

# 探索的テストを対象とする 機械学習(SOM)を利用した 進行中プロジェクトにおける探索箇所推定手法 「FaRSeT-# / ファルセットシャープ」の提案

---

○ 喜多義弘 (長崎県立大学)

上田和樹 (日本ナレッジ株式会社)

櫻井清敬 (日本ナレッジ株式会社)

## About us



長崎県立大学  
UNIVERSITY OF NAGASAKI



### 佐世保校

- 経営学部
- 地域創造学部

### シーボルト校

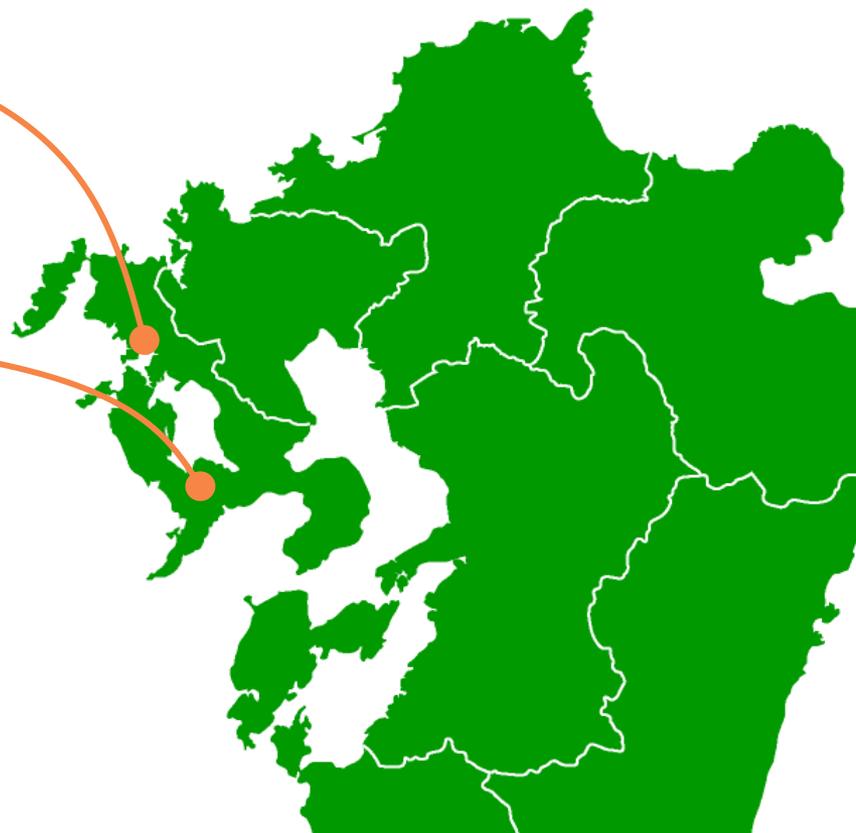
- 国際社会学部
- 情報システム学部
- 看護栄養学部



### 大学院

- 地域創成研究科
- 人間健康科学研究科

- 2008年 旧、長崎県立大学と  
県立長崎シーボルト大学が統合
- 大学：5学部9学科
- 大学院：2研究科4専攻
- 学生数：2,961名（2020年度）



## About us

## 日本ナレッジ株式会社

■本社所在地	〒111-0042 東京都台東区寿3-19-5 JSビル9階
■拠点	札幌事業所、郡山センター、つくばセンター、成田センター、諏訪センター、ホーチミンブランチ（ベトナム）
■設立	1985年10月
■代表取締役	藤井 洋一
■資本金	7,100万円
■従業員数	253名 (2018年2月時点)
■売上	27億円 (2018年度)



## 主な加盟団体

CSAJ 一般社団法人 コンピュータソフトウェア協会

iVIA 一般社団法人 IT検証産業協会



# 開発事業

## シンクライアント/セキュリティ製品



DEFESA Logger

「DEFESA Logger Series」はWindowsOS上で様々な操作履歴ログを生成・取得する製品です。



DEFESA HWL

DEFESA HWL は、勝手にアプリケーションを追加されてしまっているなど、IT 統制でお困りの企業へ提案可能な最適なソリューションです。



DEFESA REC

リーズナブルで且つ導入がカンタンにできる録画ソリューション、それが「DEFESA REC」です。



monoPack

【シンクライアントUSB】在宅勤務などのテレワークでセキュリティを高めたいユーザー向け、小規模ユーザーから大規模ユーザーまで幅広く利用可能。



mono + WEB Separation

mono + WEB Separationは、小規模ユーザー向けに開発された、低コストで導入可能なWEB分離ソリューションです。

## 鋼材業向けパッケージ 「Power Steel」



## 建材、木材卸業向けパッケージ 「Power Cubic」



年	製品名	累計出荷本数
1991	オフコン版鋼材卸売システムリリース	60本
1995	Windows 版 PowerSteel リリース ※更に販売本数倍増!	120本
1999	SMILEαVer.5 版 PowerSteel リリース ※Windows95、98se、NT 対応版。遂に 200 本突破!	200本
2004	SMILEαAD 版 PowerSteel リリース ※Windows2000、XP、2003 対応版。遂に 350 本突破!	350本
2010	SMILEBS 版 PowerSteel リリース	
2011	SMILEBS 版 (加工・見積管理) PowerSteel リリース	450本 (2010~2013累計本数)
2013	SMILEBS 版 (入出荷管理) PowerSteel リリース	

