DENSO
Crafting the Core

CONFIDENTIAL
関係者外秘

プロセステーラリングへの プロダクトライン適用の実践

株式会社デンソークリエイト
池永直樹

アジェンダ

1. はじめに
2. 背景
3. 現状分析
4. 課題提起
5. 解決策の提案
6. 解決策の実践
7. 解決策の評価
8. まとめ と 今後の取り組み

はじめに

■ これまでのプロセス改善活動

- プロジェクト目標達成のために、組織標準プロセスをプロジェクト特性にフィットした形でテーラリングし適用することが必要条件と考える。
- 組織標準プロセスからのテーラリングの正確性・効率性向上のために、**代表的なテーラリングパターンを定義**するプロセス改善活動を行ってきた。

■ 改善点

- 代表的なテーラリングパターンを用意したとしても、さらなる小修整が必要である。
- この場合でも、プロセステーラリングが適正かつ効率良く実施されるようにしたい。

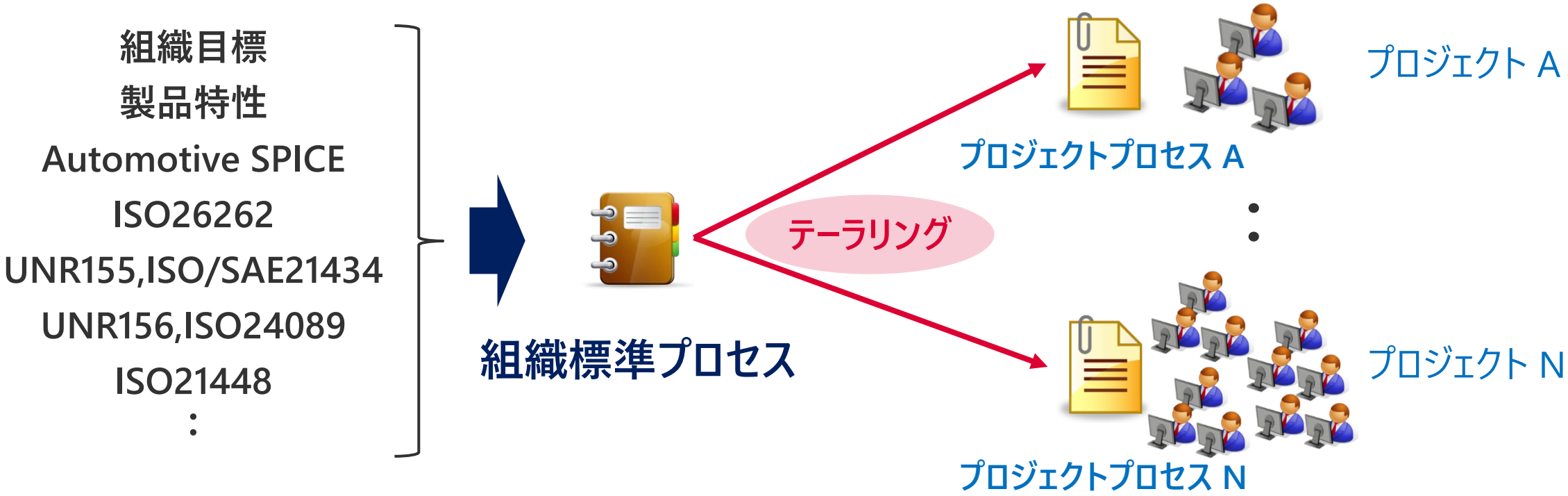
■ 課題と解決アプローチ

- 現状を分析した結果、**スキル面と仕組み面に課題**があると認識した。
- まず**プロセステーラリングの仕組み**をうまく構築することに取り組んだ。
- 課題解決のアプローチとして、"**プロセステーラリングへのソフトウェアプロダクトラインエンジニアリング（SPLE）の適用**" を実践した。

背景：プロセスの多様化

CONFIDENTIAL
関係者外秘

- 車載システムでは、数多くのバリエーション展開開発に加え、製品特性やOEM要求に応じた様々な業界標準・規格（Automotive SPICE、ISO26262、ISO21434、など）への対応が求められている。
- 従って、プロジェクトで実装されるプロセスも多様である。



