

堀川 透陽 (Toyo Horikawa)

- ・ 1978年生まれ 大阪府在住

【所属】

株式会社ベリサーブ 西日本事業部

- ・ 2007年からテスト業務に携わり、2017年ベリサーブに入社
- ・ 社内のTPI NEXT研究会メンバー
- ・ 好きな言葉はないが、嫌いな言葉は「ルーチンワーク」
- ・ 現在、検証コンサルとして活動中

- 実際の案件に導入した資料を基に作成した
- 結合テストのテスト戦略は**リスクベースドテスト**とした
- リスク分析に市場データの分析結果をインプットした
- データの解析には**テキストマイニング**を使用した
- テキストマイニングは仮説に基づいて使用した
- ツールはフリーソフトの「KH Coder」を使用した
- HAZOPのガイドワードから抽出語を導いた
- ソフトウェア品質特性から抽出語を導いた
- 解析結果は数の多さより中心性を意識した

- 派生開発にて、結合テスト以降のテスト工程を担当
- 開発スケジュールは五里霧中
- テストリソースは少ない
- 市場リリース後のトラブルの多さが悩み
- 過去不具合は管理出来ていない状態

テスト計画を作成

- テスト戦略は**リスクベースドテスト**
テストリソースに合わせる柔軟性を持たせ、効率的にテストする
- 過去データから脆弱性や傾向を掴み、
市場リリース後の問題の出にくい品質を目指す

解析データ準備

- 解析のプランニング
データの選別と得られる結果の仮説

解析ツール準備

- ツールの設定
ノイズ(不要語)の除去、辞書登録
- コーディング・ルールによる仮説検証
HAZOPガイドワード、ソフトウェア品質特性から導く

テキストマイニング データ解析

- データを俯瞰的に見る
- データ解析
固有ベクトル中心性・次数中心性、サブグラフ

リスクベースド テストへの展開

- リスク分析へのインプット
FL表(因子・水準表)への展開
機能結合マトリクス
ボトムアップテスト
観点モデルベーステスト

なぜ、テキストマイニングを使用したのか

- 大量のデータを一件一件分析することは困難
- データを俯瞰的かつ客観的に見たい

マッチング

テキストマイニングとは
自然言語からなる文章を単語レベルで分解し(形態素解析)、
出現頻度や言葉の繋がり(ネットワーク分析)から情報を取り出す技術

現状で一度も分析されていないデータに対して有効と考えた
⇒今後はODC分析などタグ付けによる管理と並行しての使用を検討

解析のプランニング

周囲のテキストマイニングを使用したテストエンジニアの声は・・・
「どう分析すればいいかわからない」
「解析した結果をどう使うかわからない」



テキストマイニングという技術への過剰な期待

- ・「銀の弾丸」にはならない
- ・分析の基本は仮説の検証
⇒やみくもに使ってどうにかなるものではない



解析にも計画を立ててみる

- ①どのデータを使用するか
- ②解析結果から得られる仮説を立ててみる

不具合報告とユーザーからのトラブル報告書、どちらを解析対象にすべきか

■表 実践例におけるデータ管理状況

データ種別	管轄	今回の状況
不具合報告	開発チーム	・テストプロセスは未成熟である ・管理はプロジェクト毎に異なる ・発生事象は詳細に記載しているが分析はされていない
トラブル報告書	ユーザーサポート	・市場リリースされた全てのデータが一元管理 ・SOAP形式で分類できている ・俯瞰的に見てはいない

テストプロセスが未成熟であることから、
テストに偏りがある＝不具合へのトリガーも偏るのでは？ ⇒仮説

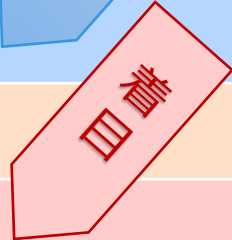
【結果】 俯瞰的に見ることの出来るトラブル報告書を解析対象に決定

解析結果から何が得られるか

ユーザーサポートは問い合わせ内容やクレームをSOAP形式で管理している

■表 SOAP形式の説明

分類		説明
S (Subjective)	主観的情報	ユーザからの問い合わせ内容、クレーム
O (Objective)	客観的情報	ユーザサポートからの情報
A (Assessment)	考察・評価	トラブルに対する考察
P (Plan)	対処・計画	トラブルへの対処



