

# 第3期活動報告会

第3回特別講義「トリガー基礎編」について  
第7回特別講義「トリガー応用編」について

2020年10月1日

日本科学技術連盟 ODC研究会

武田マサヒロ



氏名 武田マサヒロ  
所属 オリンパス株式会社  
仕事 ソフトウェア開発戦略の策定・推進など

この研究会に期待していること

**「ODC分析」を日本に広めるための集団になる！**

目的 ODCの「使い手」を育成する  
進め方 午前中は座学、午後はワークショップの2部構成  
全8回

- 第1回 ODC分析の基礎
- 第2回 タイプ属性分析(基礎編)
- 第3回 **トリガー属性分析(基礎編)**
- 第4回 インパクト・ソース属性分析
- 第5回 プロセスリファレンスモデルの応用
- 第6回 タイプ属性分析(応用編)
- 第7回 **トリガー属性分析(応用編)**
- 第8回 ODC分析の手順と手法例

トリガー・・・欠陥(バグ)が表面化するに至った切っ掛け

ボタンをクリックすると、  
アプリケーションがフリーズする

```
graph TD; A[ボタンをクリックすると、アプリケーションがフリーズする] --> B[ボタンを100回連続でクリックするとフリーズ]; A --> C[ボタンをクリックしただけでハングアップする];
```

ボタンを100回連続で  
クリックするとフリーズ

ボタンをクリックしただけ  
でハングアップする

どちらがより重篤な問題??

- ODC属性の中では、比較的理解が容易である。
- 理解が容易なため、分析結果も説明しやすい。
- 主にテスト担当者が決定する属性なので、テストチームだけでも導入できる。

**ODC分析の中では、比較的とっつき易い属性である。**

# トリガーの特徴 ②難しい点

トリガーは全部で10種類(結合テスト～システムテストの場合)

値		意味・用途
単純 ↓ 手順の複雑さ ↓ 複雑	基本	1つの機能をデフォルトの状態で行った結果、欠陥を検出した場合。
	バリエーション	1つの機能をオプション/パラメータを変更して実行した結果、欠陥を検出した場合。
	順序	複数の機能を順番に実行した結果、または単一機能を複数回実行した結果、欠陥を検出した場合(個々の機能は正しく動作する)。
	相互作用	2つ以上の機能を同時に実行した結果、欠陥を検出した場合。
	負荷/ストレス	リソースに意図的に負荷をかけるテストを実施した結果、欠陥を検出した場合。
	回復/例外	例外処理やリカバリーを試すテストを実施した結果、欠陥を検出した場合。
	起動/再起動	前回のシャットダウンや故障状態(電源の喪失、スリープ、ネットワーク遮断など)から、システムが初期化され再起動するまでの過程を試すテストを実施した結果、欠陥を検出した場合。
HW構成	HW構成	ハードウェア構成の変更に関するテストを実施した結果、欠陥を検出した場合。
	SW構成	ソフトウェア構成の変更に関するテストを実施した結果、欠陥を検出した場合。
通常系	ある意図をもってテストを実施したところ、それ以前の基本的な欠陥(結合テストで検出されるべき欠陥)が検出された(目的としているテストは実施できなかった)。	

