XDDPによる派生開発改善の最前線
～AFFORDDの研究成果と実開発への適用事例から見えた気付き～

派生開発推進協議会（AFFORDD）　関西部会

三菱電機（株）　白川 智也

アルファテクノロジー（株）　重松 信晶

ヤンマーエネルギーシステム（株）　加茂田 惣一郎

（株）島津製作所　山添 秀樹

本セッションの流れ

- 本日は派生開発推進協議会（AFFORDD）のご紹介、および関西部会の活動についてご紹介するとともに、**AFFORDDの研究成果**と実開発への**XDDP適用事例から見えた気付き**についてご紹介いたします。

・講演内容

（1－1）AFFORDDの活動紹介

（1－2）派生開発における影響箇所の気付き　　三菱電機（株）　白川 智也

（2）T 型マトリクスを用いたＸＤＤＰとテストプロセスの接続
～AFFORDD T4研究会活動紹介～　アルファテクノロジー（株）　重松 信晶

（3）ヤンマーエネルギーシステム（株）の適用事例紹介
ヤンマーエネルギーシステム（株）　加茂田 惣一郎

（4）（株）島津製作所の適用事例紹介　（株）島津製作所　山添 秀樹

(1－1) XDDPによる派生開発改善の最前線 ～AFFORDDの研究成果と 実開発への適用事例から見えた気付き～

AFFORDDの活動紹介

AFFORDD関西部会代表

三菱電機（株）　白川 智也

AFFORDDの活動内容

ソフトウェア開発の大部分は派生開発である

- ソフトウェアの高機能化によるソースコードの大規模化、複雑化
- 市場競争の激化に伴う短納期化

派生開発は新規開発よりも難しいことがある

- 他人の書いたプログラムを読んで理解するのが難しい
- 変更に対する影響箇所（範囲）を見極めるのが難しい
- 間に合わないといって拙速にソースコードを変更すると 手戻り作業が増えて却って時間がかかる

AFFORDDの活動内容

派生開発推進協議会(AFFORDD)の発足

- 派生開発の改善に向け、有志にて2010年に発足
  派生開発における効果的な方法論の開発と普及
  有効な打開策の情報交換と支援

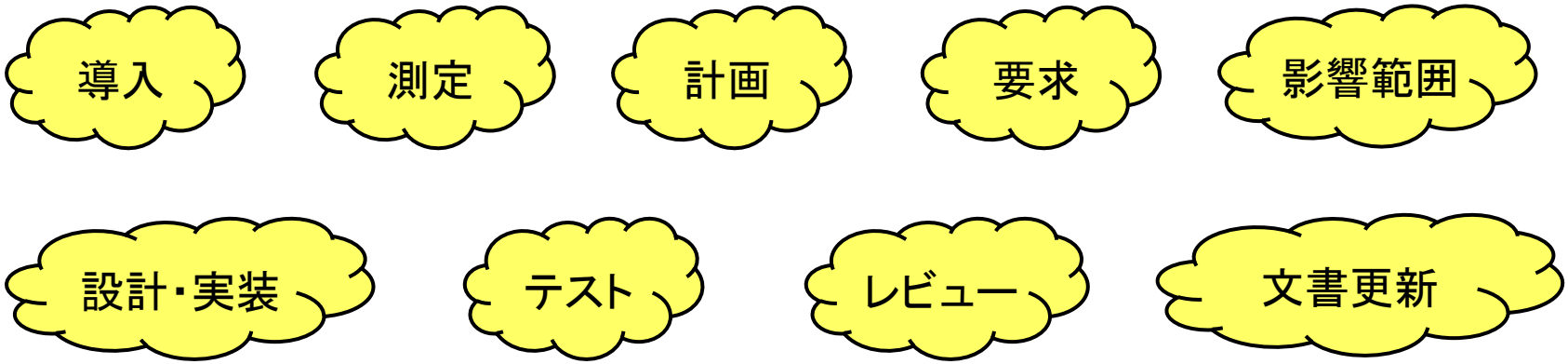
AFFORDDの活動

- 派生開発カンファレンス ～派生開発に関する研究や、事例発表の場～
- 研究会活動 ～テーマを決めて、チームで活動～
- 勉強会の開催 ～ XDDPやUSDMのミニセミナー ～