# 検証事業

## 業務系 検証サービス



電子書籍ECサイト



ガス検針



販売管理



スマホアプリ

## 組込系 検証サービス



液晶タブレット



スマートフォン検証

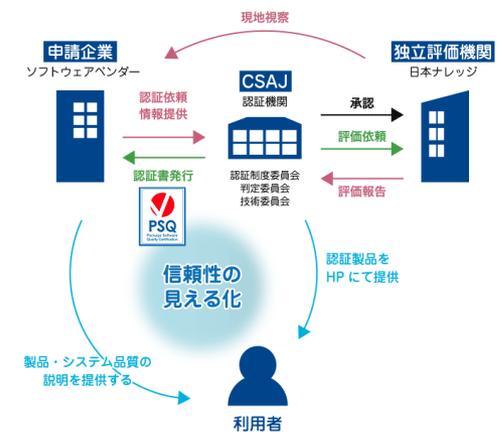


医療機器



ビデオレコーダー検証

## PSQ 評価機関



## 札幌テストセンター



## ISO/IEC 17025試験所



# アジェンダ

---

1. 研究背景
2. FaRSeT-#の提案
3. FaRSeT-#の実用性について

# 1. 研究背景

---

# 短納期型開発プロジェクトの現状

## 短納期型開発プロジェクト

開発から納品までの工期が短いプロジェクト

### 従来の問題点

開発途中で製品の仕様変更が多い。



テストの手戻りが発生し、  
毎回テストケースの修正・再作成を  
行う必要がある。

**大きな負担**

# 問題解決のためのテスト手法「FaRSeT」

## FaRSeT (Flexible and Rapid Software Test)

- マインドマップによる業務分析
- 探索的テストマトリクス

を用いて、**タイムリーに探索箇所を判断しながら**  
柔軟(Flexible)かつ迅速(Rapid)に探索的テストを行う。

**事前にテストケースを準備する必要がなくなり、  
テストの手戻りによる負担を軽減できる**

上田和樹, 丹場順次, 工藤修悟,

“短納期型開発プロジェクトのためのテスト手法「FaRSeT (Flexible and Rapid Software Test)」の適用と効果”,  
ソフトウェア品質シンポジウム2018, pp.1-8, 2018.



# 探索的テストマトリクスの概要

機能	1	2	3	4	5	6	7	8	9	10	11	12
	機能完全性	機能正確性	機能適切性	共存性	相互運用性	適切度認識性	習得性	運用操作性	ユーザエラー防止性	ユーザ/ F 快美性	アクセシビリティ	機密性
A	10	12	9	0	1	1	0	4	5	2	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0
C	5	6	1	0	0	3	0	3	1	0	0	0
D	4	9	1	0	0	0	0	2	0	1	0	0
E	2	7	2	0	0	0	0	2	0	0	1	0
F	3	8	2	0	0	0	0	1	0	3	0	0
G	2	3	2	2	1	1	0	4	1	1	1	0
H	0	3	0	0	0	1	0	0	1	0	0	0
I	9	23	4	0	2	5	0	6	1	2	0	0
J	1	2	0	0	0	0	0	3	3	0	0	0
K	1	2	0	1	0	0	0	0	0	0	0	0

列：  
テストチャータ  
(この例は  
品質特性)

行：  
機能

各項目の交点  
=セッション

数字は  
不具合数

# FaRSeTの手順

1. マインドマップを活用し，業務分析を行う
2. 業務分析をもとに，品質特性からブレークダウンした**テストチャータ**（探索的テストの抽象目的）を作る
3. 「行：機能，列：テストチャータ」とする  
**探索的テストマトリクス**を作る.
4. 探索的テストマトリクスの各項目の交点を**セッション**とし，**ステークホルダの合意を得ながらセッションの選択と決定**し，探索的テストを行う.
5. 不具合を発見したら当該セッションに**不具合数**を記す
6. 手順4, 5を繰り返しながら，探索を深めていく

# これまでの探索箇所の推定方法

不具合数が多く出ている  
**行（機能）と**  
**列（テストチャータ）**を  
 中心に探索箇所を推定



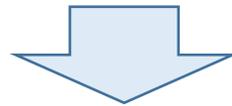
**黄色のセッション**が  
 主な探索候補になる。

機能	1	2	3	4	5	6	7	8	9	10	11	12
	機能完全性	機能正確性	機能適切性	共存性	相互運用性	適切度認識性	習得性	運用操作性	ユーザエラー防止性	ユーザインターフェイス快美性	アクセシビリティ	機密性
A	10	12	9	0	1	1	0	4	5	2	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0
C	5	6	1	0	0	3	0	3	1	0	0	0
D	4	9	1	0	0	0	0	2	0	1	0	0
E	2	7	2	0	0	0	0	2	0	0	1	0
F	3	8	2	0	0	0	0	1	0	3	0	0
G	2	3	2	2	1	1	0	4	1	1	1	0
H	0	3	0	0	0	1	0	0	1	0	0	0
I	9	23	4	0	2	5	0	6	1	2	0	0
J	1	2	0	0	0	0	0	3	3	0	0	0
K	1	2	0	1	0	0	0	0	0	0	0	0

# FaRSeT (を用いた探索的テスト) の問題点

## 探索箇所を選定した**根拠を示す**ことが難しい

- **テスト未実施の箇所**では不具合が発生していない  
→ **重要な探索箇所として判断できない**
- 不具合数だけでは**残存不具合を想定できない**



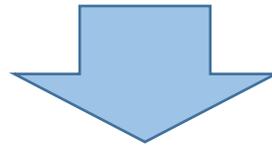
探索分析や探索的テストマトリックスを用いても  
次なる探索箇所の判断は「**担当者の勘や経験則**」

