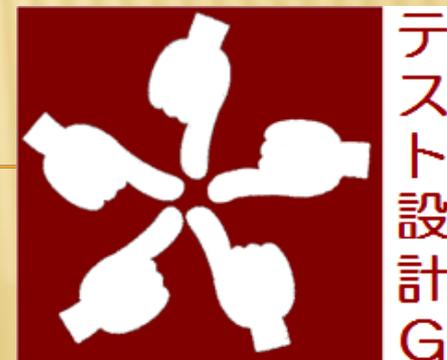


マインドマップを適用した ソフトウェアテストプロセスの構築

2014/2/28

29 SQiP研究会
第5分科会 テスト設計G

主査 : 奥村 有紀子 (有限会社デバッグ工学研究所)
副主査 : 秋山 浩一 (富士ゼロックス株式会社)
副主査 : 堀田 文明 (有限会社デバッグ工学研究所)
研究員 : 来海 大輔 (株式会社イシダ)
田中 桂三 (オムロン株式会社)
中嶋 良秀 (株式会社ノーリツ)
平岡 岳 (株式会社リンクレア)
村上 仁 (株式会社ニコン)



はじめに

～何を伝えたいのか～

研究目標

～問題解決に向けた目標設定～

研究内容

～これがメイン～

研究内容の検証

～問題解決できるのかを検証～

目標の達成度と、今後の展開

～目標にどれだけ達成したか、今後の取り組みは～

はじめに

～何を伝えたいのか？～

ブラックボックステストでの マインドマップの適用方法を提案します

ブラックボックステストでは、**機能&テスト観点抽出**が重要
“**マインドマップ**” は手軽に観点が抽出できる

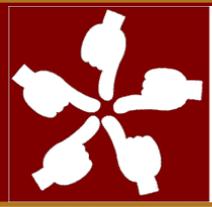
しかし、マインドマップを導入するだけでは問題あり

- ・余計な観点が増え、テストケースが発散する
- ・抽出結果を整理しなければ、テストに使えない

そこで、以下を研究成果として構築

- ①テストケース発散抑制のため、**目的別のマインドマップ**の作成方法
- ②マインドマップでの**抽出結果の整理方法**
- ③マインドマップを適用した**テストプロセスの構築**

「マインドマップを十分活用できていない」と思っている人へ
⇒実業務で是非試してください！



マインドマップの選定理由

テスト対象の機能とテスト観点を抽出するための手段として、
“マインドマップ” を選定

【理由】

SQIP研究会第5分科会活動で、マインドマップ（※）を学習

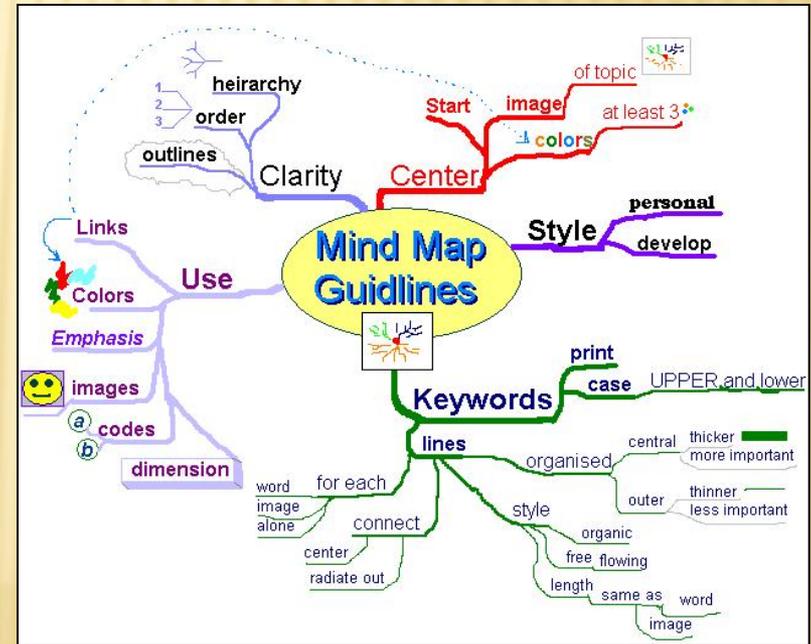
※1960年代にトニー・ブザン氏が提唱した、脳の仕組みを取り入れた「考え」や「アイデア」を思考に沿って描いていく図解表現技法