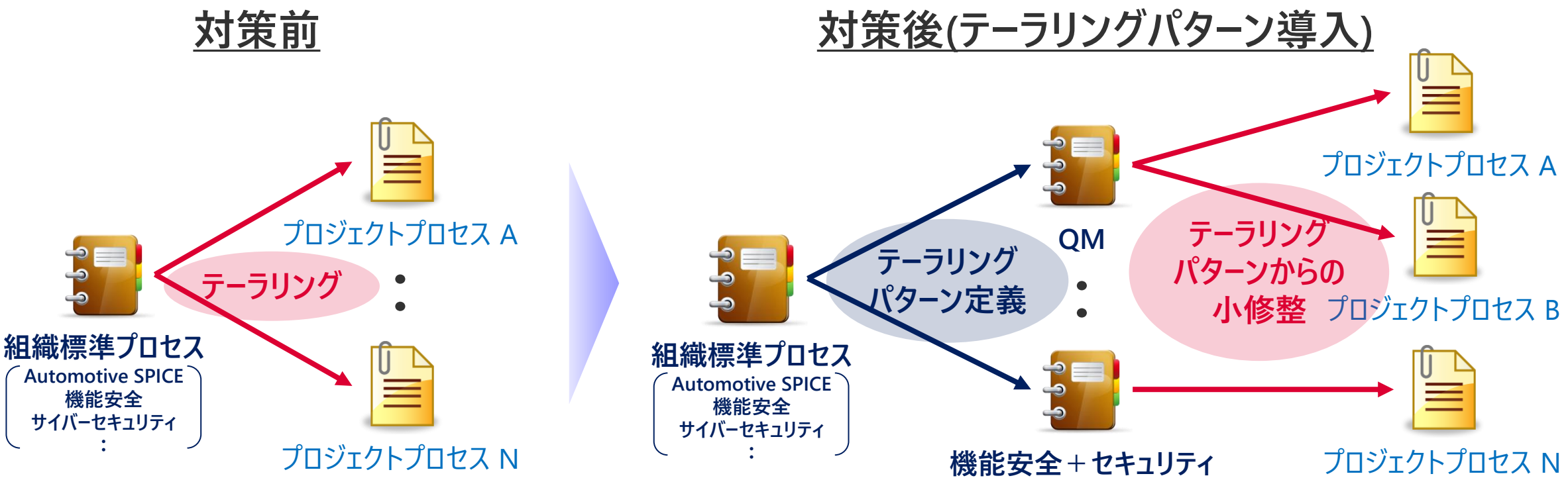
プロセステーラリングの重要性がより高まってきている

Automotive SPICE：車載システム開発向けのプロセスモデル
ISO26262：自動車-機能安全
ISO21448：自動車-意図した機能の安全性

ISO/SAE 21434：自動車-サイバーセキュリティ
ISO 24089：自動車-ソフトウェアアップデート

背景：テーラリングパターンの導入

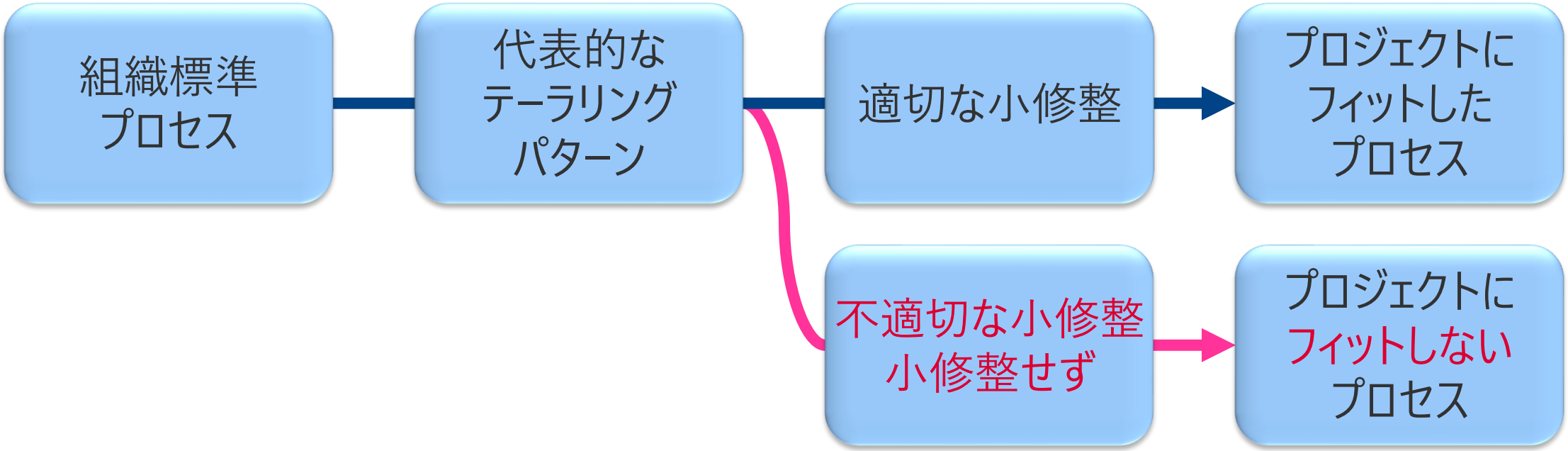
- プロジェクト目標達成のためには、組織標準プロセスを**プロジェクト特性にフィットした形でテーラリング**し適用することが必要条件である。
- フル定義された組織標準プロセスからの**テーラリングの正確性・効率性向上**のために、**代表的な（規格を軸にした）テーラリングパターンを定義**し、テーラリング実施を支援するようにした[3]。
  - ✓ テーラリングパターンを用いたプロセステーラリングのアプローチは他でも提唱されている[5][6]。



