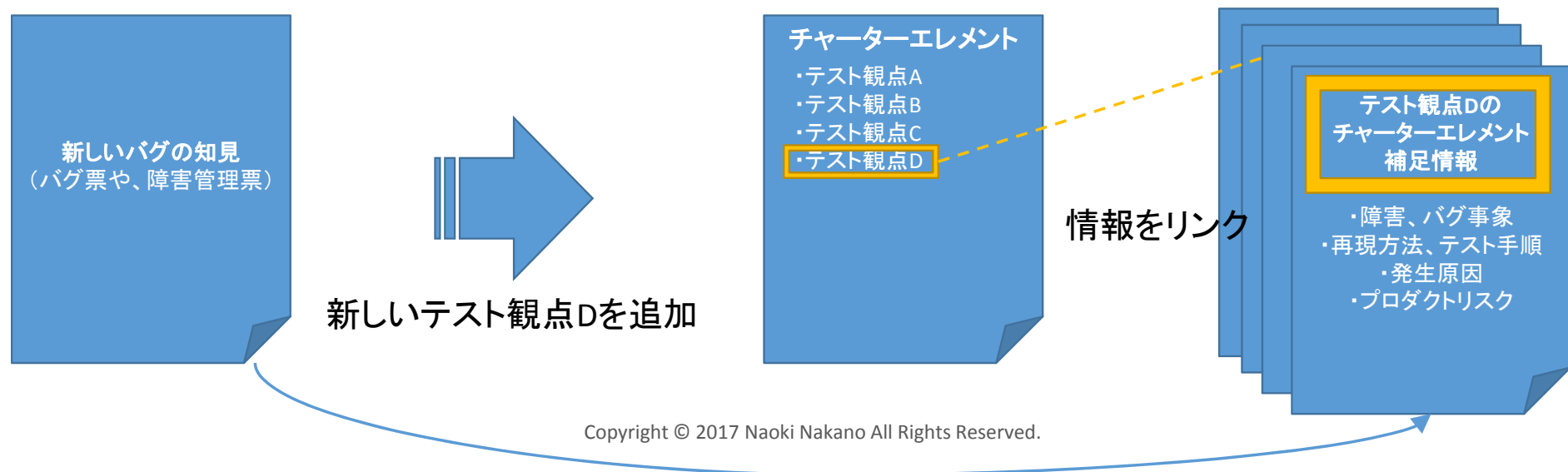


# チャーターエレメントの管理フローのイメージ



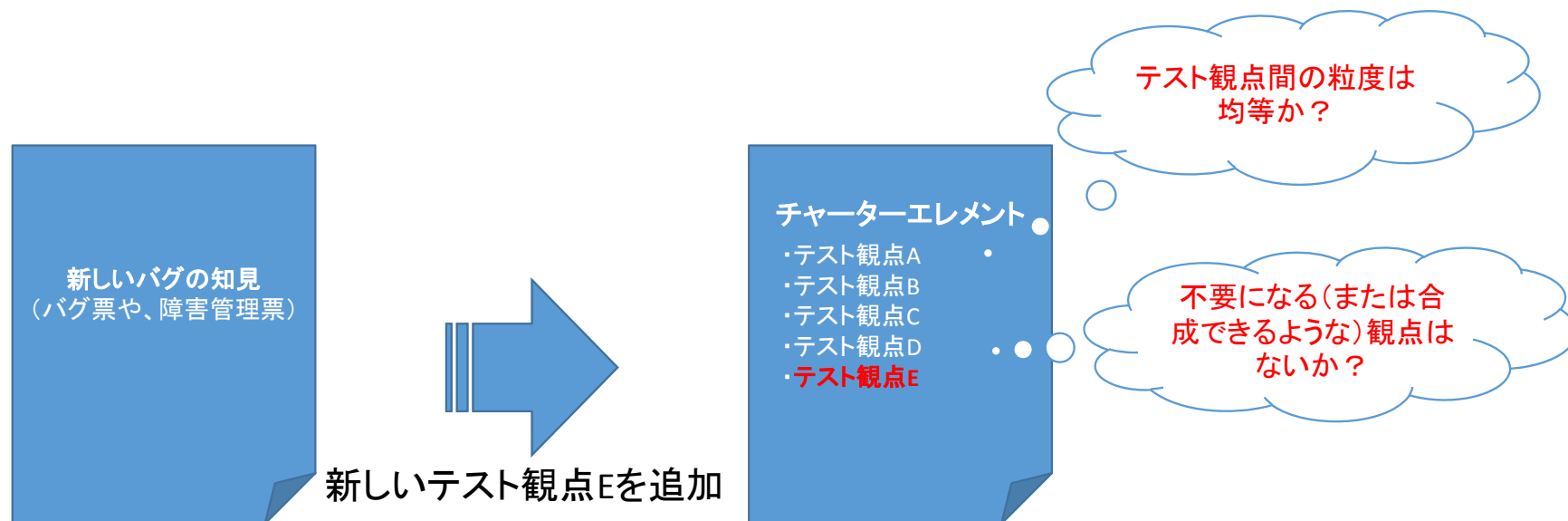
# チャーターエレメントの管理のポイント

- ・ドメイン情報を排除し抽象度の高い状態で保存
  - ・復元できる範囲で抽象化をとどめる
  - ・元となる情報(不具合、障害情報の一部)を参考情報として別資料として作成し、チャーターエレメントからリンクさせて、すぐに閲覧できるようにする