⇒メンバの印象：

- ・ **簡単**に利用でき，導入の敷居が低い
- ・ 仕様書に書いていない機能，テスト観点の**発想**がどんどん生まれる
- ・ メンバーの**意思疎通**を取りやすい



テストに使える！！



出展

<http://ja.wikipedia.org/wiki/%E3%83%9E%E3%82%A4%E3%83%B3%E3%83%89%E3%83%9E%E3%83%83%E3%83%97>

研究目標

～問題解決に向けた目標設定～

目標：マインドマップを適用したソフトウェア テストプロセスを構築する

マインドマップの利用問題を解決するために、本目標を設定

【研究内容】

- ・ **マインドマップ**で抽出した、機能とテスト観点を**整理**する方法を研究
- ・ **実業務**で利用するために、**テストプロセス**を構築
- ・ 実際のソフトウェア開発プロジェクトに適用し、以下の問題が解決できたかを検証

問題解決1) マインドマップで抽出したテスト観点でテスト設計すれば、
テストケースの**発散を抑制**できるか

問題解決2) マインドマップの抽出内容を整理した一覧表が、テスト
アーキテクチャ設計以降の入力として十分**利用できる**か

【研究前の問題】

- ・ただマインドマップを導入しただけでは、**抽出結果が発散**
⇒**膨大なテストケース**になる
- ・その結果、マインドマップの利用はテストケースの項目出し程度

【問題解決のために必要なこと】

- ①テストケースの発散を抑えるために、マインドマップの**作成目的**を機能とテスト観点ごとに**絞り**、効率よくアイデアを抽出する
- ②マインドマップで抽出した結果を、テストアーキテクチャ設計以降への入力として利用できるよう**整理**する

研究内容

～これがメイン～

- **テストプロセスの構築**

実業務で使えるプロセス定義書とPFDを作成

- **目的別マインドマップの作成**

4つのマインドマップに分け、抽出結果の余計な発散を抑制
テストケースの発散抑制につなげる

- **マインドマップの抽出結果を一覧表に整理**

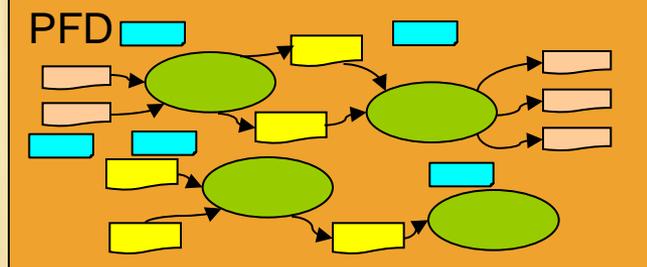
マインドマップの抽出結果を整理し、テストアーキテクチャ設計以降
の入力として利用できるようにする

各マインドマップの作成方法と要素の整理方法をテストプロセスとして構築した。

- (1) 「テスト要件定義・要求分析」プロセス
各マインドマップの作成 (OUTPUT:後頁 表1.)
一覧表にて整理 (OUTPUT:後頁 表2.)
- (2) 「テストアーキテクチャ設計」プロセス
マインドマップと一覧表を元に、テストアーキテクチャを設計
- (3) 「テスト詳細設計・実装」プロセス
テストとして実施すべき内容を洗い出す。
OUTPUT:テストケース
- (4) 「テスト実施」プロセス
テストケースを使用して、テストを実施
OUTPUT:テスト結果, 不具合情報

プロセス定義書

TSC-001 テスト要件定義
TSC-001-01
仕様書より機能を抽出する。
(目的)
・仕様書より、テスト対象物を機能ごとに整理し、機能の関連性やボリュームを俯瞰できるようにする。
(手段)
1) 設計仕様書から機能を抽出し、「機能マインドマップ」を作成する。
(注意点)
1) 重要と思う機能については、赤字でマーキングしておく。 2) 各機能の仕様不明点があれば、緑色で抽出しておく。 ※仕様不明点に対しては、仕様定義不十分/仕様矛盾を指摘する程度にとどめる(自ら仕様検討しない)。