データ解析にて得られる効果

分類	品質スコープ	仮説
S (Subjective)	外部品質	トラブルになりやすい操作、製品に対する印象から使用性の向上に繋がるのではないかな
A (Assessment)	内部品質	製品の脆弱性を把握し、機能性が向上するのではないかな

テキストマイニングツール

今回の市場データの解析に用いたテキストマイニングツール、
及び参考書籍を紹介する

テキストマイニングツール

フリーソフト「KH Coder」 <http://khcoder.net/>

参考書籍

「社会調査のための計量テキスト分析」(樋口耕一 著)

「やってみようテキストマイニング」(牛澤賢二 著)

ツールの設定

ノイズの除去、強制語の登録

データをそのまま解析すると、期待と裏腹な結果になることが多い
表示させたくない不要な語句は使用しない設定に、
業界や現場でのみ通用する語句は辞書(強制語)として登録しておく

※例「不具合」「異常」

⇒不具合や異常があって報告しているので当たり前であり、不必要な語句

■表 「KH Coder」 前処理画面(一部)

分析に使用する語の取捨選択

parts of speech	force pick up	force ignore
品詞による語の選択 :	強制抽出する語の指定 :	使用しない語の指定 :
☒ 名詞	(複数の場合は改行で区切る)	(複数の場合は改行で区切る)
☒ サ変名詞	---cell---	---cell---
☒ 形容動詞		

Appendix「品詞について」

データ解析に際して、
解析対象とする品詞を絞り込む
ことで、読み取りやすさが向上
します(データの抽象化)
名詞・サ変名詞・動詞の3つ以外
を一度非表示にしてみてください

「KH Coder」では解析者の仮説検証を行う為に、オリジナルの分類を作成する機能がある

コーディングルール記載例

*野球 ピッチャー or バッター or 甲子園

*サッカー ゴールキーパー or ミッドフィルダー or 国立競技場

⇒ 抽出語「ピッチャー」「バッター」「甲子園」は「**野球**」として分類される

テスト観点で分類できれば、

問題の発生しやすい観点と機能の組み合わせが見えてくるかも ⇒仮説



ガイドワードに着目、既にある定義を利用する

① **HAZOP ガイドワード**

② **システム及びソフトウェア品質モデル (JIS X 25010-2013)**

⇒ 上記のガイドワード、定義分類から抽出語を導いてみる

コーディング・ルール①HAZOP ガイドワード

■表1 HAZOP ガイドワードの説明

ID	分類	誘導語	説明
G 1	存在 Existence	無 No,Not	設計で意図することが起こらない
G 2	方向 Direction	逆 Reverse	論理的反意： 設計の意図と反したことが起きる
G 3		他 Other than	置換： 設計の意図と異なることが起きる
G 4	量 Quantity	大 More	量的増加： 設計で意図した値を上回る
G 5		小 Less	量的減少： 設計で意図した値を下回る
G 6	質 Quality	類 As well as	質的增加： 設計で意図したことは達成されるが余分なことも起きる
G 7		部 Part of	質的減少： 設計で意図したことの一部しか達成されない
G 8	時間 Time	早 Early	設計で意図した時期・タイミングより早い
G 9		遅 Late	設計で意図した時期・タイミングより遅い
G A	順番 Order	前 Before	設計で意図した順番の前に処理してしまう
G B		後 After	設計で意図した順番より後で処理してしまう

抽出方法
ブレンストーミング
類語辞典 など

■表2 ガイドワードから導いた抽出語

存在	方向		量		質		時間		順番	
無	逆	他	大	小	類	部	早	遅	前	後
いなくなる	あべこべ	アクシデント	溢れる	浅い	余り	ある程度	あっという間に	動かない	奪う	後から
失う	裏返し	あぐねる	以上	至らない	行き過ぎ	一部	いきなり	延滞	追い越す	後に
打ち消す	裏目に	あらぬ	上ぶれ	今一つ	浮いた	限られた	急ぎ過ぎ	延長	置いてきぼり	後になって

コーディングルール②システム及びソフトウェア品質モデル

■表1 JIS X 25010:2013 システム/ソフトウェア製品品質特性

品質特性	説明	結合テスト	システムテスト
機能適合性	ニーズを満足させる機能を製品が提供できるか	機能名 = 抽出語となる為、対象外	
性能効率性	指定される条件で使用する資源の量に関係する性能の度合い	—	○
互換性	同じ環境を共有する間、製品・システムが他の製品と交換することができるか	○	—
使用性	ソフトウェアがユーザにとって使いやすいか	—	○
信頼性	指定される条件下でソフトウェアがパフォーマンスレベルを維持する能力	○	—
セキュリティ	製品が情報及びデータを保護できるか	○	—
保守性	障害の発生、環境の変更などが発生した際、保守(修正)がやりやすいか	○	—
移植性	他の環境にシステム・製品を効率良く移すことができるか	○	—