関西部会の活動内容

- 2013年9月に発足
- 親会社と関係会社が連携してソフトウェアの派生開発改善に向けた知恵を出している
- 同じ問題意識を持った仲間が知恵を出し合う

各自の課題を抽出し、9つのカテゴリに分類



⇒各自の課題を解決するヒントは「腹を割った事例紹介」にある！

⇒相互解決の場における「改善の気付き促進」

関西部会の活動内容

←白熱した楽しい議論

議論で渇いた喉を潤す会→

(1－2) XDDPによる派生開発改善の最前線
～AFFORDDの研究成果と
実開発への適用事例から見えた気付き～

派生開発における影響箇所の気付き

T5研究会員

三菱電機（株）　白川 智也

まずは、
簡単にXDDPについて
おさらいします。

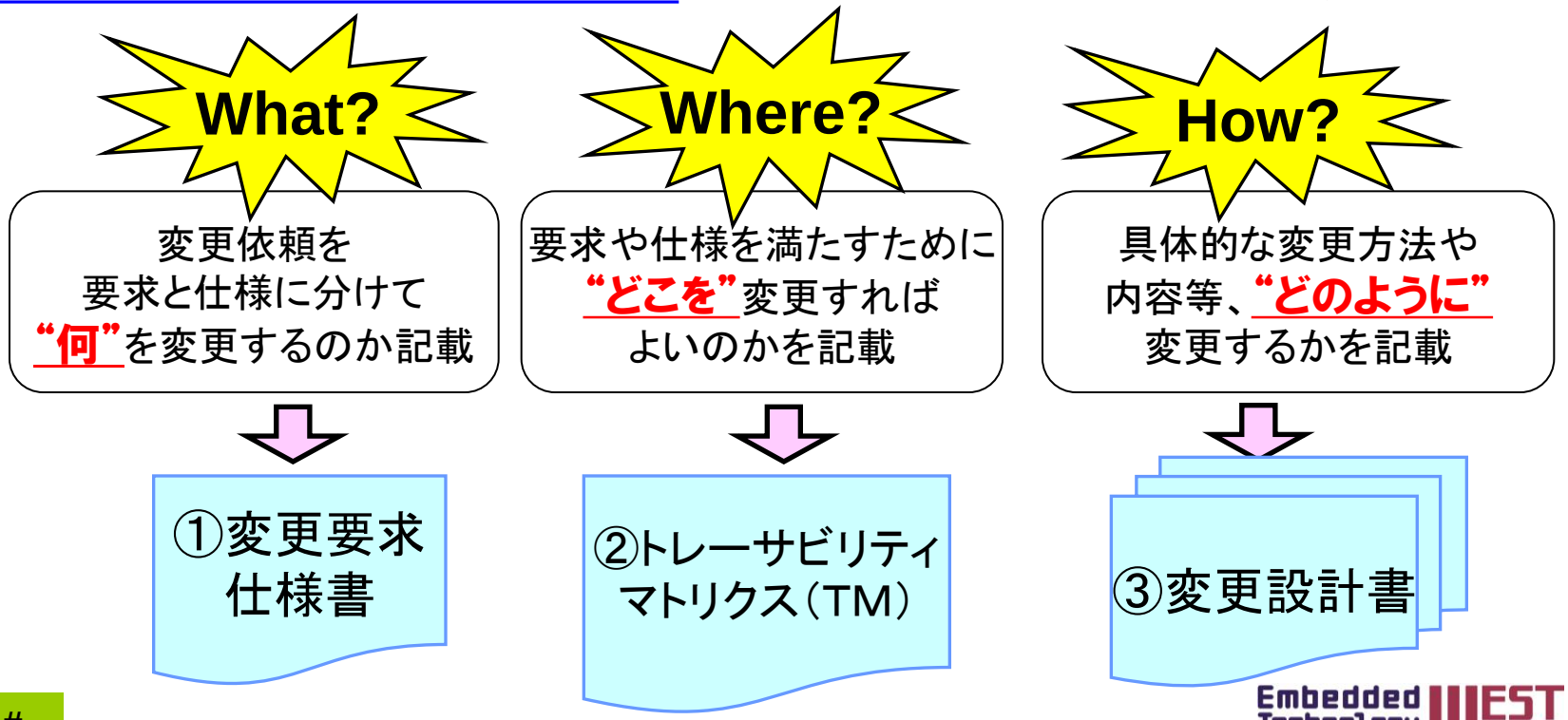
XDDPとは？

(1)変更点に着目した開発プロセス

(2) “部分理解”を前提とし、他者が担当者の“思い込み・勘違い”に気付きやすい適切な成果物を規定

⇒ 何を・どこを・どのように変更するかをドキュメント化（3点セット）

(3)3点セットのレビューに注力することで “思い込み・勘違い” の検出



XDDPとは？

①変更要求仕様書

(1)要求と仕様を階層化(*1)

(2)要求と仕様の
トレーサビリティを明示

(3)要求の理由を記述
（適切な変更を引き出す）

(4)before/afterによる
変更の表現

・～を〇〇に**変更する**
・～を〇〇に**追加する**
・～を**削除する**

要求	AAA-010			
	理由			
	説明			
	要求	01		
		理由		
		説明		
		仕様	01	
			理由	
			説明	
		仕様	02	
			理由	
			説明	
	要求	02		
		理由		
		説明		
		仕様	01	
			理由	
			説明	

*1)USDM:
Universal Specification Describing Manner

XDDPとは？

②トレーサビリティ・マトリクス

変更要求仕様書に対して、どのソースに変更が発生するかを分析する

変更要求仕様書

各ソースの関数単位での追加、変更、削除の分析結果

					source1		source2		source3		source4	
要求	RXR-010	○○○において受信レートとして64kbps,128kbpsを設定										