- 個人差が出やすい
- **ステークホルダへの説明が困難**である

# 本提案の目的

**探索的テストにおいて、  
次なる探索箇所を根拠に基づいて示すこと**

- 不具合数だけでなく様々な観点から不具合の傾向を推測する
- テスト未実施箇所でも不具合の傾向から探索の必要性を推測する



## 必要なもの

- 様々な観点を得るための**多様な視点**
- 根拠に基づいた判断をするための**客観性**

## 2. FaRSeT-#の提案

---

# 探索箇所推測手法「FaRSeT-#」

## FaRSeTに**機械学習**を取り入れた強力な拡張（進化）版

機械学習によってセッションを分類し，分類ごとに不具合の傾向を分析することで探索の優先度やその根拠を導き出す。

### 機械学習のメリット

- **複数の観点（パラメータ）**を**一度にまとめて評価**できる
- **客観的な評価**ができる

# FaRSeT-# の手順

1. FaRSeTの手順1～3を行い，探索的テストマトリクスを作る
2. 探索的テストマトリクスをもとに以下の情報を抽出する
  - **行（機能）に関するメトリクス**  
例）機能規模，開発担当者のスキルなど
  - **列（テストチャータ）に関するメトリクス**  
例）品質特性重要度など
  - **セッションに関するメトリクス**  
例）不具合数，直近のテスト結果，テスト実施時間，  
テスト担当者のスキルなど
3. 機械学習に手順2の情報を入力し，セッションを分類する
4. 分類ごとに手順2の情報をもとに不具合の傾向を分析し，次なる探索箇所を推測する

# 機械学習 (SOM)

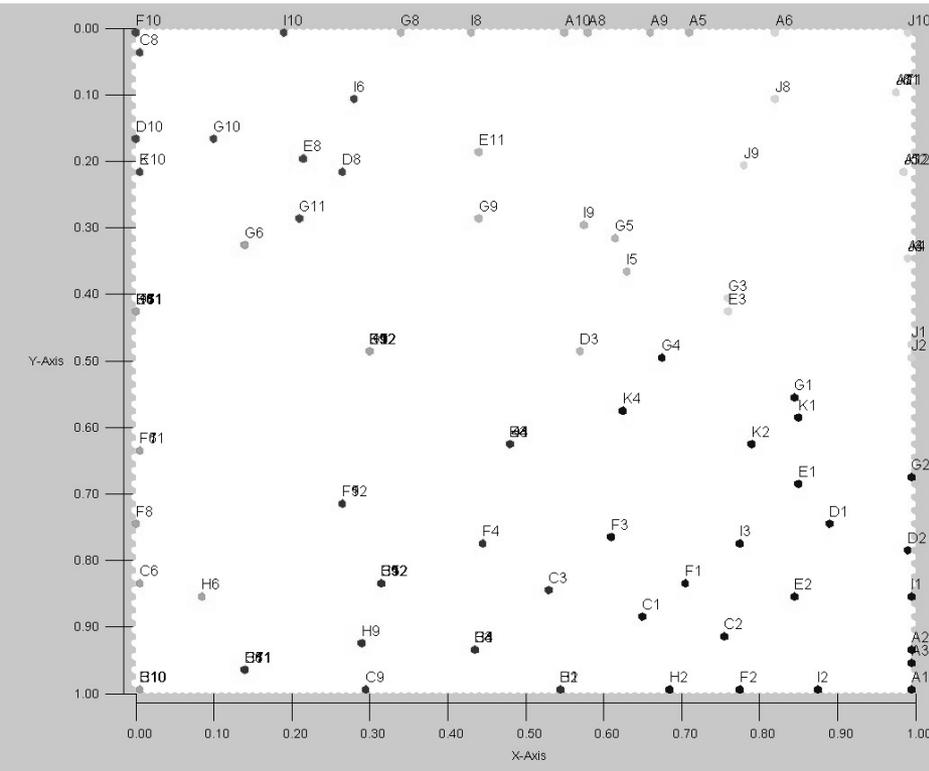
## 自己組織化マップ (Self-Organizing Map : SOM)