テスト工程別で狙う
品質特性を特定しておく

抽出方法
ブレンストローミング
類語辞典 など

■表2 ソフトウェア品質特性から導いた抽出語(一例)

機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性
	CPU	Win-Win	UI	MTBF	DoS攻撃	EoC	アップデート
	応答	合性	アクセシビリティ	RAMS規格	F5アタック	カプセル化	アンインストール
	遅い	相性	色	UPS	VPN	MTTR	移管
	遅さ	レガシー	音	安心	XSS	テストモード	移行

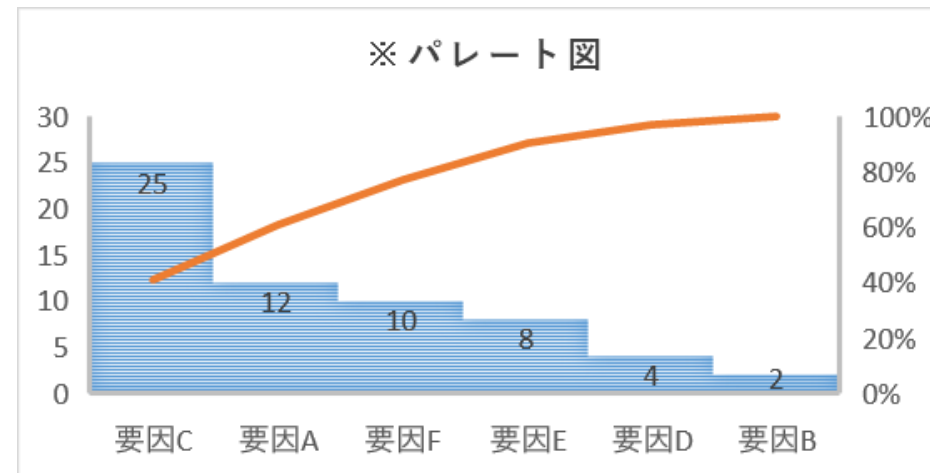
準備完了

データを俯瞰的に見る

データを俯瞰的に見る、とはどういうことか考えてみる

例えば、不具合分析では、
分類項目を設けパレート図を作成する

⇒「80:20の法則」の仮説に基づく
俯瞰的に見る一つの手法