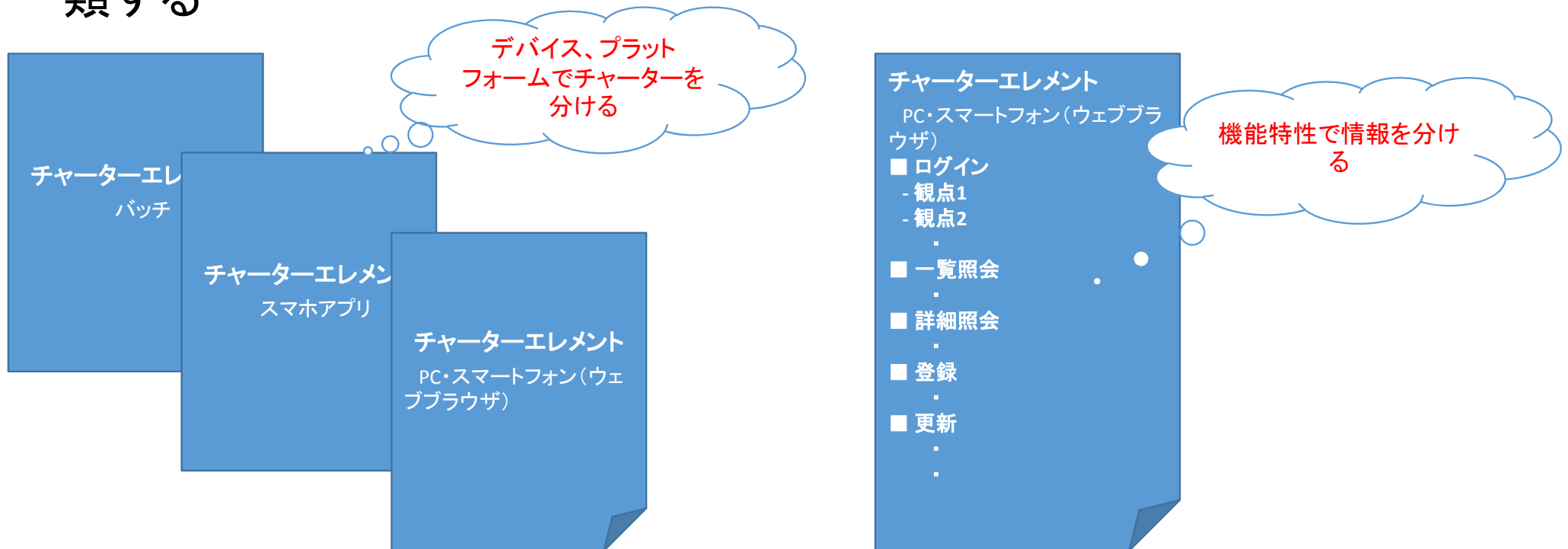
# チャーターエレメントの管理のポイント

- ・ 定期的にメンテナンスを行う
  - ・ 追加する際、MECEになるように全体の整合性を確認



# チャーターエレメントの管理のポイント

- ・チャーターエレメントの情報はデバイスと機能特性（CRUDなど）で分類する



# チャーターエレメントの管理のポイント

チャーターエレメント

バグ・障害管理票

PC・スマートフォン ブラウザ

【検索】

・  
・  
・

・同一のデータを異なる画面で照会した場合の  
値の差異

・  
・  
・

・観点B  
・観点C

元の不具合情報にリンクして閲覧できるように管理



【概要】

詳細画面の金額の四捨五入されておらず切り捨てられて表示される

【手順】

- 1.Top画面から一覧画面へ遷移
2. 一覧画面で検索実行後、詳細へ遷移
- 3.詳細画面で金額欄を確認

【期待値】

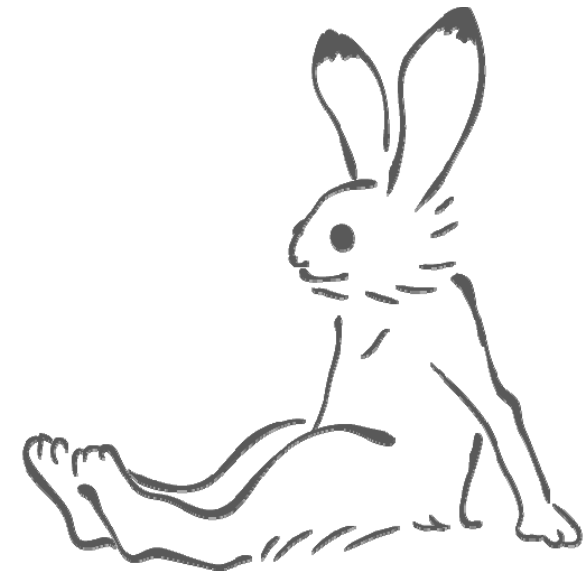