- 教師なしの競合学習型ニューラルネットワーク
- 与えられた多次元データの類似度を二次元上の距離として自動的に表すマップ

似ているデータ → 互いに近づく  
 似ていないデータ → 互いに遠ざかる



隣接し合うデータは  
 似ているデータ同士が集まる



スライドp.11にある探索的テストマトリクスの全セッションを入力したSOM

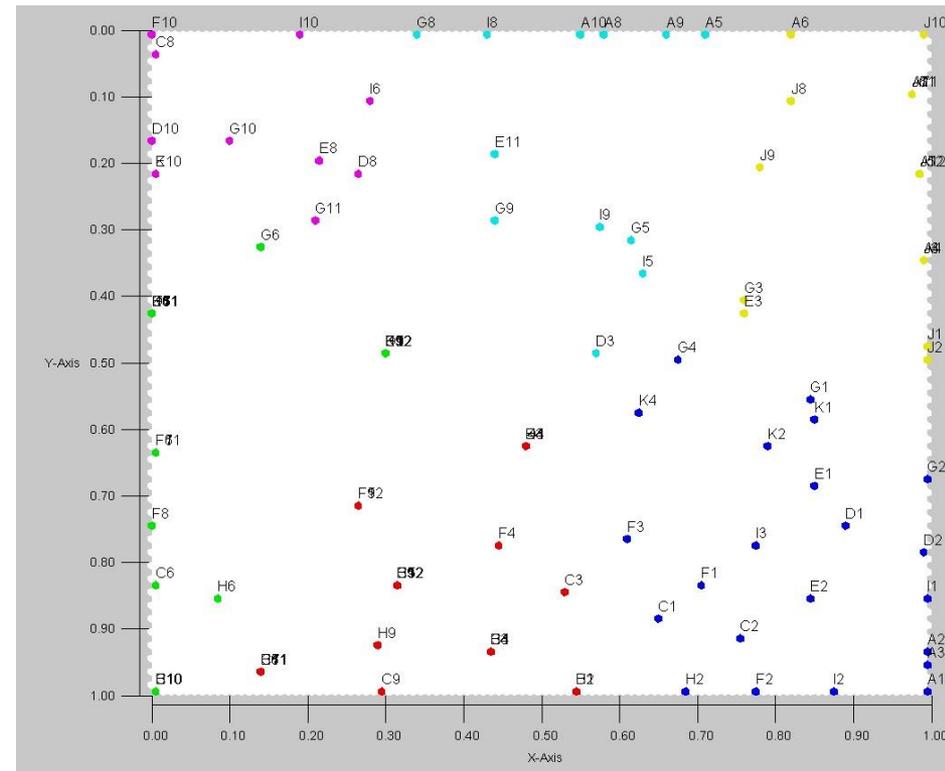
# クラスタリング手法

## k-means法

- 教師なしの非階層型クラスタリング手法
- 平面上に散在するデータを互いの距離によって、いくつかのグループ（クラスタ）に分類する

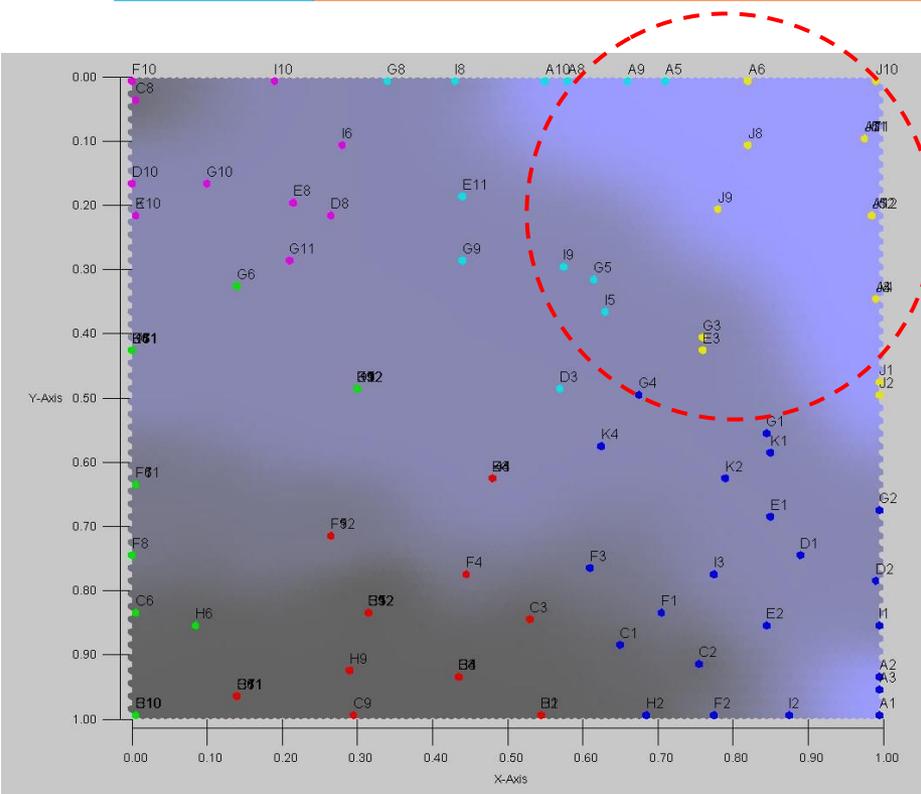
セッションが持つ情報（中身）は全く考慮せず、**マップ上の距離（見た目）のみ**でクラスタに分類する

今回は6つのクラスタに分類し、**RED, BLUE, GREEN, YELLOW, CYAN, MAGENTA**の色を各クラスタに配色する



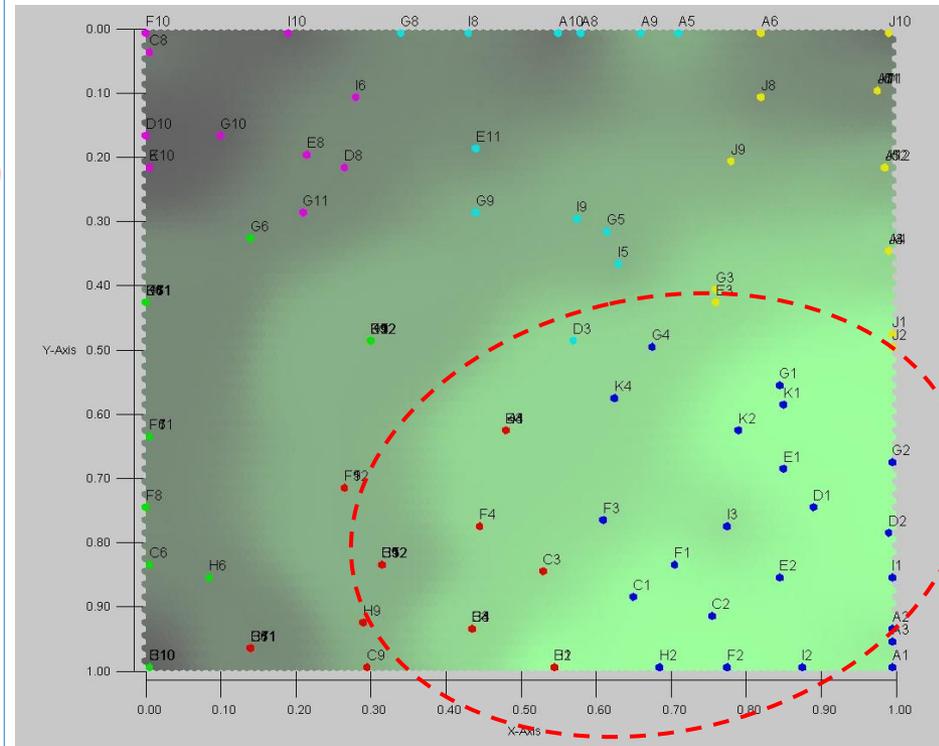
SOM上のセッションをk-meansで分類し、クラスタごとに色分けした結果

# マップの分布内訳 (行・列)



行に関するメトリクス  
(機能規模)