【疑問】 数の多さは問題の“本質”と言えるだろうか

⇒ 数の多さは経験則でも、ある程度掴めていたりする



「中心性」にて視点を変えてみる

データを俯瞰的に見る 中心性に着目する①

テキストマイニングでは、
分解した**言葉同士の繋がり**を可視化出来る(ネットワーク分析)
この分析において中心を示す「**中心性**」に着目してみる

■表 中心性の説明

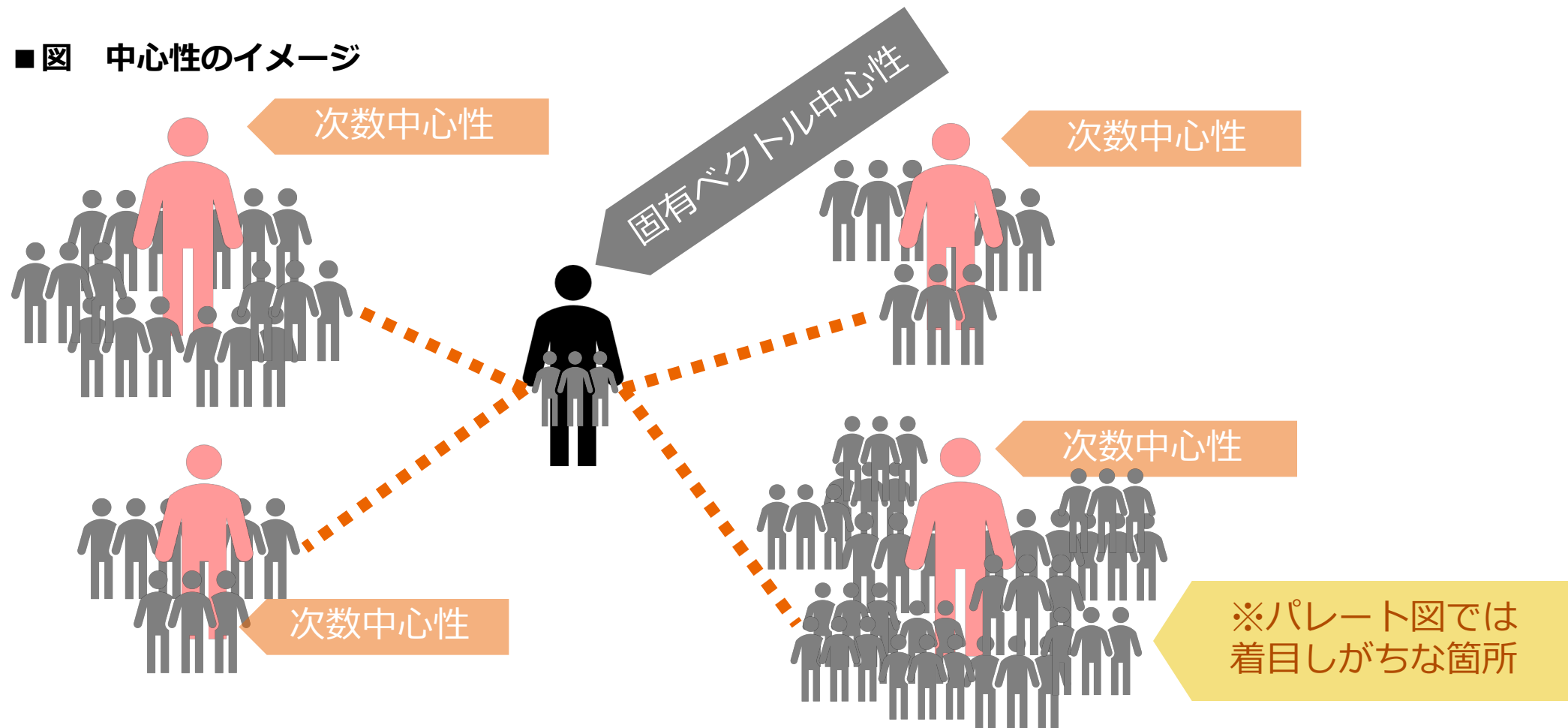
中心性	説明	SNSで例えると...	仮説
固有ベクトル 中心性	中心性の高いノードとの 連結度合いを示す	Twitterでフォロワーの多い人と 相互フォローしている人 ⇒この人自身のフォロワーが多いとは限らない	黒幕 (問題の起点)
次数中心性	隣接するノードの多さを 中心とする	Twitterでフォロワーの多い人	小悪党のボス お山の大將 (共通する問題)