SQLでhogeテーブルの値を確認(一覧画面の金額でも確認可能)

【発生原因】

・  
・

# チャーターエレメントの効果検証

- チャーターエレメント効果として以下の2点において評価を行った
  - 若手テスター<sup>(\*6)</sup>とベテランテスター<sup>(\*7)</sup>の不具合発見数の比較
  - 若手テスター<sup>(\*6)</sup>とベテランテスター<sup>(\*7)</sup>の重大欠陥発見数の比較

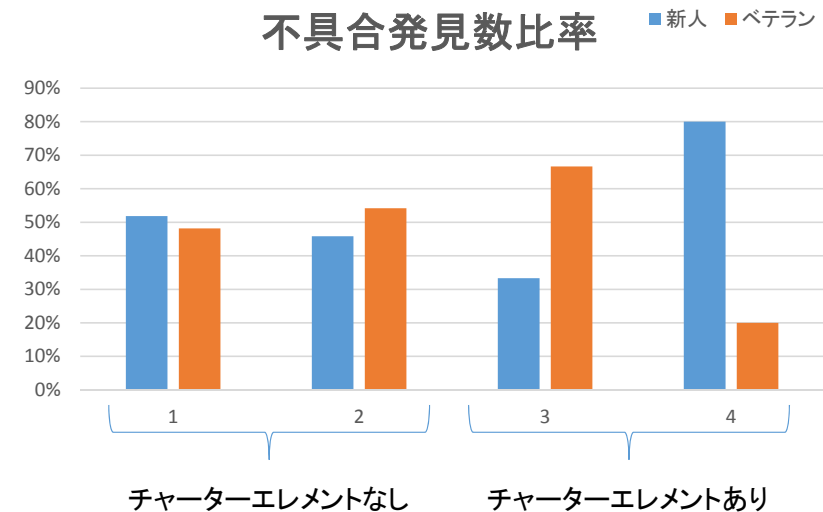


\*6: テスト業務の経験が2年未満のテスター  
\*7: テスト業務の経験が5年以上のテスター

## 効果検証：不具合発見数の比較

- ・若手テスターとベテランテスター各2名で実施
- ・チャーターエレメントの使用有無で2回ずつテスト

	チャーターエレメント	不具合発見数	
		新人	ベテラン
1回目	なし	52%	48%
2回目	なし	46%	54%
3回目	あり	33%	67%
4回目	あり	80%	20%