*QM：Quality Management

背景：改善が必要な点

- 代表的なテーラリングパターンを用意したとしても、**さらなる小修整が必要**なプロジェクトがある。
- この場合でも、プロセステーラリングが適正かつ効率良く実施されるようにしたい。



プロジェクトにフィットした形で組織標準プロセスの
テーラリングが実施されるようにさらに改善したい

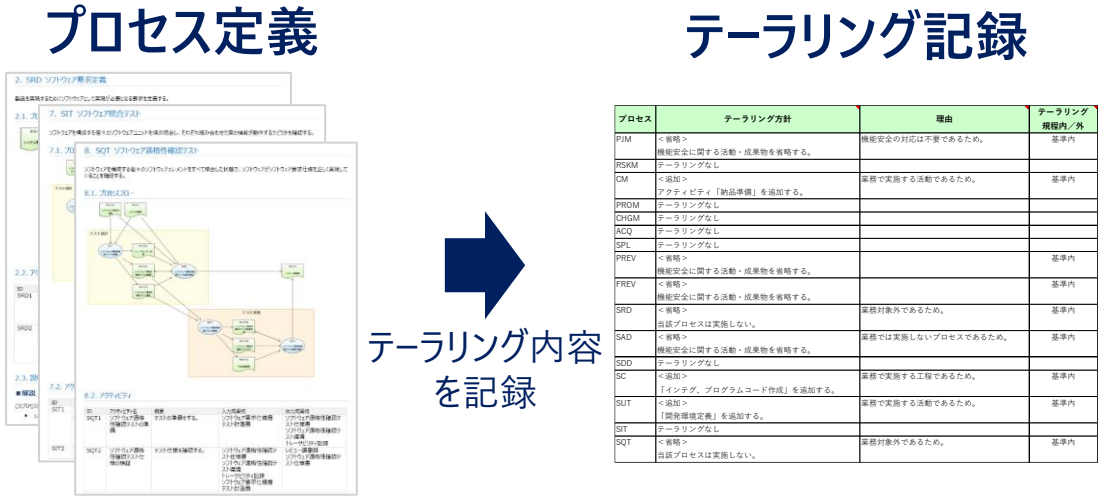
現状分析

■ テーラリング実施者の力量

- どのテーラリング観点が、プロセス定義のどこに関係するのか特定できない／気が付かない。
- 特定できても、どう修整していいのか分からない。

■ テーラリング時の手間

- テーラリング結果の記録が面倒である。
- 数あるプロセス定義情報の修整要否を確認することに時間が掛かる。
- プロセスの補正が必要な場合がある。
  - ✓ 例えば、活動を省略した場合、その前後の活動を適切につなぐ必要がある