**YELLOW** (マップ右上) 付近  
機能規模が大きい

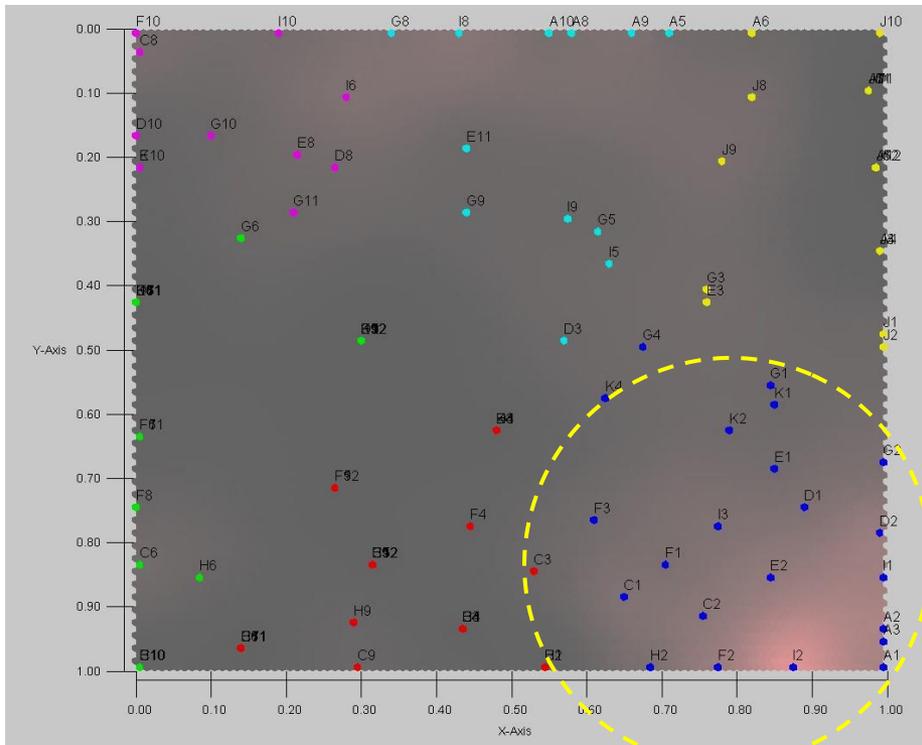


列に関するメトリクス  
(品質特性重要度)

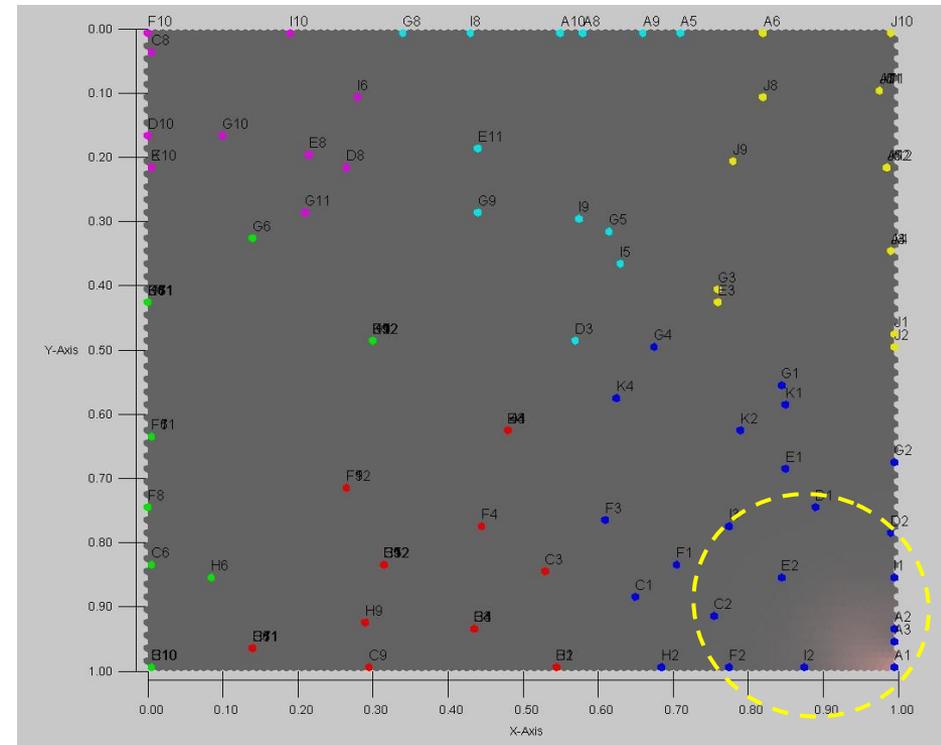
**BLUE~RED** (マップ下) 付近  
品質特性の重要度が高い

**YELLOW, BLUE, RED** は探索の優先度が高いと推測できる

# マップの分布内訳 (セッション)



セッションに関するメトリクス  
(不具合数)