次頁イメージ図

データを俯瞰的に見る 中心性に着目する②

それぞれのグループの次数中心性、それらを繋ぐ固有ベクトル中心性は、**リスクが高い**=テストの優先度を高く設定すべき、と考えた

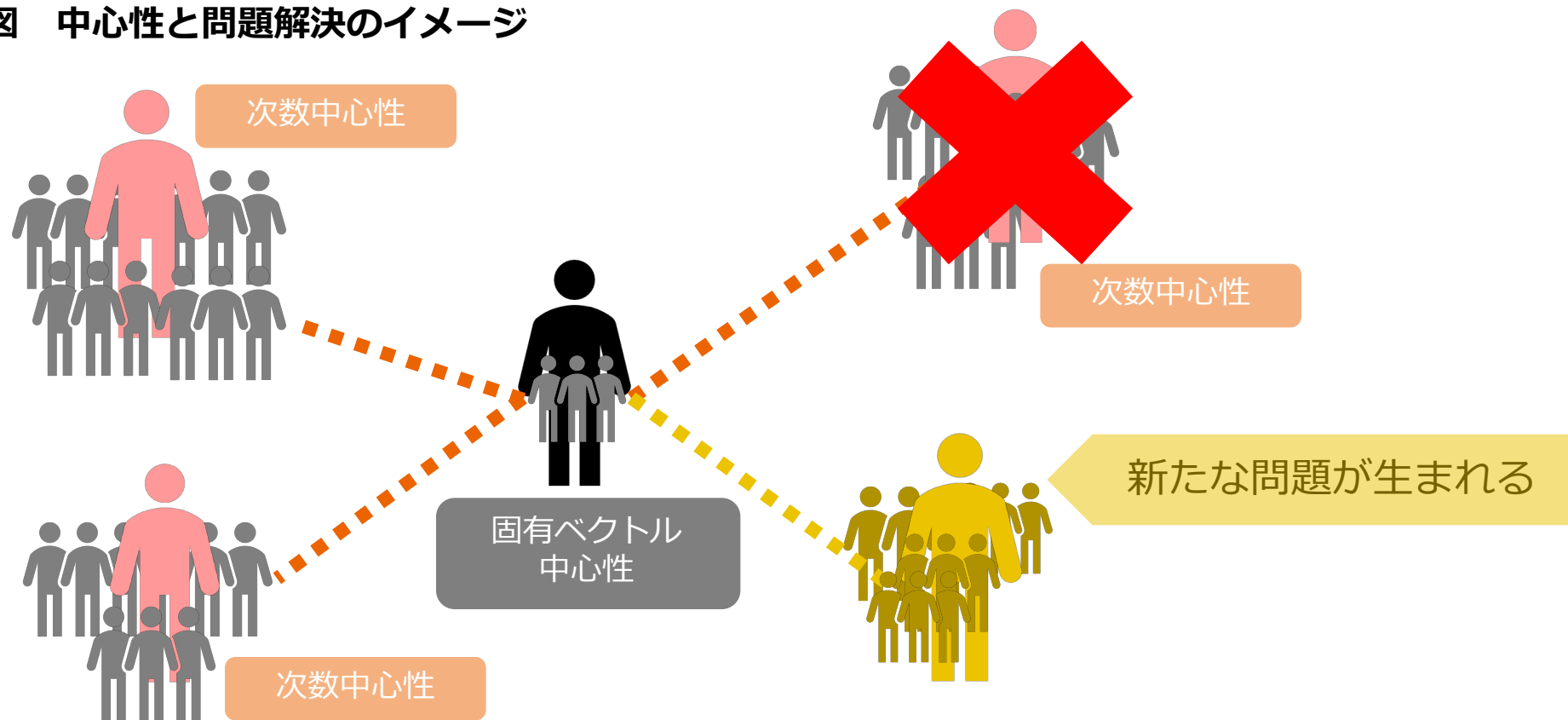
■図 中心性のイメージ



問題は、根本となる要因を解決しない限り、新たな問題が発生するリスクを残す
問題解決にあたり、起因となる要素をまず潰す

⇒固有ベクトル中心性を最も高いリスクに設定する理由