正しいトリガーを選べる？ トリガーから何が言える？

# 第3回「トリガー基礎編」午前

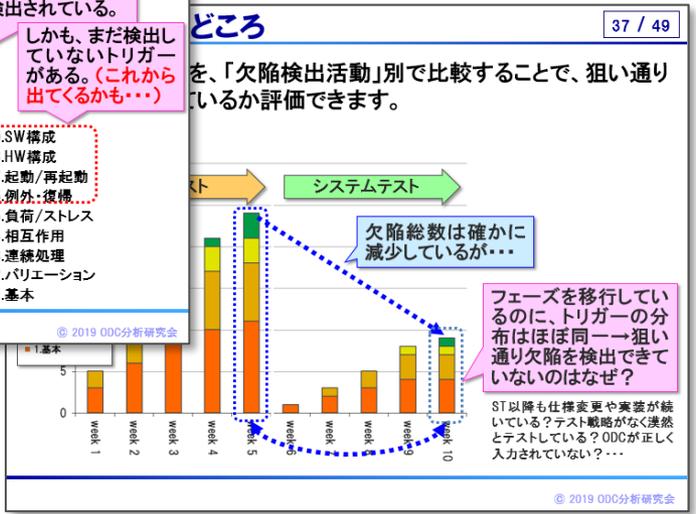
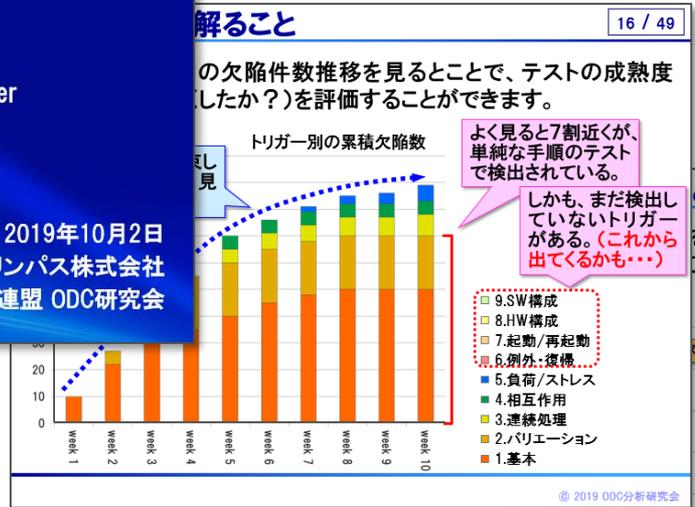
## トリガーの基礎を学ぶための座学

- ①正しいトリガーを選ぶための基礎知識
- ②トリガーを使った分析事例の紹介

**ODC トリガー属性**  
Orthogonal Defect Classification - Trigger

2019年10月2日  
武田匡広 / オリンパス株式会社  
日本科学技術連盟 ODC研究会

No data copy / No data transfer permitted



## ワークショップ

「実際の欠陥例をみて、そのトリガーを選ぶ」

**最初は人によりバラバラ...**



人の意見を聞くことでトリガーの「真髓」を習得  
(表面的ではなく、本質を捉えることが大事)



**徐々に揃ってくる**

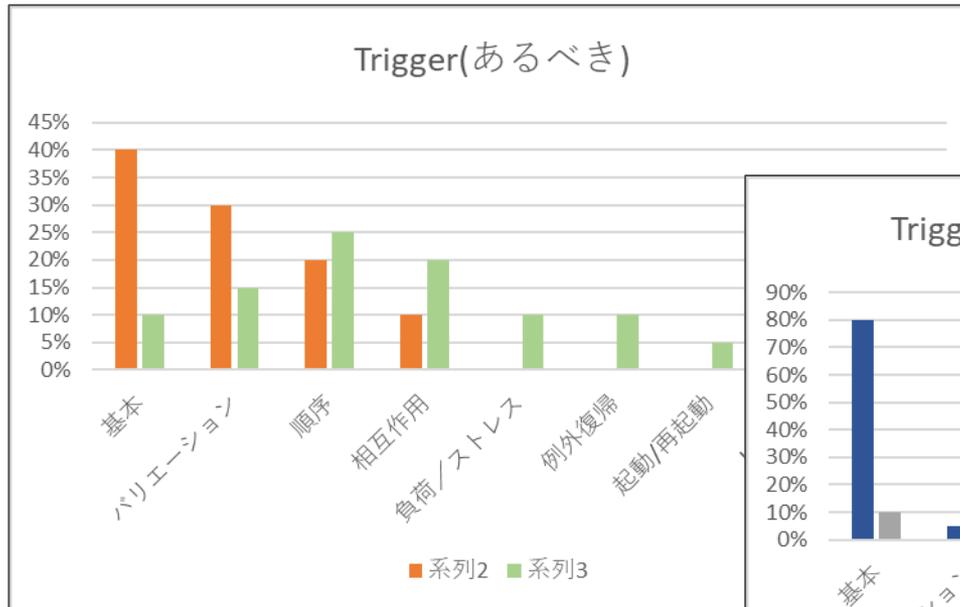
- ①基礎編の振り返り&フィードバック
- ②トリガーを使った分析事例の紹介
- ③レビューへの応用の紹介