重要不具合件数

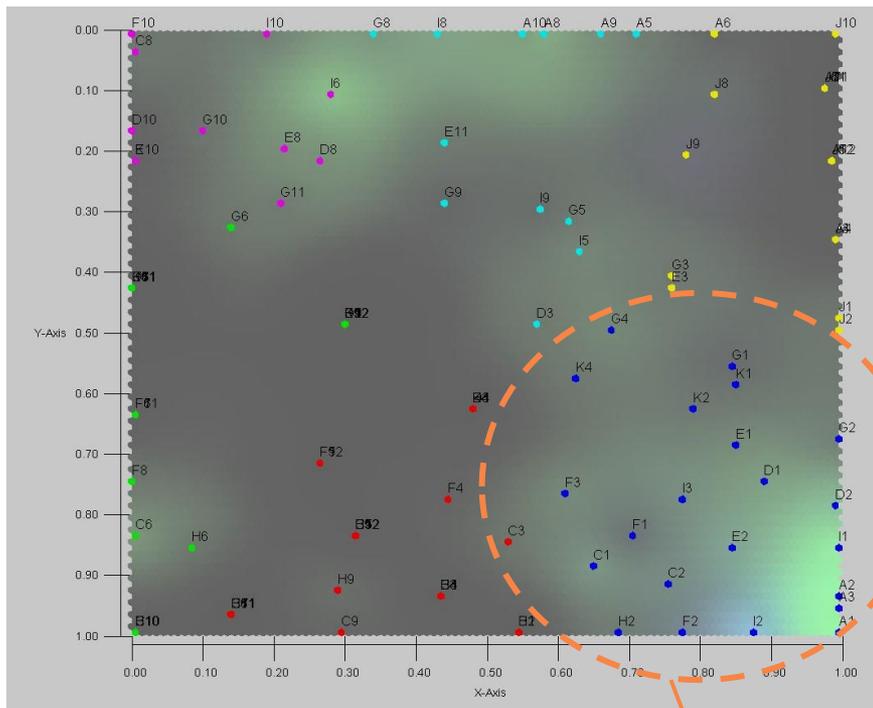
**BLUE** (マップ右下) 付近

不具合数が多く、重要不具合も含まれている

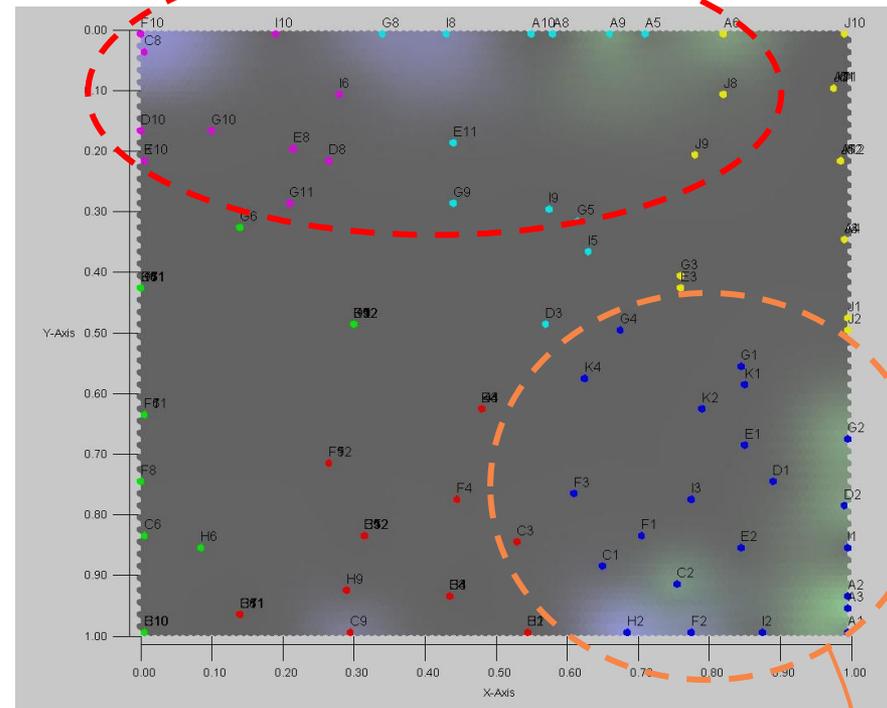
→ **前ページの結果も踏まえ、BLUEが探索の優先度が最も高い**

# テスト担当者別の発見した不具合数の分布

CYANとMAGENTA付近  
優先度は高くないが、**初心者や  
中級者でも不具合を発見している**



熟練者 2 名 (青と緑)



初心者 (青) と中級者 (緑)

BLUE付近  
初心者や中級者は不具合の発見数が少ない  
(ここには**重要な不具合**が含まれる)

# 探索箇所の優先順位

全セッションのクラスタ色分け

機能	1	2	3	4	5	6	7	8	9	10	11	12
	機能完全性	機能正確性	機能適切性	共存性	相互運用性	適切度認識性	習得性	運用操作性	ユーザーエラー防止性	ユーザー/F快美性	アクセスしやすさ	機密性
A	10	12	9	0	1	1	0	4	5	2	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0
C	5	6	1	0	0	3	0	3	1	0	0	0
D	4	9	1	0	0	0	0	2	0	1	0	0
E	2	7	2	0	0	0	0	2	0	0	1	0
F	3	8	2	0	0	0	0	1	0	3	0	0
G	2	3	2	2	1	1	0	4	1	1	1	0
H	0	3	0	0	0	1	0	0	1	0	0	0
I	9	23	4	0	2	5	0	6	1	2	0	0
J	1	2	0	0	0	0	0	3	3	0	0	0
K	1	2	0	1	0	0	0	0	0	0	0	0

探索箇所の優先度（色の濃い順）

機能	1	2	3	4	5	6	7	8	9	10	11	12
	機能完全性	機能正確性	機能適切性	共存性	相互運用性	適切度認識性	習得性	運用操作性	ユーザーエラー防止性	ユーザー/F快美性	アクセスしやすさ	機密性
A	10	12	9	0	1	1	0	4	5	2	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0
C	5	6	1	0	0	3	0	3	1	0	0	0
D	4	9	1	0	0	0	0	2	0	1	0	0
E	2	7	2	0	0	0	0	2	0	0	1	0
F	3	8	2	0	0	0	0	1	0	3	0	0
G	2	3	2	2	1	1	0	4	1	1	1	0
H	0	3	0	0	0	1	0	0	1	0	0	0
I	9	23	4	0	2	5	0	6	1	2	0	0
J	1	2	0	0	0	0	0	3	3	0	0	0
K	1	2	0	1	0	0	0	0	0	0	0	0