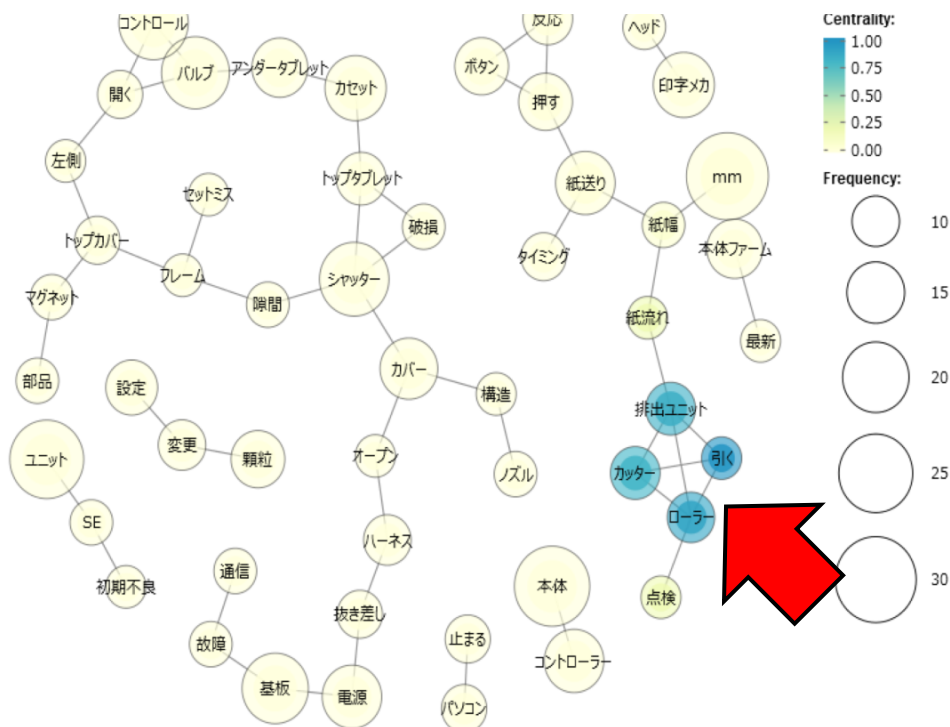
■図 中心性と問題解決のイメージ



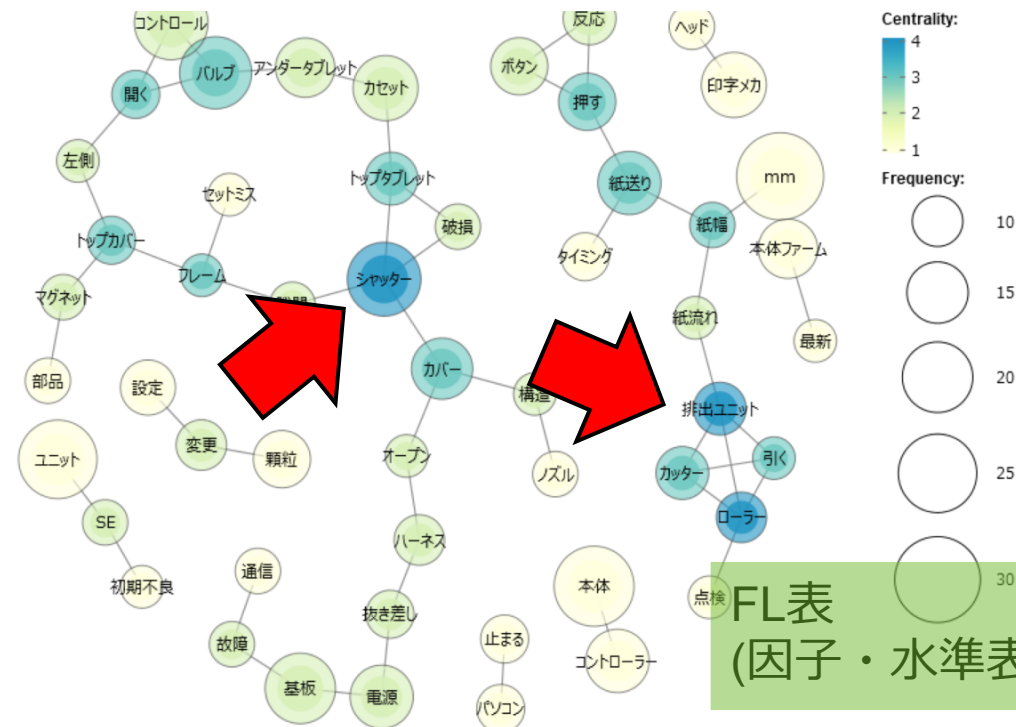
KH Coderにて、分類Aの全データを対象に、中心性を変えて表示した結果、固有ベクトル中心性と次数中心性は異なる言葉を示した

中心性を高さを示す(色の濃い)言葉を、**リスクの高い機能・動作とする**

固有ベクトル中心性(分類Aの解析結果)



次数中心性(分類Aの解析結果)