	理由	○○○に対応するため										
	説明											
	要求	01	○○○において受信レートとして64kbps,128kbpsを設定、保存できるようにする									
		理由	○○○に対応するため									
		説明										
		仕様	001	○○○において受信レートとして64kbps,128kbpsを設定できるようにする			関数3 関数4	削除 変更			関数3 関数4	削除 変更
			理由	○○○において受信レートをユーザが設定するため								
		仕様	002	○○○においてユーザが設定した受信レートを保存できるようにする	関数2	追加			関数1	変更		
			理由	設定した受信レートをFWに反映するため								
	要求	02	○○○において設定ファイルに保存された受信レートを読み込み、表示できるようにする									
		理由	○○○に対応するため									
		説明										
		仕様	001	○○○において保存されている受信レートを読み込みできるようにする								
			理由	保存された設定を表示するため								
		仕様	002	○○○において保存されている受信レートを表示できるようにする	関数1	追加						
			理由	保存された設定を表示するため								
要求	RXR-020	○○○における受信レートから自動を削除する										
	理由	受信レート仕様変更のため										
	説明	–										
	要求	01	○○○において受信レートとして自動を設定、保存できないようにする									

XDDPとは？

③変更設計書

「どのソースファイルのどの関数をどのように変更するか？」を記載する

プロジェクト名		作成日			
ソース名/タスク名		作成者			
		修正者			
変更要求仕様		確認者			
		見積り行数	15	見積り時間	0.5
		変更行数	15	作業時間	0.5
				変更設計書作成時間	1

・関数の変更

ファイル名	XXXX.c	
関数名	void XXXXXX( )	■変更、□追加、□削除

変更内容:

項目#	変更内容	変更行数	追加行数	削除行数
1	現在の操作モードを示すXXXXの値がの場合、XXXタを初期化する関数XXXX()をコールしないようにする。	0	0	1