優先 

優先 

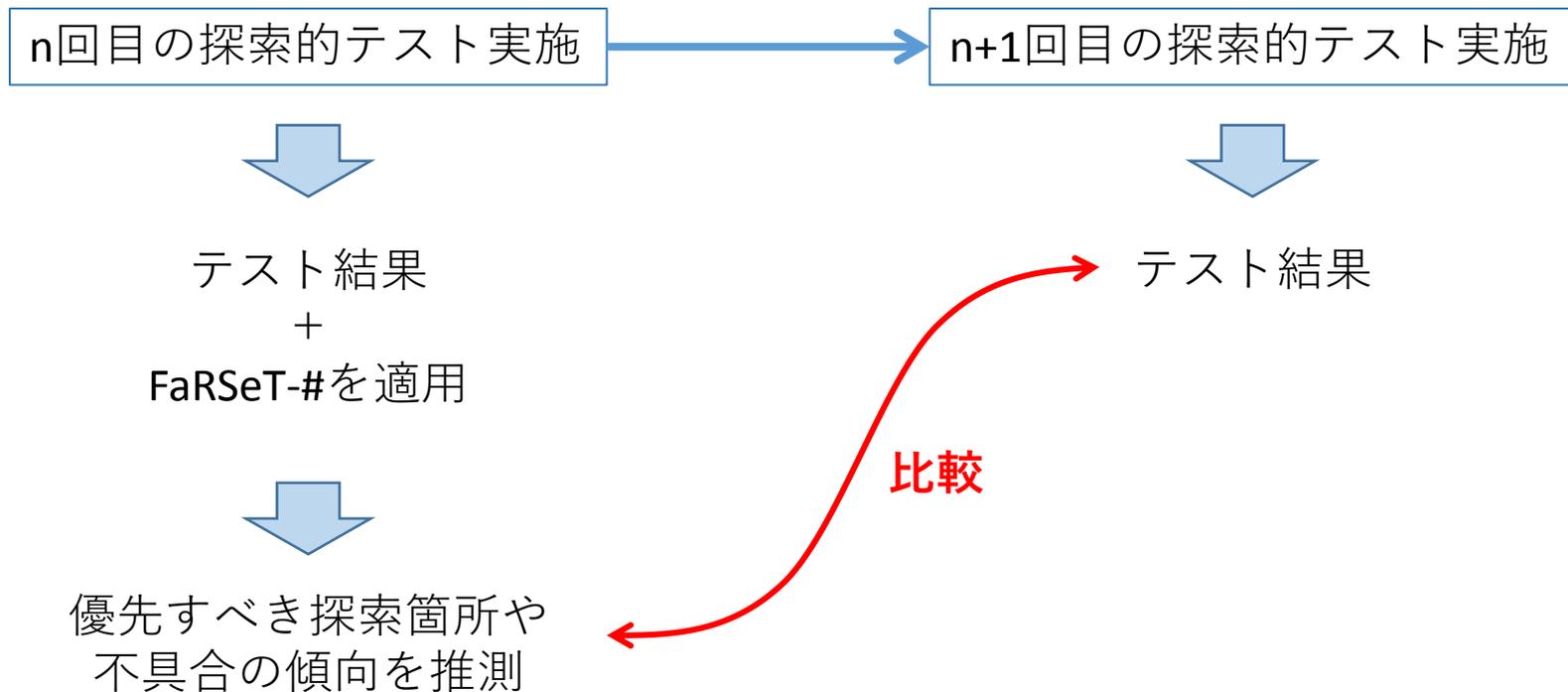
 は初心者や中級者も対応可

## 3. FaRSeT-#の妥当性について

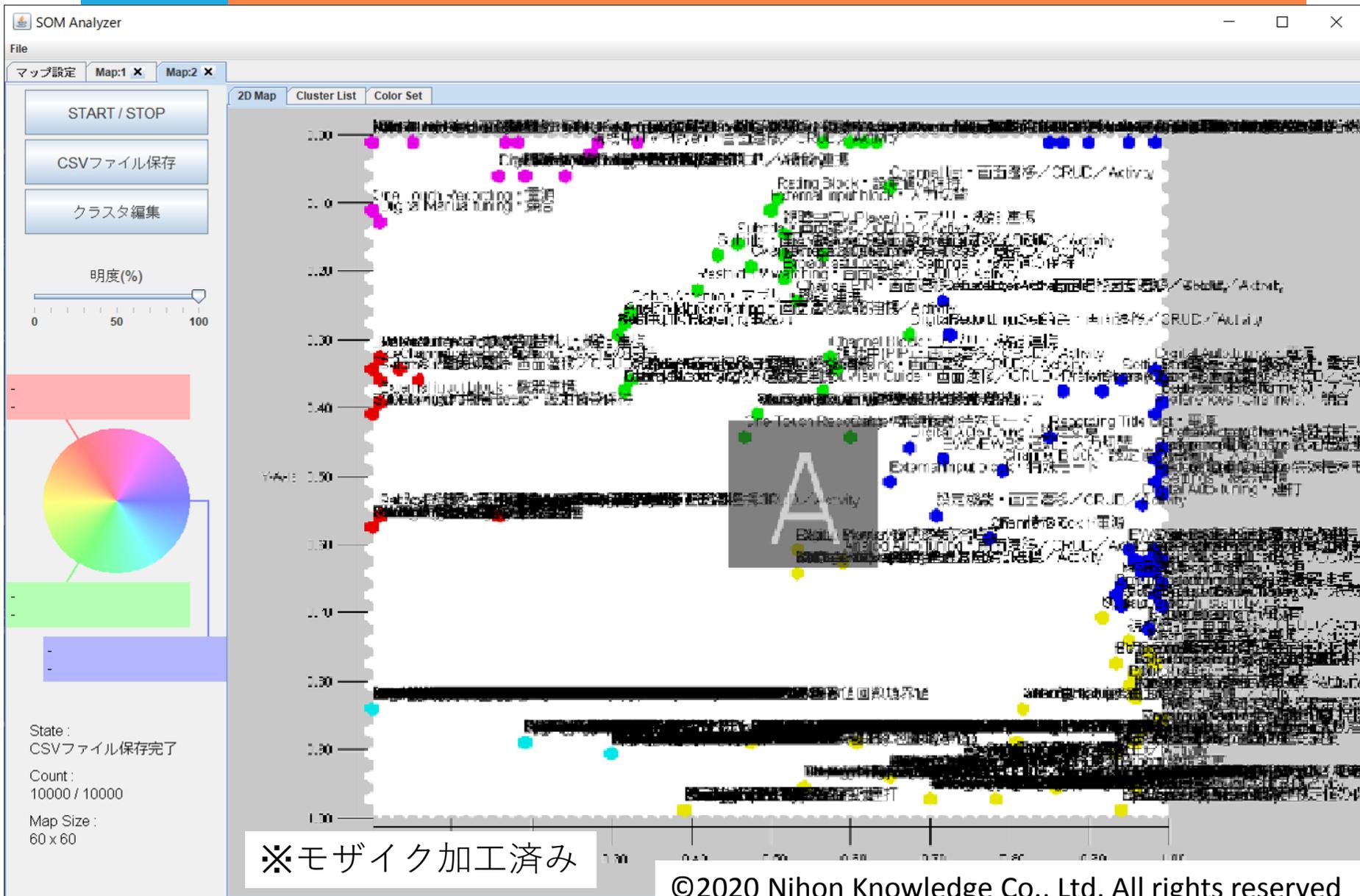
---

# 妥当性の評価方法

進行中のプロジェクトで、実際にFaRSeT-#を用いて推測し、  
次回のテスト結果と比較する。



# テスト結果から生成したSOM





# 探索的テスト n 回目の分析

探索的テストn回目におけるクラスタごとのメトリクス集計結果

クラスタラベル	平均：不具合数	平均：テストスキルAの発見数	平均：テストスキルBの発見数	平均：テストスキルCの発見数	平均：機能重要性 (3>2>1)	平均：機能規模 (3>2>1)	平均：観点優先度 (3>2>1)	平均：新規要件に関連 (1 or 0)	平均：重要不具合件数 (Critical)	平均：重要不具合件数 (Major)	平均：重要不具合件数 (Minor)	平均：修正件数 (Fixed)	平均：工数/h
RED	1.5	1.4	0.1	0.0	2.8	2.8	2.4	0.9	0.0	0.0	0.0	0.5	2.1
BLUE	<b>0.9</b>	0.4	0.4	0.1	2.8	2.6	2.4	0.8	0.2	0.2	0.2	0.3	<b>16.8</b>
GREEN	<b>3.0</b>	1.6	1.4	0.1	2.9	2.9	2.3	0.9	<b>1.7</b>	<b>1.7</b>	0.0	1.9	9.1
YELLOW	0.0	0.0	0.0	0.0	2.5	2.7	2.0	0.8	0.0	0.0	0.0	0.0	2.3
CYAN	0.0	0.0	0.0	0.0	2.6	2.6	0.9	0.8	0.0	0.0	0.0	0.0	0.0
MAGENTA	1.7	0.3	0.2	<b>1.2</b>	2.9	2.6	2.3	0.8	0.7	0.7	0.0	0.8	4.7

推測

- GREENは不具合数・重要不具合件数が最も多い  
→ **優先的に探索する箇所である**
- MAGENTAはテストスキルCの発見数が多い  
→ **中級者以下でも担当可能である**
- BLUEは平均工数が多い割に不具合数が少ない  
→ **これ以上は不具合が出にくい**