プロセス	テーラリング方針	理由	テーラリング規程内／外
PJM	＜省略＞ 機能安全に関する活動・成果物を省略する。	機能安全の対応は不要であるため。	基準内
RSKM	テーラリングなし		
CM	＜追加＞ アクティビティ「納品準備」を追加する。	業務で実施する活動であるため。	基準内
PROM	テーラリングなし		
CHGM	テーラリングなし		
ACQ	テーラリングなし		
SPL	テーラリングなし		
PREV	＜省略＞ 機能安全に関する活動・成果物を省略する。		基準内
FREV	＜省略＞ 機能安全に関する活動・成果物を省略する。		基準内
SRD	＜省略＞ 当該プロセスは実施しない。	業務対象外であるため。	基準内
SAD	＜省略＞ 機能安全に関する活動・成果物を省略する。	業務では実施しないプロセスであるため。	基準内
SDD	テーラリングなし		
SC	＜追加＞ 「インテグ、プログラムコード作成」を追加する。	業務で実施する工程であるため。	基準内
SUT	＜追加＞ 「開発環境定義」を追加する。	業務で実施する活動であるため。	基準内
SIT	テーラリングなし		
SQT	＜省略＞ 当該プロセスは実施しない。	業務対象外であるため。	基準内

■ テーラリングパターン定義は代表のみ

- SEPGの労力や選択時の使用性を考えると、テーラリングパターンは代表パターンに留めるのが妥当。

課題提起

改善点

テーラリングパターンからの小修整も含めて、
プロセステーラリングをより適正かつ効率良く実施されるようにしたい

現状

- テーラリング実施者の力量が不足している場合がある　⇒　スキル
- テーラリングに手間が掛かってしまう　⇒　仕組み

課題

- テーラリング実施者（プロジェクトマネージャ）のテーラリングスキルの向上
- **より効率的で適正なテーラリングの実施を支援する仕組みの確立**

発表の対象

まずはプロセステーラリングの仕組み面の課題解決に取り組む

解決策の提案：解決策に対する要件

■「より効率的で適正なプロセステーラリングの実施を支援する仕組み」に対する要件

1. **標準的なテーラリング観点（＝プロジェクト特性）をカバーできること**
   標準プロセスの適用範囲が制限されないようにするため、テーラリング観点に制約が出ないようにする。

2. **用意できるテーラリングパターンに制約がでないこと**
   もともと用意できていた代表的なテーラリングパターンの提供を継続する。

3. **省略以外の修整ケースも実現できること**
   "省略" だけでなく、"簡略化・追加・統合・代替"[4] の修整ケースにも対応する。

4. **プロジェクトプロセスの完全性を確保できること**
   テーラリング時のプロセス補正の手間の削減や放置を防止するために、テーラリング基準内でテーラリングされたプロセスは完全な状態にしておく。

5. **テーラリング結果の記録が手軽であること**
   従来のようにテーラリング内容を一つ一つ記録しなくてよいようにする。

解決策の提案：課題解決アプローチ

着想

- ✓ プロセステーラリングは、**ソフトウェアプロダクトラインエンジニアリング（SPLE）**の考え方と共通性がある[1]。
- ✓ すなわち、

テーラリング観点（プロジェクト特性） を **フィーチャ**

組織標準プロセスのプロセス定義情報＊ を **コア資産**

＊プロセス定義情報：アクティビティ／タスク、成果物、ロール、利用するガイドライン・テンプレート・チェックリスト、など

と捉えることで、ソフトウェアプロダクトラインエンジニアリングの適用が可能である。

課題解決アプローチ

組織標準プロセスのテーラリングへのソフトウェアプロダクトラインエンジニアリング（SPLE）の適用

解決策の提案：前提条件

CONFIDENTIAL
関係者外秘

■ プロセステーラリングへの SPLE の適用を成功させるための前提条件

1. **プロセス定義情報が、テーラリングで調整したい粒度で分別できること**

   プロセス定義情報（活動、成果物、ロール、品質基準など）が明確に定義されていないと、
   テーラリングで調整したい対象を正確に特定できない。
   また、粒度がバラつくことでテーラリングされたプロセスの質が低下してしまう。