確認項目:

項目#	確認内容	チェック
1	<<変更の内容を確認する方法を記載>>	□

変更行数が見積もれるレベルまで設計する

作業時間などの見積もりと実績を記載する。

変更を試験で確認する方法を記載する。

派生開発における影響箇所の気付き

- AFFORDDのT5研究会において「派生開発における影響箇所の気付き」を研究
- 発足の目的

  派生開発の難しいところは、変更に対して仕様上で関連する箇所と、
  予想外のところで影響する箇所があることである。
  特に変更に関連して影響する箇所の見落としが大きな問題になる。
  もっと効果的に影響範囲、影響箇所を気づく方法は？

影響箇所の気付きに関する課題

（1）**大規模、複雑化したソフトウェア**に対する変更において、
影響箇所を見落とすことが多い
⇒**影響箇所の見通しをよくしたい**

（2）**影響箇所の調査（スペックアウト）が属人的**に実施されている
⇒**影響箇所の調査の属人性を低減したい**

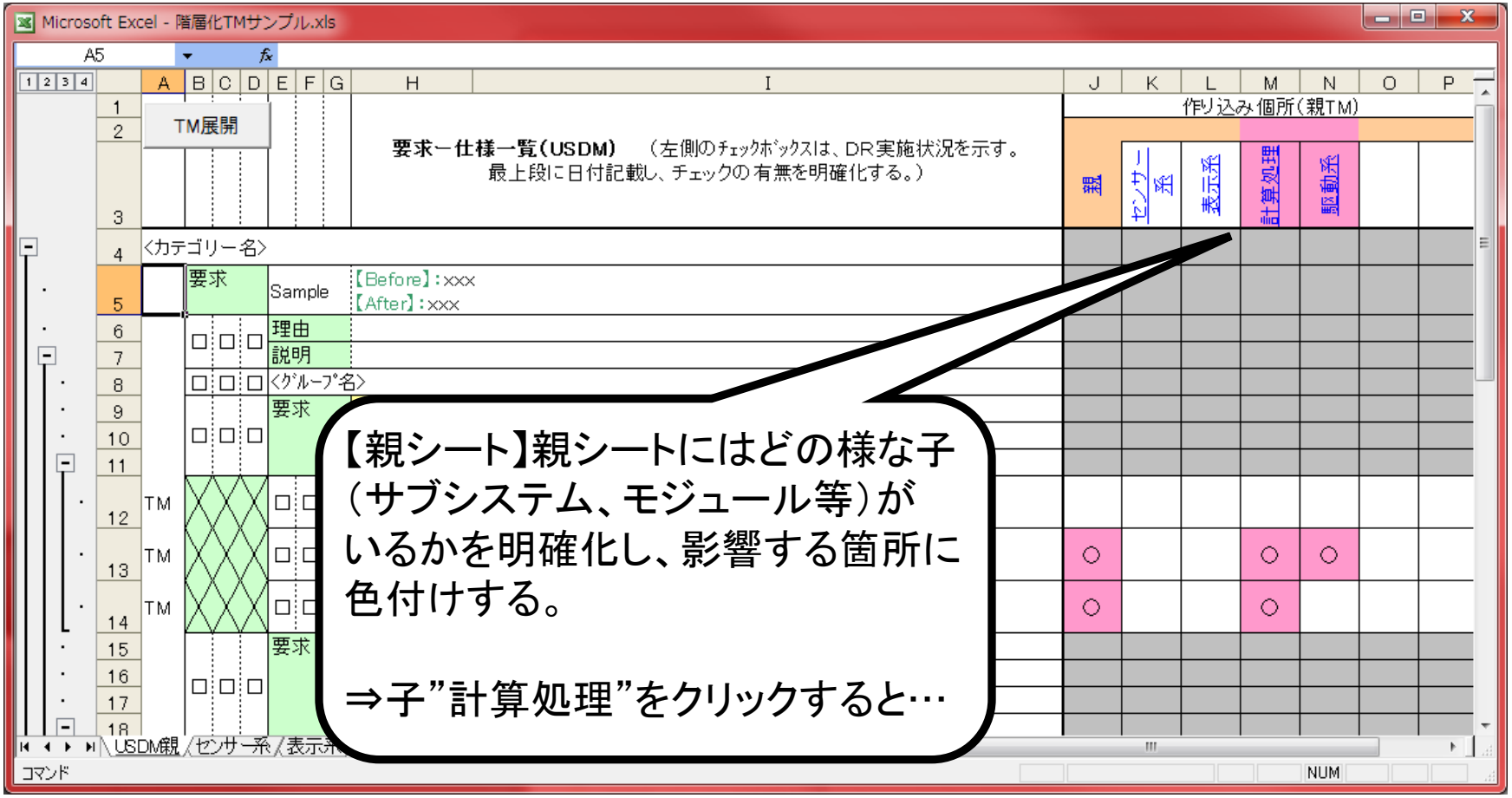
派生開発における影響箇所の気付き

課題に対する活動の成果

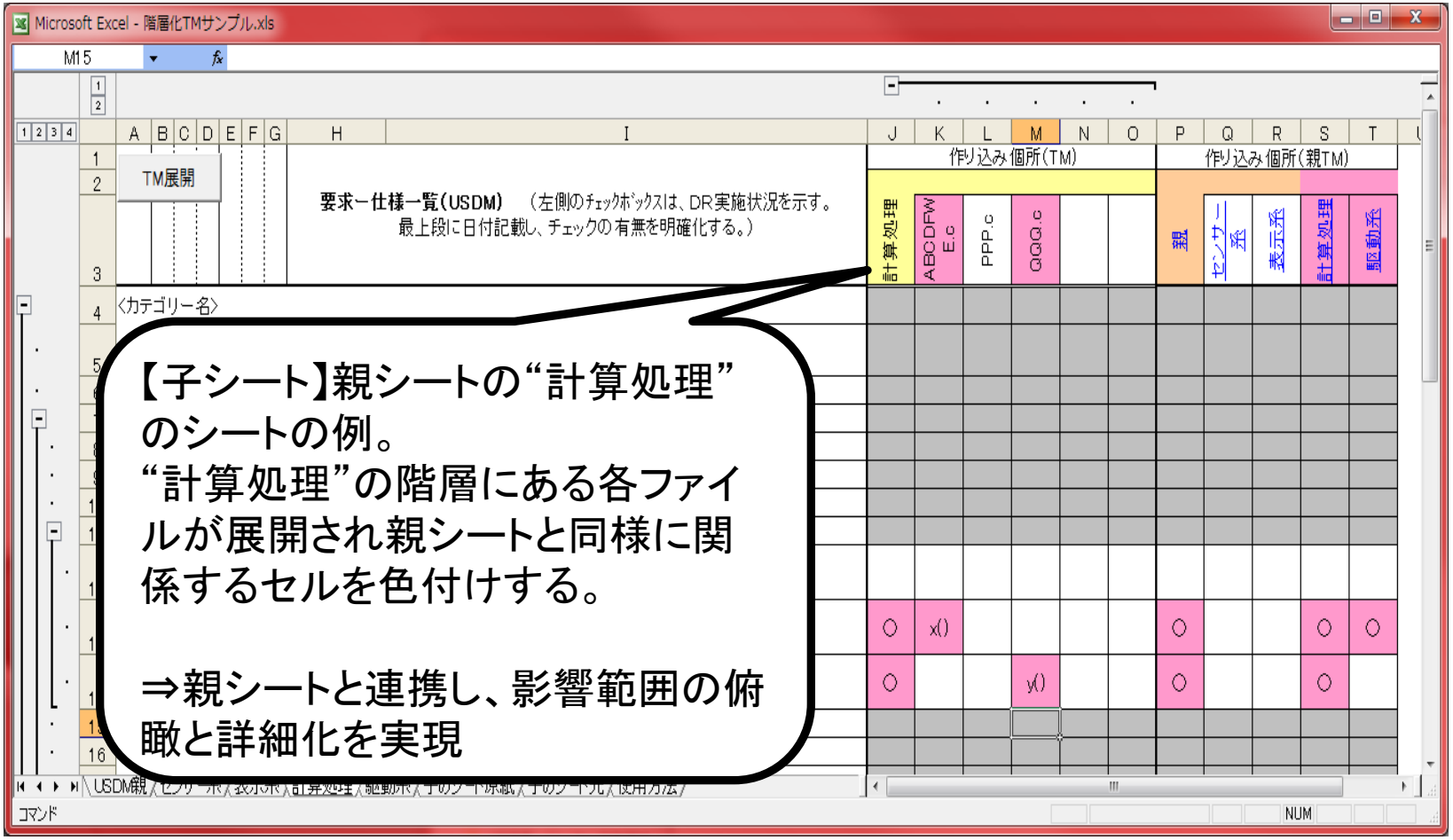
（1）**階層化したトレーサビリティマトリクス**を
　　活用した影響箇所の表現