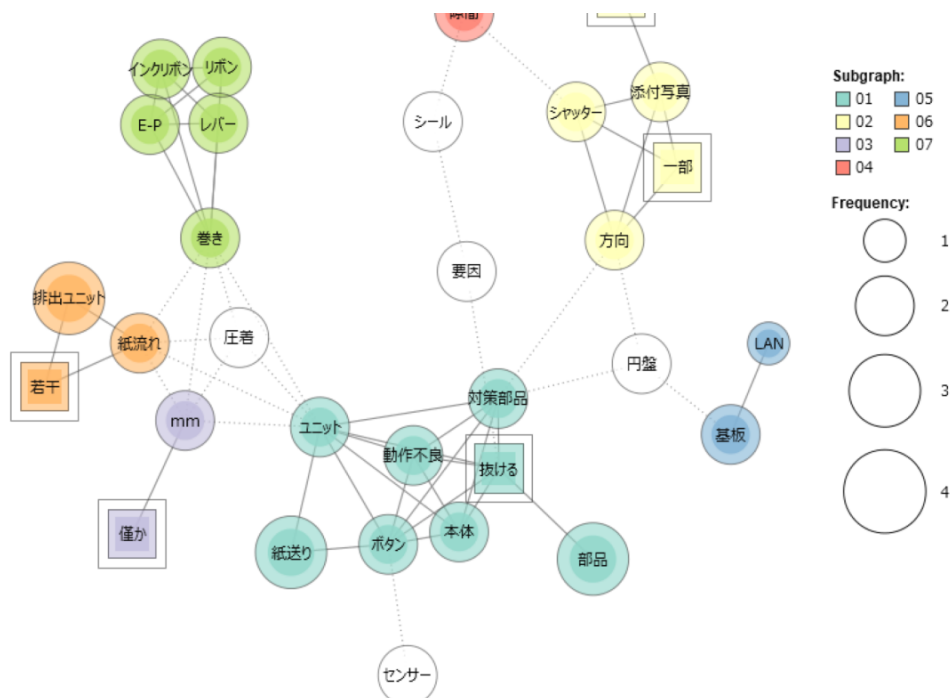


FL表
(因子・水準表)へ

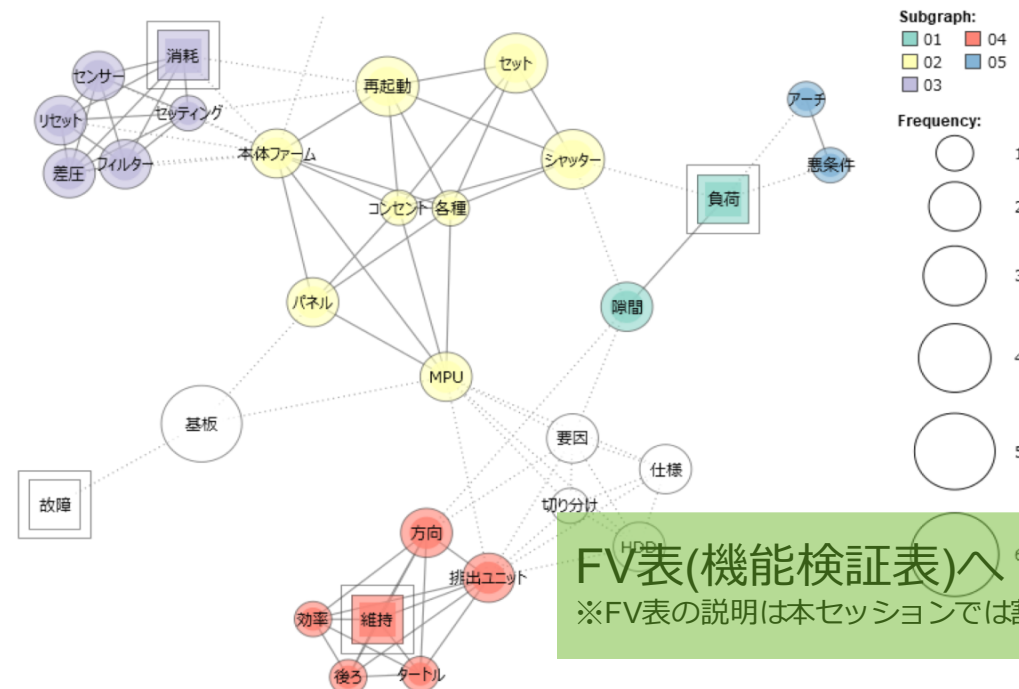
テキストマイニングツールの自動グルーピング機能(サブグラフ)を利用する
ガイドワードと関連する機能・動作を同色でグルーピングしている(観点のモデリング)

抽出された**ガイドワード**と関連する**機能・動作**をテスト設計の**インプット**とする

HAZOPガイドワード



ソフトウェア品質特性



FV表(機能検証表)へ
※FV表の説明は本セッションでは割愛

テキストマイニングで解る中心性の高さの開発区分、それぞれにリスク値を設定する

■表1 中心性に設定するリスク値

反映先	リスク判定	リスク値	説明
因子	大	3	問題の起点となる抽出語【固有ベクトル中心性】に該当する因子、及び水準
	中	2	共通性の高い抽出語【次数中心性】に該当する因子、及び水準
	小	1	問題の起点に直接的な繋がりを持つ抽出語に該当する因子、及び水準
	微	0	上記に該当しない他の因子、及び水準

■表2 開発区分に設定するリスク値

反映先	リスク値	説明
因子	7	新規機能
	4	仕様変更
	2	新規機能、仕様変更に関連する機能
	1	変更なし(要求仕様に記載あり)
	0	変更なし(要求仕様に記載なし)