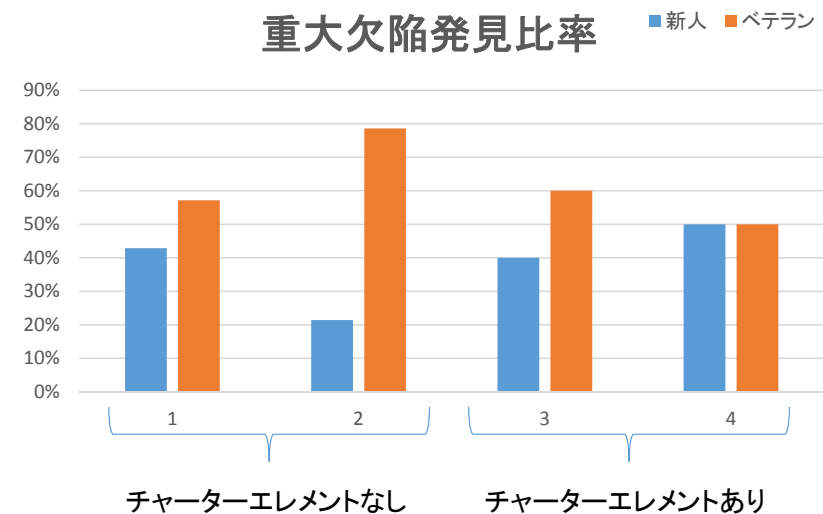




## 効果検証：重大欠陥発見数の比較

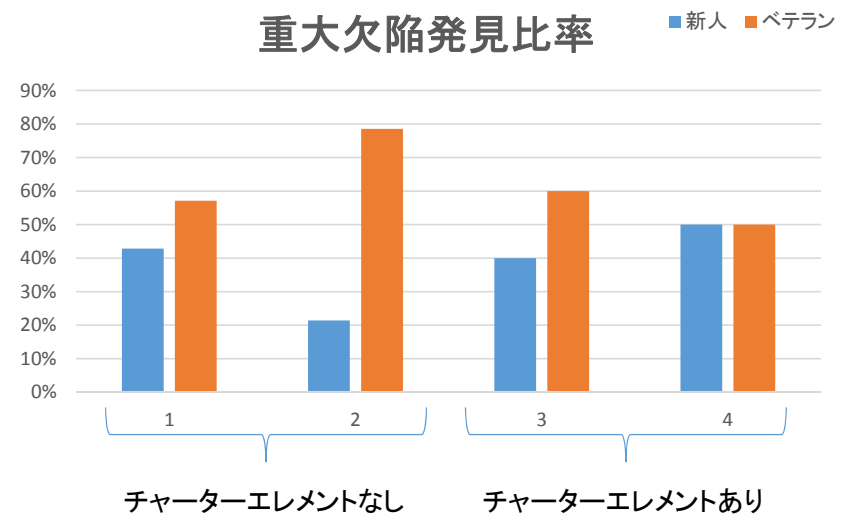
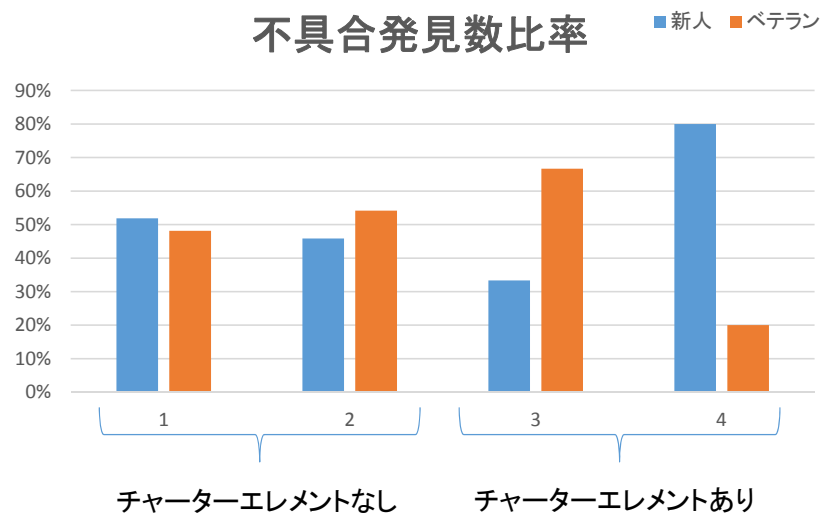
- ・若手テスターとベテランテスター各2名で実施
- ・チャーターエレメント有無毎に異なるテストアイテムに2回ずつテスト

	チャーターエレメント	不具合発見数	
		新人	ベテラン
1回目	なし	43%	57%
2回目	なし	21%	79%
3回目	あり	40%	60%
4回目	あり	50%	50%



# 検証結果考察

- ・ 不具合発見数はよりバラつく結果となった
- ・ 重大欠陥発見数の比率はチャーターエレメントがあった場合の方がバラつきが少ないケースがあった



# 今後の課題

- チャーターエレメントのデータベース化(タグやメタデータなどを検討)
- 効果検証の目的でサンプリング数を増やす
- チャーターエレメントの作成(抽象化)ルールを明確にする