（2）**変更パターン毎に影響箇所の調査方法**を定義

（１）階層化したトレーサビリティ
マトリクスを活用した影響箇所の表現



（１）階層化したトレーサビリティ
マトリクスを活用した影響箇所の表現



（２）変更パターン毎に影響箇所の調査方法を定義

USDM：仕様変更パターンのマトリックス

			変更パターン			
			A	B	C	D
要求1						
	要求1．1					
		仕様1.1.1		○		
		仕様1.1.2	○			
	要求1．2					
		仕様1.2.1			○	

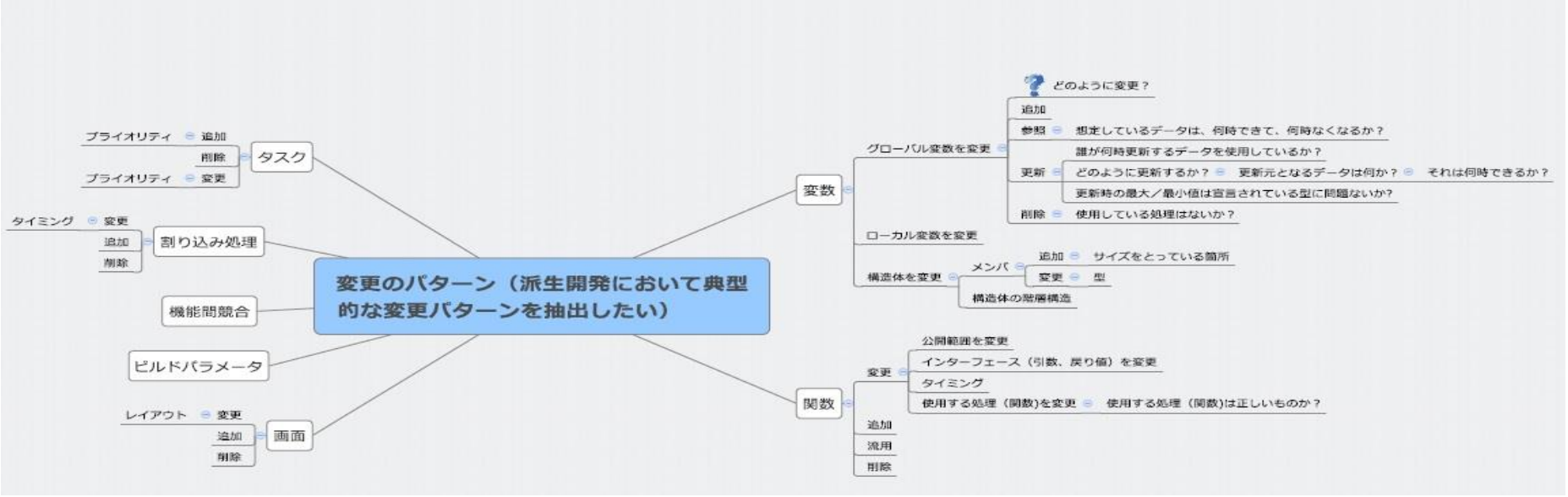
1つの仕様に対して変更パターンが決まってくる！！

変更パターン毎の調査方法（SpecOut作成ガイド）

変更パターン	調査方法					
	DFD	STD	シーケンス図	コラボレーション図	・	・
A	○					
B			○			
C		○				
D				○		

（２）変更パターン毎に影響箇所の調査方法を定義

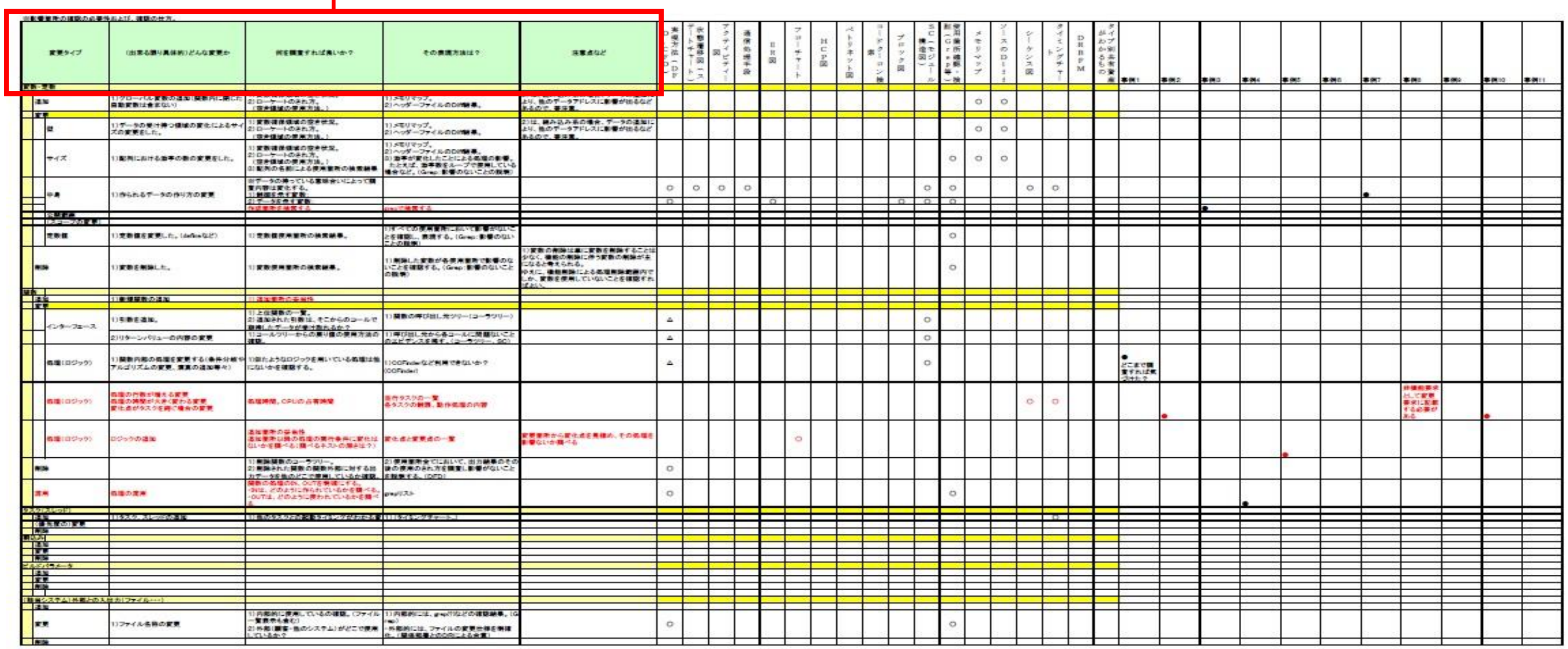
派生開発において想定される変更タイプを抽出



（２）変更パターン毎に影響箇所の調査方法を定義

影響範囲の気づきを得る為の調査対象・範囲の表現

- ・変更タイプ
- ・変更内容
- ・調査方法
- ・調査結果の表現手段
- ・注意点



（２）変更パターン毎に影響箇所の調査方法を定義

変更タイプ			（出来る限り具体的）どんな変更か	何を調査すれば良いか？	その表現方法は？
変数・定数					
	追加		1）グローバル変数の追加（関数内に閉じた自動変数は含まない）	1）変数確保領域の空き状況。 2）ローケートのされ方。 （空き領域の使用方法。）	1）メモリマップ。 2）ヘッダーファイルのDiff結果。