開発区分で新規機能が7点である理由

- ・新規機能は市場データにない為
リスクとして抽出されない
- ・上記から、既存の機能で高リスクが
多くある場合、新規機能が
優先度に埋もれるリスクを回避する

テキストマイニングで抽出した中心性の高い言葉を、FL表(因子・水準表)と紐付ける

■表1 解析結果まとめ例

解析結果					リスクベースドテスト	
解析方法	可視化方法	関連語	抽出語	説明	リスク判定	リスク値
共起ネットワーク	固定ベクトル中心性	全体	排出ユニット	問題の起点	大	3
		全体	引く	問題の起点に対する動作	大	3
		全体	カッター	問題の起点	大	3
		全体	ローラー	問題の起点	大	3
	次数中心性	全体	シャッター	共通性の高い語句	中	2
		全体	紙送り	共通性の高い語句	中	2

後述で
説明

■表2 FL表(因子・水準表)

No	大項目	中項目	小項目	因子		パラメータ			
1-1	起動機能	基本起動	—	電源スイッチ	電源状態	ON	OFF		
					状態ランプ	消灯	橙色点灯	青色点灯	
				コントローラ電源	電源状態	ON	OFF		
					状態ランプ	電源ランプ	アクセスランプ	バッテリーランプ	
					電源ランプ	消灯	緑色点灯		
					アクセスランプ	橙色点灯			
					バッテリーランプ	消灯	橙色点灯	緑色点灯	赤色点滅

中心性の高い抽出語に該当する
因子を探し、紐付ける

紐付けた因子に該当するリスク値を当てはめる
開発区分のリスク値と合わせ、因子毎のリスク値として使用する

■表1 解析結果まとめ例

解析結果					リスクベースドテスト	
解析方法	可視化方法	関連語	抽出語	説明	リスク判定	リスク値
共起ネットワーク	固定ベクトル中心性	全体	排出ユニット	問題の起点	大	3
		全体	引く	問題の起点に対する動作	大	3
		全体	カッター	問題の起点	大	3
		全体	ローラー	問題の起点	大	3
	次数中心性	全体	シャッター	共通性の高い語句	中	2
		全体	紙送り	共通性の高い語句	中	2

■表2 FL表(因子・水準表)

No	大項目	中項目	小項目	因子		パラメータ(水準)					リスク分析		
											市場データ 分析	開発区分	
1-1	起動機能	基本起動	—	電源スイッチ	電源状態	ON	OFF					3	3
					状態ランプ	消灯	橙色点灯	青色点灯					1
				コントロー ラー電源	電源状態	ON	OFF					3	1
					状態ランプ	電源ランプ	アクセスランプ	バッテリーランプ					1
					電源ランプ	消灯	緑色点灯						1
					アクセスランプ	橙色点灯						3	1
					バッテリーランプ	消灯	橙色点灯	緑色点灯	赤色点滅				2

リスク値の設定されたFL表から、機能結合マトリクスを作成
結合テスト(ボトムアップテスト)のテスト対象の策定に利用する
リスクスコアの合計を**テストの優先度**として使用する

■表 機能結合マトリクス

目的機能		リスク スコア	目的機能に対して結合する機能(ドライバ)				
			1	2	3	4	5
			AAA 機能	BBB 機能	CCC 機能	DDD 機能	EEE 機能
			7	3	1	5	1
1	AAA 機能	7	—	10	—	—	—
2	BBB 機能	3	10	6	4	8	4
3	CCC 機能	1	8	4	2	6	2
4	DDD 機能	5	12	—	6	—	6
5	EEE 機能	1	8	4	2	—	2

運用例

新規機能のリスク値である
7点以上をテスト対象とする

リスクスコアの高い組み合わせから
FV表(機能検証表)へ

※FV表の説明は本セッションでは割愛

- テキストマイニングは大量のデータを俯瞰的に見ることができるツールとして有効
- ネットワーク分析に於ける中心性の定義を利用することで、問題の本質(根本)を捉えることができる
- データ解析はプランニングを行い、データから何を得るか仮説を立てることが重要
⇒仮説に対しての結果分析は必須
- ガイドワードを利用し、観点ベースのモデリングも可能

**テキストマイニングは、リスクベースドテストにおける
リスク分析の手法として有効だと感じている**

長時間のご清聴を頂き、
ありがとうございました