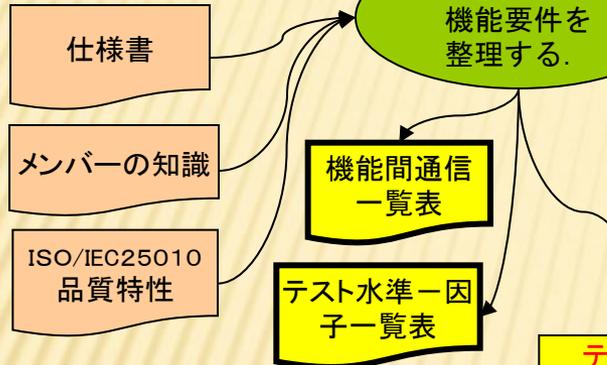
テスト担当者へのテストプロセスの理解促進のため、
プロセス定義書, プロセスフローダイアグラム (PFD) に整理した。

テスト プロセスフローダイアグラム(PFD)

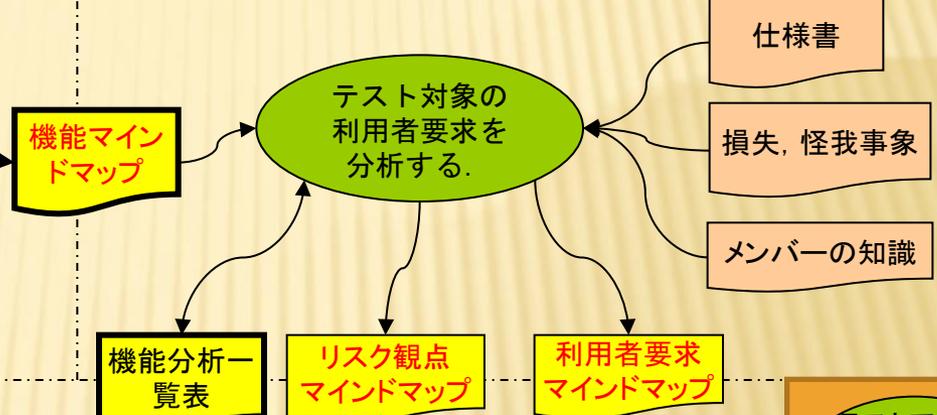


各工程でのプロセス内容と必要成果物を定義

テスト要件定義



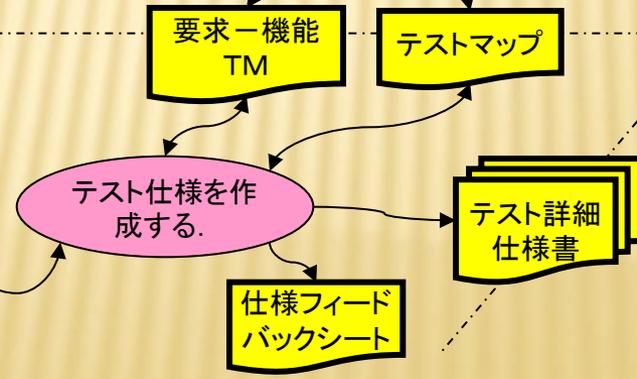
テスト要求分析



テストアーキテクチャ設計



テスト詳細検討・作成



テスト実施



テストアーキテクチャ関連プロセス

テスト詳細設計関連プロセス

中間成果物

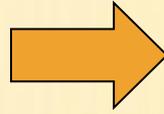
最終成果物

入力となる成果物

目的別マインドマップの作成

【From: 漠然とした視点】

余計に発散し効率よく抽出できない
有効な抽出結果が得られない

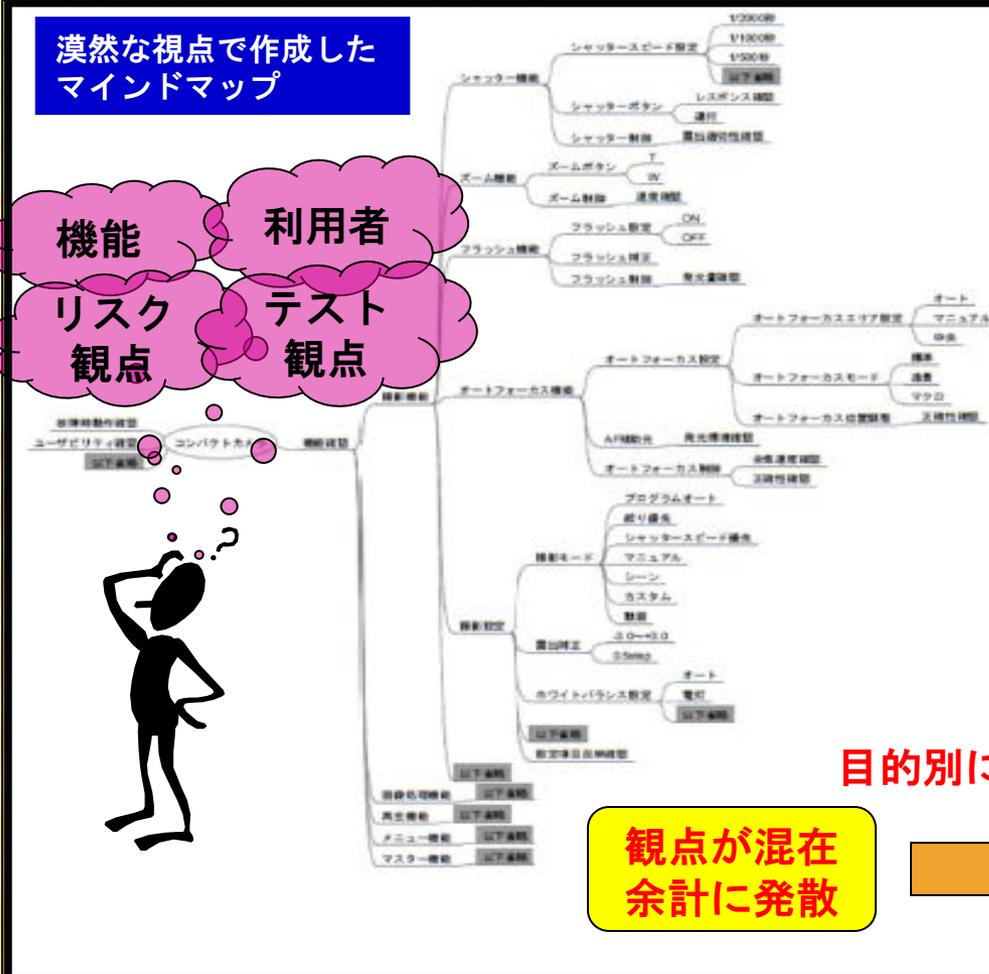


【To: 目的別に複数のマインドマップ】

4つのマインドマップに分けて作成
機能, 利用者要求, リスク観点, テスト観点