	変更				
		型	1）データの受け持つ領域の変化によるサイズの変更をした。	1）変数確保領域の空き状況。 2）ローケートのされ方。 （空き領域の使用方法。）	1）メモリマップ。 2）ヘッダーファイルのDiff結果。
		サイズ	1）配列における添字の数の変更をした。	1）変数確保領域の空き状況。 2）ローケートのされ方。 （空き領域の使用方法。） 3）配列の名前による使用箇所の検索結果	1）メモリマップ。 2）ヘッダーファイルのDiff結果。 3）添字が変化したことによる処理の影響。 たとえば、添字数をループで使用している場合など。（Grep：影響のないことの説明）
		中身	1）作られるデータの作り方の変更	※データの持っている意味合いによって調査内容は変化する。 1）制御を示す変数： 2）データを示す変数： 作成箇所を検索する	 grepで検索する
		公開範囲（スコープの変更）			
		定数値	1）定数値を変更した。（defineなど）	1）定数値使用箇所の検索結果。	1)すべての使用箇所において影響がないことを確認し、表現する。（Grep：影響のないことの説明）
	削除		1）変数を削除した。	1）変数使用箇所の検索結果。	1）削除した変数が各使用箇所で影響のないことを確認する。（Grep：影響のないことの説明）

（２）変更パターン毎に影響箇所の調査方法を定義

その表現方法は？	注意点など	実現方法（DFD，CFD）	状態遷移図（ステートチャート）	アクティビティー図	通信処理手段	ER図	フローチャート	HCP図	ペトリネット図	コードクローン検索	ブロック図	SC（モジュール構造図）	使用箇所確認・検証（Grep等）	メモリマップ	ソースのDiff	シーケンス図	タイミングチャート