**基礎編より一步踏み込んだ内容**

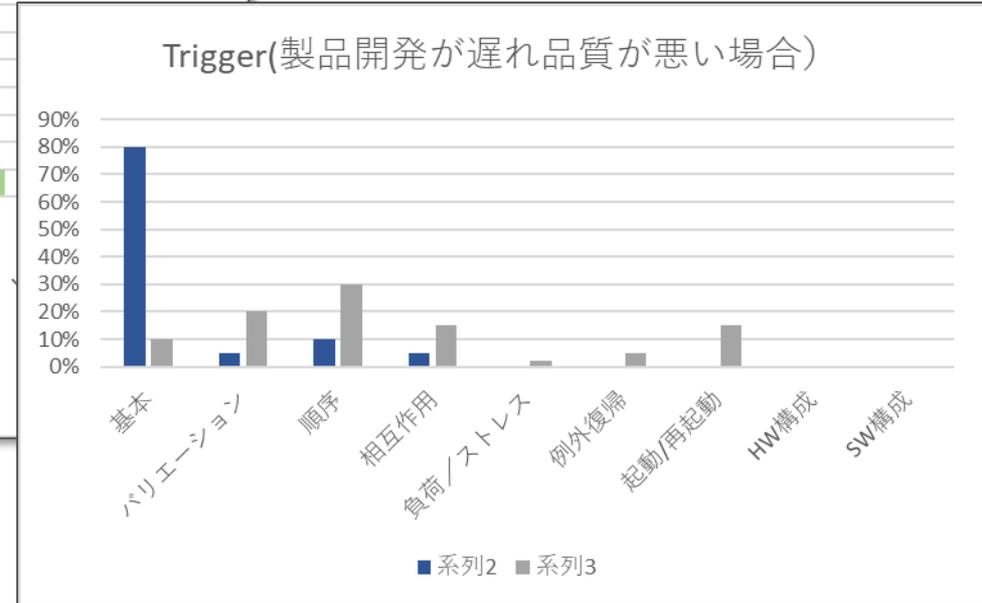
# 第7回「トリガー応用編」午後

## ワークショップ

「実際の欠陥例を使って、品質を語ってみる」



当日のワーク結果例



あるべき姿との乖離から何が起きているかを知る

2.トリガー解説（復習）	
良かった点	わかりにくかった点
レビュー対象とトリガーの関係範囲と値の図が、とても分かりやすかったです。	レビュートリガーを仕様レビューに使う際の割り付け方が未定義な感じがやはりします。
総合システムテストの通常系のトリガーは武田さんがおっしゃる通り、基本、バリエーションなどで分類したほうが分析は早そうに感じました。	レビューに適用するトリガーは例として捉えています、もっと自由に使って良いのではないかと思います。
レビューに適用するトリガーの内容を知ることができた点、スライド24は分かりやすかったです。 トリガー分析事例がないとのことでしたがレビュー観点として使えそうに感じました。	通常系は他のトリガー属性とは性質が異なると思われます。弊社では潜在バグ、既存バグは通常と異なるバグとして大きく分類した後にそのバグの中身の分類をしています。レベルを2つに分けて通常系とそれ以外のトリガー属性としてはいかがでしょうか？
レビュー、単体テストに対するトリガー実用例と実際に考えるべきポイントが分かりました。	結合システムの通常系、属性として用意はされているが通常系として分類するか、基本系のどれかに分類するかは分析して何を知りたいかによる、ということでしょうか？ 通常系プラス、基本系のどれかのように2つ属性を付けるやり方はいかがでしょうか？
レビューに適用するトリガーは活用したかったので解説していただけて良かったです。	
レビュートリガーについて、丁寧に解説していただき良かったです。ODCの現状や自分自身の分からないことが分かって良かった。	
理解が曖昧な点がクリアになりました。	