漠然な視点で作成した
マインドマップ

機能
利用者
リスク
観点
テスト
観点



機能マインドマップ



機能

利用者要求マインドマップ

利用者

リスク観点マインドマップ

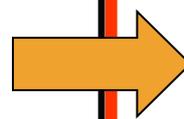
リスク
観点

テスト観点マインドマップ

テスト
観点

目的別に作成

観点が混在
余計に発散



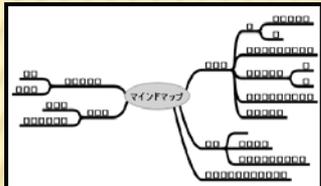
観点が独立
余計な発散抑制

マインドマップと一覧表の関係

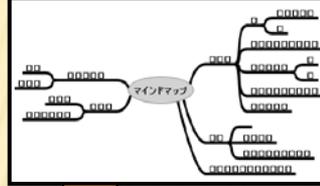
～マインドマップ抽出結果を一覧表で整理する流れ～

テストタイプ一覧表
(テスト観点とテストタイプを紐付け)

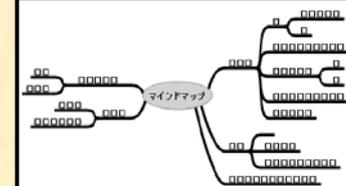
目的：利用者視点
リスク観点マインドマップ



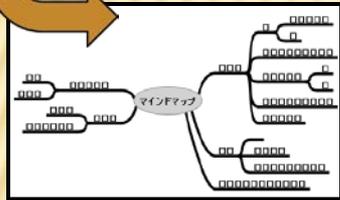
目的：機能の抽出
機能マインドマップ



目的：テスト観点
テスト観点マインドマップ



利用者要求マインドマップ



機能分析一覧表 (機能の理解と整理)