1）メモリマップ。 2）ヘッダーファイルのDiff結果。	2）は、組み込み系の場合、データの追加により、他のデータアドレスに影響が出るなどあるので、要注意。													○	○		
1）メモリマップ。 2）ヘッダーファイルのDiff結果。	2）は、組み込み系の場合、データの追加により、他のデータアドレスに影響が出るなどあるので、要注意。													○	○		
1）メモリマップ。 2）ヘッダーファイルのDiff結果。 3）添字が変化したことによる処理の影響。 　たとえば、添字数をループで使用している場合など。（Grep：影響のないことの説明）													○	○	○		
		○	○	○	○							○	○			○	○
		○				○					○	○	○				
grepで検索する																	
1)すべての使用箇所において影響がないことを確認し、表現する。（Grep：影響のないことの説明）													○				
1）削除した変数が各使用箇所で影響のないことを確認する。（Grep：影響のないことの説明）	1）変数の削除は単に変数を削除することは少なく、機能の削除に伴う変数の削除が主になると考えられる。 ゆえに、機能削除による処理削除範囲内でしか、変数を使用していないことを確認すればよい。												○				

今後の課題

1. 実開発への適用を通じた**効果確認**

2. 変更パターンの**テーラリング**
   ・現時点における変更パターンは、研究会メンバの
   経験・事例に基づいて構築したもの（後日、公開予定）

   →変更パターンの枠組みはできたが
   開発現場の特性に合わせた最適化が必要