2. **テーラリング観点（フィーチャ）とプロセス定義情報（コア資産）間の関係が正確に定義できること**

   テーラリング観点と調整対象のプロセス定義情報との関係が定義されていないと、
   テーラリングされたプロセスが構築できない。
   調整したい粒度でテーラリング観点とプロセス定義情報間の関係を定義できる必要がある。

**プロセス定義のデータ構造を定義した上で組織標準プロセスを記述し、
プロセステーラリングへの SPLE を適用する**

解決策の実践：プロセス記述ツールと実施ステップ

CONFIDENTIAL
関係者外秘

組織標準プロセスを記述するツールとして、設計ツールである Next Design[13] を使用する。

組織標準プロセス定義

		≪対応するSPLE活動≫
Step1	**テーラリング観点とテーラリングパターンの定義** テーラリングの調整観点となるプロジェクト特性の関係・構造を整理し、フィーチャモデルを作成する。代表的なテーラリングパターンを定義する。	可変性分析 コンフィグレーション
Step2	**プロセス記述のデータ構造定義** メタモデルを使い、プロセス記述のデータ構造を定義する。テーラリングで調整したい構成要素を考慮して定義する。	ドメインアーキテクチャ設計
Step3	**組織標準プロセスの記述** メタモデルに基づいて、全てのテーラリング観点（フィーチャ）で必要となるプロセスを定義し、文書化する。	ドメインアーキテクチャ設計 コア資産開発
Step4	**プロセス定義情報へのフィーチャ割り当て** テーラリング観点（フィーチャ）と関係するプロセス定義情報を特定し、フィーチャを割り当てる。	ドメインアーキテクチャ設計 コア資産開発
Step5	**プロジェクトでのプロセステーラリング** プロジェクトに適用するテーラリングパターンを選択し、さらに小修整する。	コンフィグレーション

解決策の実践：【Step1】テーラリング観点とテーラリングパターンの定義（１）

CONFIDENTIAL
関係者外秘

■ テーラリングで考慮すべきプロジェクト特性

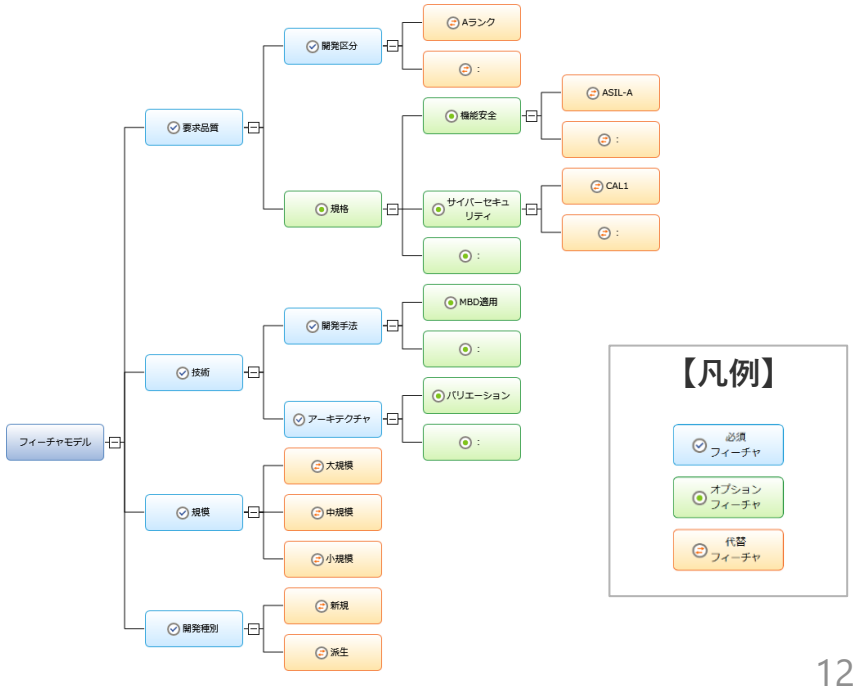
- プロジェクト特性のうち、開発プロセスに影響を与えるもの。テーラリングの観点となる。
  - ✓ 例[4] ：要求品質、規模、開発期間、開発種別（新規・派生）、技術的複雑度