機能ID	機能分類	機能名	機能説明	テスト要求	EW2H分析
				When	Where
Func01	シャッター機能	シャッタースピード設定	シャッタースピードを設定したい	シャッターボタンを押して撮影したい	撮影時の状況や場所
Func02	ズーム機能	ズームボタン	ズームボタンを押して撮影したい	ズームボタンを押して撮影したい	撮影時の状況や場所
Func03	フラッシュ機能	フラッシュ設定	フラッシュをON/OFF設定したい	フラッシュボタンを押して撮影したい	撮影時の状況や場所
Func04	オートフォーカス機能	AF補助光	AF補助光をON/OFF設定したい	撮影ボタンを押して撮影したい	撮影時の状況や場所
...

テストタイプ	テスト観点
シナリオテスト	ユースケース確認
運用テスト	メンテナンス確認
単機能テスト	個別機能確認
機能間連携テスト	機能間連携確認
セキュリティテスト	セキュリティ確認
異常系テスト	異常系確認
非機能要件テスト	非機能要件確認
ホワイトボックス	ホワイトボックス確認

要求一覧

要求分類	要求内容	機能ユニット名				テストタイプ	テスト概要仕様書名
		撮影	画像処理	再生	メニュー		
利用者	写真を明るく撮りたい	○	○	○	○	シナリオテスト概要仕様書 (ユースケース)	
	今すぐ写真を撮りたい	○	○	○	○		
	遠くの建物をアップして撮りたい	○	○	○	○		
	撮った写真を観たい	○	○	○	○		
.....						

整理した機能一覧

テスト観点, テストタイプ

機能分類	機能名	テスト観点																			
		シナリオ	運用	単機能	機能間	セキュリティ	異常系	非機能	ホワイト	その他	ユース	メンテ	個別	連携	セキュ	異常	非機	白ボ	その他		
機能分類1	シャッタースピード設定	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能分類2	シャッターボタン	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能分類3	ズーム機能	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能分類4	フラッシュ機能	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能分類5	オートフォーカス機能	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能分類6	撮影モード	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能分類7	撮影設定	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	

要求-機能トレサビリティマトリクス

システムテスト

テストマップ (機能-テストタイプトレサビリティマトリクス)

機能テスト

表1. 目的別に作成するマインドマップ

作成目的	マインドマップ	説明
機能の抽出	(1)機能マインドマップ	機能仕様書をベースに、未記載の仕様も含めて、 機能を抽出 する
要求, 非機能要求の抽出	(2)利用者要求マインドマップ	利用者の視点として、実際にテスト対象物を利用する立場で、 必要な要求事項を抽出 する
リスク観点の抽出	(3)リスク観点マインドマップ	利用者に 多大な損害をもたらす不具合防止 の観点で、必要な要求事項を抽出する
テスト観点の抽出	(4)テスト観点マインドマップ	製品・システムの妥当性確認を実施するため、テストに必要な観点を洗い出す。 ISO/IEC25010の品質特性 を参考にする

マインドマップで抽出した要素の整理



表2. マインドマップで抽出した要素の整理に用いる一覧表

一覧表名称	整理対象	説明
機能分析一覧表	機能マインドマップ	<ul style="list-style-type: none">・機能マインドマップで抽出した要素を、6W2Hで理解し、機能を一覧表で整理・縦軸：機能一覧 横軸：6W2H
テストタイプ一覧表	テスト観点マインドマップ	<ul style="list-style-type: none">・同じ目的を持ったテスト観点をまとめる・テストタイプとテスト観点を紐づける。
要求-機能トレーサビリティマトリクス	利用者要求マインドマップ リスク観点マインドマップ	利用者要求のシナリオとそれを実現している機能の関係をトレースできるようにする。 <ul style="list-style-type: none">・縦軸：要件一覧 横軸：要求
テストマップ（機能-テストトレーサビリティマトリクス）	テスト観点マインドマップ	<ul style="list-style-type: none">・各機能とテストタイプを整理する・テスト観点マインドマップで抽出した要素を整理。後工程のテストアーキテクチャ設計で、その分類のまま利用できるようにまとめる。・縦軸：機能一覧 横軸：テスト観点，テストタイプ