# n + 1 回目のテスト結果

探索的テストn+1回目を実施した際のメトリクス集計結果

クラスタラベル	平均：不具合数	平均：テストスキルAの発見数	平均：テストスキルBの発見数	平均：テストスキルCの発見数	平均：重要不具合件数 (Critical)	平均：重要不具合件数 (Major)	平均：重要不具合件数 (Minor)	平均：修正件数 (Fixed)	平均：工数/h	時間当たりの検出件数
RED	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.94	0.00
BLUE	<b>0.06</b>	0.00	0.06	0.00	0.00	0.06	0.00	0.00	4.00	<b>0.02</b>
GREEN	<b>0.50</b>	0.25	0.25	0.00	<b>0.25</b>	<b>0.25</b>	0.00	0.08	5.65	0.09
YELLOW	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.15	0.00
CYAN	0.09	0.00	0.09	0.00	0.09	0.00	0.00	0.00	2.43	0.04
MAGENTA	0.11	0.00	<b>0.11</b>	0.00	0.11	0.00	0.00	0.00	3.72	0.03

## 結果

- GREENは**不具合数・重要不具合件数が最も多い** → **推測どおり**
- MAGENTAは**テストスキルBの担当者が発見している** → **推測どおり**
- BLUEは不具合は検出しているが、時間当たりの検出件数は少ない  
→ **ほぼ推測どおり**

**FaRSeT-#による探索箇所推測は妥当性がある**

# まとめ

---

## 目的

探索的テストにおいて、  
次なる探索箇所を根拠に基づいて示すこと

## 提案

機械学習(SOM)を利用した探索箇所推測手法「FaRSeT-#」  
を提案した

## 評価

FaRSeT-#による探索箇所推測の妥当性を検証した結果、  
妥当性があることを確認した

**本目的において、FaRSeT-#は有用である**

## Special Thanks

データ提供&協力

吉田 絵理 (日本ナレッジ株式会社)

金丸 優介 (日本ナレッジ株式会社)

ご清聴ありがとうございました

---