■ フィーチャモデル化

- テーラリング観点（プロジェクト特性）の関係・構造を整理し、フィーチャモデルを作成する。

テーラリング観点	内容
開発区分	Aランク、Bランク、Cランク、・・・
機能安全	ASIL-A、ASIL-B、ASIL-C、ASIL-D
サイバーセキュリティ	CAL1、CAL2、CAL3、CAL4
開発手法	モデルベース開発、・・・
バリエーション	バリエーション有／無
規模	大規模製品、中規模製品、小規模製品
開発種別	新規、派生
：	

テーラリング観点のフィーチャモデル



解決策の実践：【Step1】テーラリング観点とテーラリングパターンの定義（２）

CONFIDENTIAL
関係者外秘

■ テーラリングパターンの定義

- 組織標準プロセスとしてあらかじめ用意しておくとよい代表的なテーラリングパターンを定義する。
- **テーラリングパターン＝プロダクト**と見立て、テーラリングパターンをコンフィグレーションする。

テーラリング観点（フィーチャ）

名前	QM	機能安全	CS	機能安全＋CS
フィーチャモデル				
要求品質	☑	☑	☑	☑
開発区分	☑	☑	☑	☑
Aランク	☑	☑	☑	☑
：	☐	☐	☐	☐
規格	☐	☑	☑	☑
機能安全	☐	☑	☐	☑
ASIL-A	☐	☑	☐	☑
：	☐	☐	☐	☐
サイバーセキュリティ	☐	☐	☑	☑
CAL1	☐	☐	☑	☑
：	☐	☐	☐	☐
技術	☑	☑	☑	☑
開発手法	☑	☑	☑	☑
MBD適用	☐	☐	☐	☐
：	☐	☐	☐	☐
アーキテクチャ	☑	☑	☑	☑
バリエーション	☑	☑	☑	☑
：	☐	☐	☐	☐

テーラリングパターン（プロダクト）

テーラリングパターンごとに有効なテーラリング観点（プロジェクト特性）を選択

解決策の実践：【Step2】プロセス記述のデータ構造定義（１）

CONFIDENTIAL
関係者外秘

■ プロセス記述

- プロセス記述には**データ構造（構成要素、関係）**がある。
- プロセス定義では、構成要素の内容、および、それらの関係を定義している。
- プロセスの構成要素、および、それらの関係性を明確に定義する。

構成要素	✓ プロセス・アクティビティ・タスク、成果物、ロール、品質基準、など
関係	✓ 入出力成果物、プロセス間の順序、プロセスに対する責任分担、など

■ プロセス記述のデータ構造を定義するには

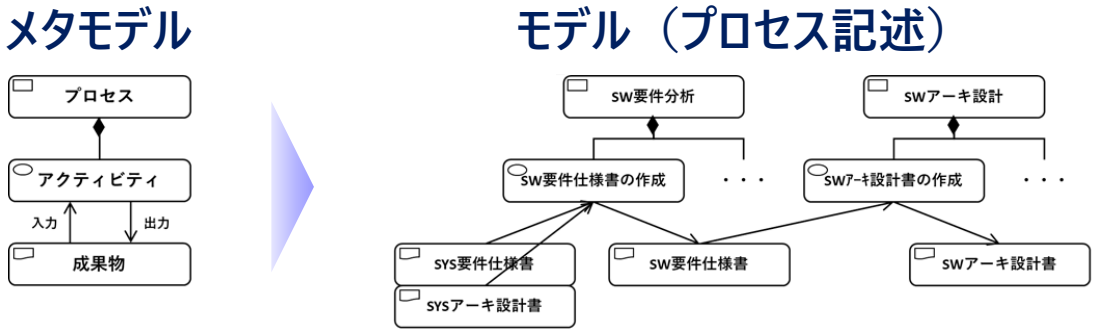
- **メタモデル**を使い、プロセス記述のデータ構造（構成要素、関係）を定義する。
- メタモデルは、個別事情に対する拡張性を備えており、**各組織にフィット**させることが期待できる[7]。
- メタモデルは、既存のプロセス記述ツールである EPF Composer[9]（SPEM2.0[10]）や Method Park 社の Stages[11] でも用いられている。