研究内容の検証

～問題解決できるのかを検証～

実際のソフトウェア開発プロジェクトに、本研究で構築したテストプロセスを適用。

⇒以下の問題を解決したかを検証した。

問題解決1) マインドマップで抽出したテスト観点でテスト設計すれば、テストケースの発散を抑制できるか

問題解決2) マインドマップの抽出内容を整理した一覧表が、テストアーキテクチャ設計以降の入力として十分利用できるか

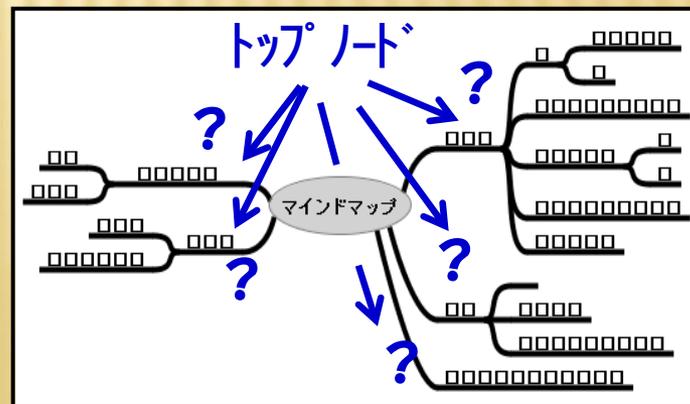
問題解決1)の検証

【結果】問題解決につながった。

- ・ 目的別に複数のマインドマップに分けて作成
⇒ 余計な発散抑制。「目的」に集中して観点出しができた。
- ・ +αの効果：
テストマップにて「他のテスト観点も当てはまるのではないか。」
という視点で横展開
⇒ 従来の開発プロジェクトで抜けていた機能とテスト観点の組み合わせを発見

【課題】

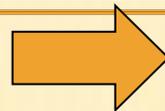
- ・ トップノードの選定方法が未確立（機能/利用者要求/リスク観点マインドマップ）
⇒ トップノードは、観点出しの入り口。
選定が誤ると、観点が十分抽出できないリスクあり。



問題解決1) 効果事例



【From: 漠然とした視点】



【To: 目的別に複数のマインドマップ】

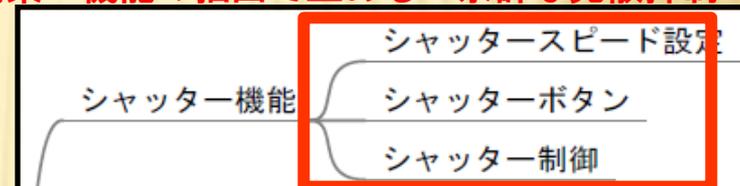
漠然な視点で作成した
マインドマップ

機能の下にテスト観点を記載
⇒マインドマップが発散！



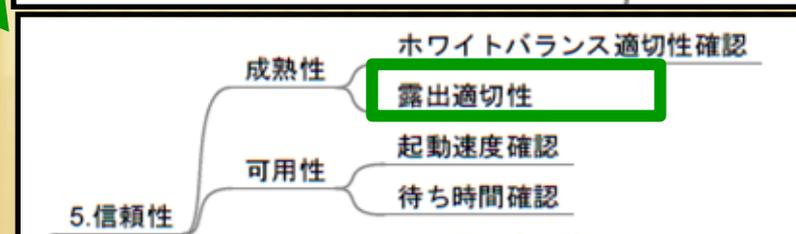
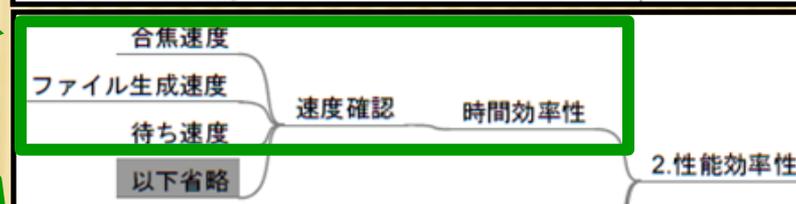
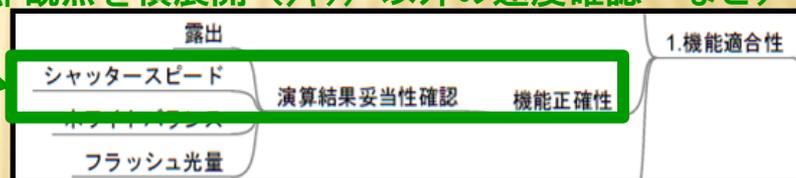
機能マインドマップ