## 応用編はちょっと駆け足すぎ・・・(反省)

### 理解度

### 有益度

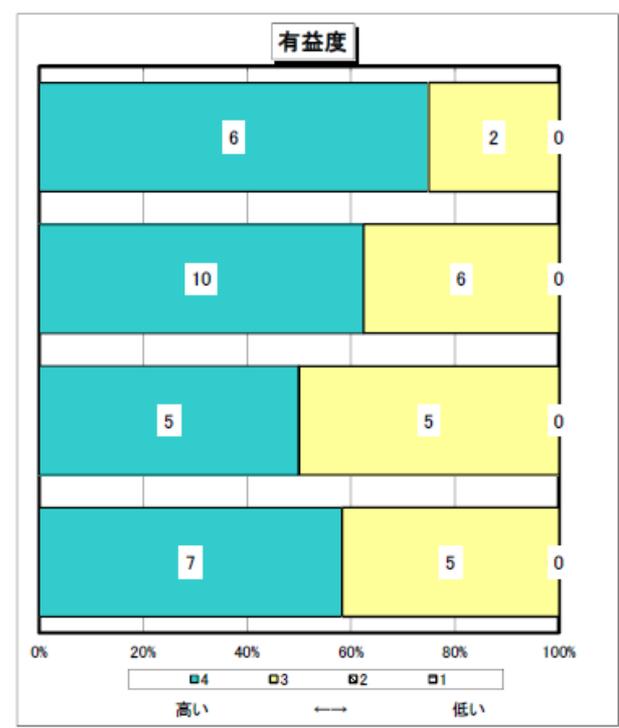
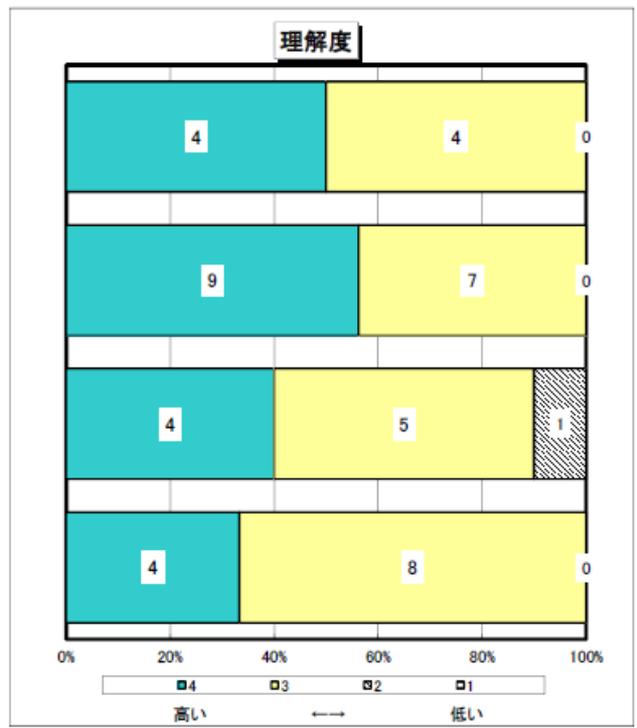
科目

1.ODC分析とは？(復習)

2.トリガー解説(復習)

3.分類演習

分析事例(1)



### 応用編事後アンケート

## テスト対象のとらえ方により、トリガーが変わってしまう

### 問題点

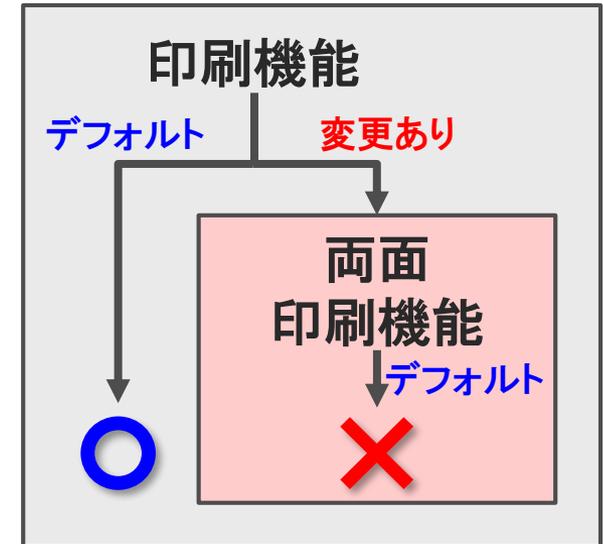
印刷機能にはサブ機能として両面印刷機能がある。通常の印刷は問題ないが、両面印刷を行うため、両面印刷オプションを指定して印刷すると、正しく印刷できない。「印刷機能」で考えればトリガーは「バリエーション」だが、「両面印刷機能」で考えると「基本」となる。どちらが正しいか？

### 対策

テストケースを作成する段階で、そのテストが「何機能を対象としたテストか」を決めておく。この場合は、「印刷機能」に対するテストなのか「両面印刷機能」なのか、テスト設計の段階で決めておくといよい。