解決策の実践：【Step2】プロセス記述のデータ構造定義（２）

CONFIDENTIAL
関係者外秘

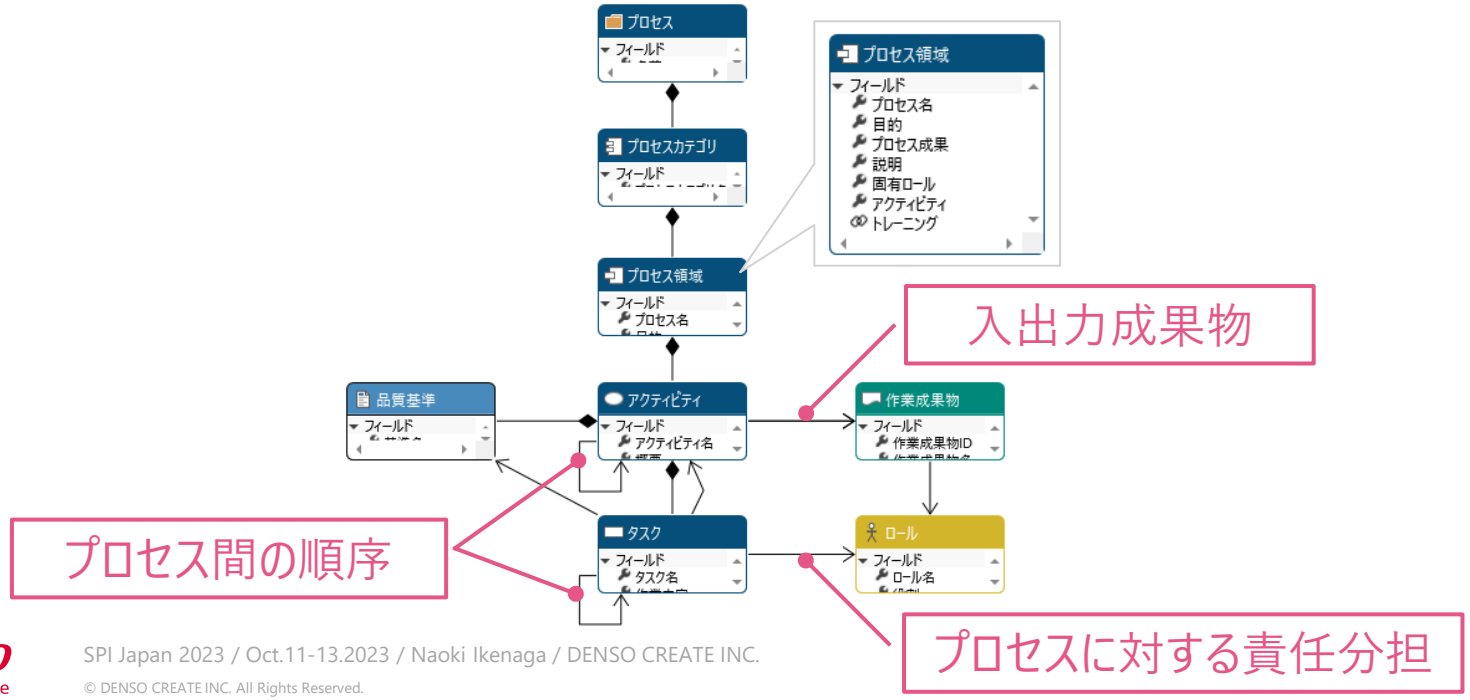
■ メタモデルとは

- モデルを記述するモデル。
  モデルを構成する概念、規則を示す[8]。



■ データ構造のメタモデリング

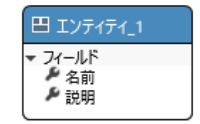
- メタモデル言語を使い、組織標準プロセスのメタモデルを定義する。
- フィーチャを構成要素に割り当てるので、**テーラリングで調整したい構成要素を考慮**して定義する。