効果：機能の抽出で止める⇒余計な発散抑制



テスト観点マインドマップ

効果：機能とは分離しテスト観点を抽出⇒他機能にも
テスト観点を横展開 (シャッター以外の速度確認 など)



問題解決2) の検証

【結果】問題解決につながった。

マインドマップの抽出内容を整理した一覧表の分類方法で、そのままアーキテクチャ設計に活用できた。

⇒分類変更作業が必要ないため、**テストアーキテクチャ設計に集中**できた。

名称	検証ポイント	検証結果 ○：解決できた △：一部解決できた ×：できなかった
機能分析一覧表	<ul style="list-style-type: none"> 各マインドマップで抽出した内容を整理できたか 後工程のテストアーキテクチャ設計で使いやすくまとめることができたか 	△ <ul style="list-style-type: none"> 整理結果をそのまま流用でき、テストアーキテクチャ設計に集中できた。 機能間のダブリや漏れがないという確認ができた。 機能マインドマップのトップノードの選定方法に悩んだ。(マインドマップのトップノード=機能分類の第1階層?)
要求-機能トレーサビリティマトリクス	利用者要求マインドマップで抽出した結果を、テストアーキテクチャ設計で利用できたか	○ <ul style="list-style-type: none"> 整理結果をそのまま流用でき、テストアーキテクチャ設計に集中できた。 利用者要求をベースとしたシナリオテストで機能間連携を網羅しているかを一目で確認可能となった。 要求・機能それぞれの抜け漏れがないかを確認できた。
テストマップ	テスト観点マインドマップで抽出した結果を、後工程のテストアーキテクチャ設計で利用できたか	○ <ul style="list-style-type: none"> 各機能に対してどのテスト観点でテストすべきかが容易に理解できた。 各機能で実施すべきテストタイプを表現でき、テスト詳細設計・実装の明確なインプットとして利用できた。

目標の達成度と，今後の展開

～目標にどれだけ達成したか，
今後の取り組みは～

目標達成！

マインドマップを適用したソフトウェアテストプロセスを構築できた。

マインドマップの作成手順や整理の方法を定義

⇒マインドマップをテスト要件定義・要求分析プロセスに利用できた。



【マインドマップの導入効果】

本プロセス適用によるテスト観点抽出効果

- ・マインドマップと一覧表の照らし合わせ

⇒ある機能のテスト観点が**他の機能にも当てはまる！**

- ・ISO/IEC25010の品質特性をトップノードに

⇒**時間効率性、障害許容性など、導入前に思いつかなかったテスト観点を発見！**



マインドマップを作成する上で、**トップノードの選定方法が未確立**
(機能/利用者要求/リスク観点マインドマップ)

⇒トップノードは**観点出しの入り口**。選定が誤ると観点抽出が不十分となる



他の開発プロジェクトでも、確実に効果を出すために



今後の展開

最適な**トップノードの選定方法**の確立

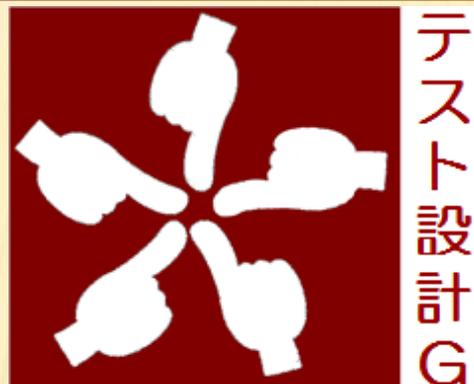
- ・機能マインドマップ：例) 機能要件の分類, 構造仕様, GUI etc
- ・利用者要求マインドマップ：例) テスト対象の利用シーンでの分類 (ユースケース)
- ・リスク観点マインドマップ：例) テスト対象で最も重要度が高い不具合

トップノードの選定

マインドマップ作成

抽出結果の考察





以上で発表を終わります。
ご清聴ありがとうございました！