\* 主トリガーと副トリガーに分けて記録する方法もある。この場合、各トリガーを使って何を分析するか、あらかじめ決めておく。「とりあえず記録しておく」はダメ

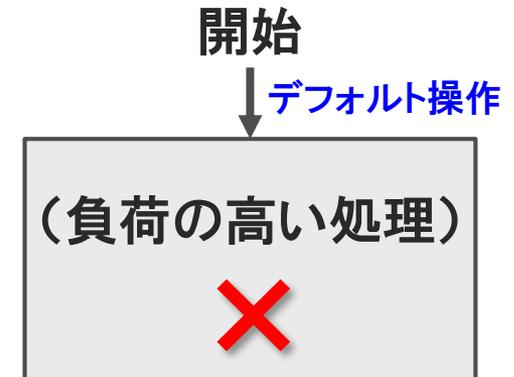
「何機能に対するテストか？」を明確にする\*



テスト観点のとらえ方により、トリガーが変わってしまう

## 問題点

ある機能に対して高負荷下での動作を確認することにした。この場合、操作手順としては特にオプションを変更していないので「基本」となる。もしこれで欠陥が検出された場合、トリガーは「基本」か、あるいは「負荷/ストレス」か？



## 対策

「高負荷下での動作検証」という目的の下にテストを実施しているならば、トリガーは「負荷/ストレス」となる。システムテストにおいては、テストの手順よりも、「テストの狙い目」に着目する。

システムテストのトリガーは、手順よりも目的で考える

## テスト観点のとらえ方により、トリガーが変わってしまう②

### 問題点

ある機能に対して高負荷下での動作を確認することにした。ところがその事前段階で欠陥が検出された。(意図して負荷テストを実施していたが、実際は負荷がかかる処理に入る前段階でNGとなった)

この場合のトリガーは「負荷/ストレス」でよいのか？

開始

↓ デフォルト操作



(負荷の高い処理)

### 対策

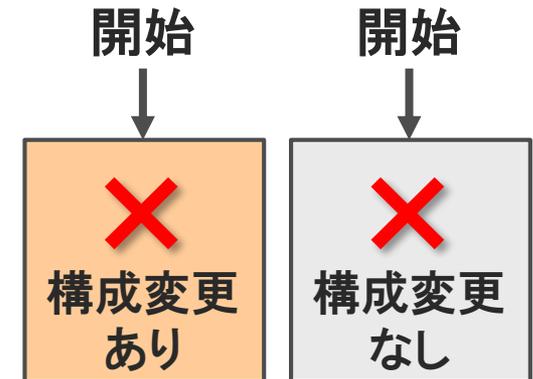
負荷テストという狙いが、基本的な欠陥(結合テストまでに検出されるべき欠陥)により達成できない場合、トリガーは「通常系」となる。もし通常系を採用していない場合は、結合テストとしてトリガーを選択する。いずれにしても、システムテストの狙いが達成されていないので、「負荷/ストレス」にはならない。

システムテストのトリガーは、手順よりも目的で考える

## 検出時に選択したトリガーが直接の原因ではなかった

### 問題点

SW構成に関するテストを実施し、欠陥を検出したのでトリガーに「SW構成」を選択した。ところが欠陥を修正する段階になって、SW構成の変更とは関係なく発生する陥であることが分かった。この場合でもトリガーは「SW構成」とするべきか？



実は構成変更は関係なかった

### 対策

トリガーは基本的に「テストの成熟度」を評価するための属性である。そのため欠陥の原因に影響されず、あくまでも「欠陥が表面化したきっかけ」で決定すればよい。ただし誤ったトリガーが分析上のノイズとなるならば、「真のトリガー」へ修正する。必要であれば起票者が決定した「第1のトリガー」とは別に修正者が決めた真のトリガーを分けて記録する。いずれにしても「トリガーを使って何を分析(改善)するか」次第であるといえる。

分析上のノイズとなるなら修正する

- トリガーの基礎を、しっかり伝えることができた点はよかったと考える。
- 特にレビュー向けのトリガーは、いままで話をする機会が少なかった。
- 研究員との対話の中で、新たな気づきを得ることができた。(第1のトリガーと真のトリガー)
- 今後はODC研究会として、**トリガーFAQ**と、**分析事例**を充実させていきたい。

# ODC分析研究会