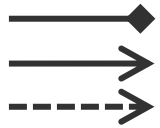
メタモデル言語

組織標準プロセスのメタモデルを表現するための言語
（MOF[12] のM2層に相当）

構成要素：



所有：
関係：
トレース：



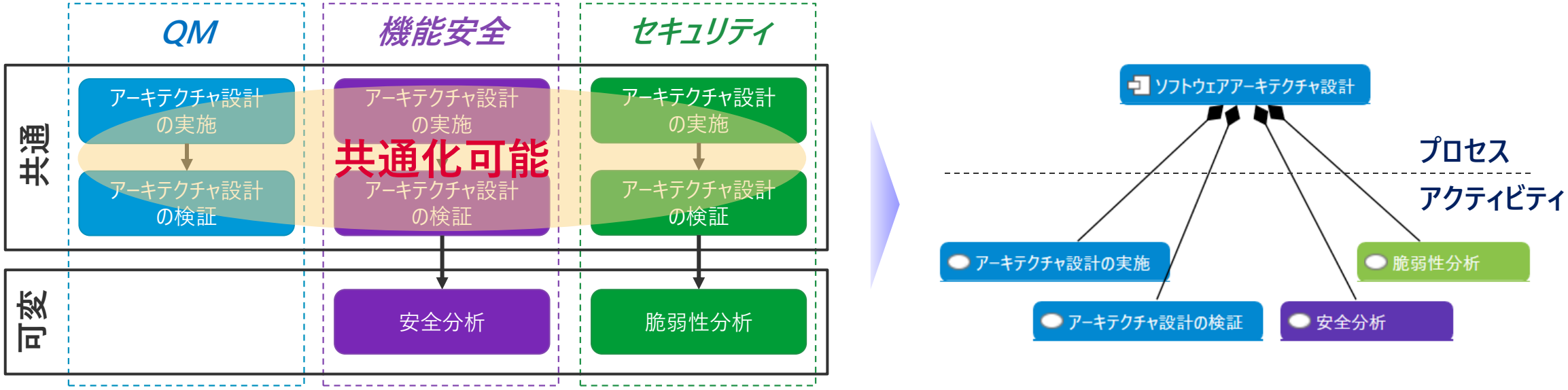
解決策の実践：【Step3】組織標準プロセスの記述（１）

CONFIDENTIAL
関係者外秘

■ メタモデルに基づく組織標準プロセスの定義とプロセス記述

- メタモデルのデータ構造に基づいて、組織標準プロセス（150%モデル）を定義し、文書化する。
- このとき、テーラリングによるプロセスの調整箇所を考慮してプロセスを定義する 。
  - ✓ “共通化可能な部分” と “可変（フィーチャにおける独自）部分” に分離して、プロセスを定義する。
  - ✓ 修整ケースとして、簡略化・結合・代替されるアクティビティ・タスクや成果物などがあれば、それらも定義する。

例）ソフトウェアアーキテクチャ設計プロセス



*150%モデル：全てのフィーチャで必要となるモデル

解決策の実践：【Step3】組織標準プロセスの記述（２）

CONFIDENTIAL
関係者外秘

■ テーラードプロセスの完全性確保

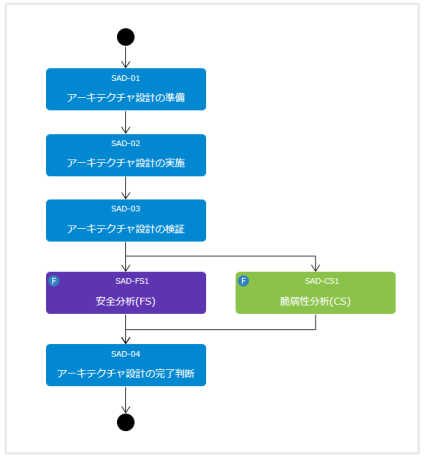
テーラリング時の問題点

- 活動や成果物が省略された場合、活動間・活動と成果物、などの関連が切れることがある。
- この場合、テーラリングされたプロセスが不完全な状態になってしまう。

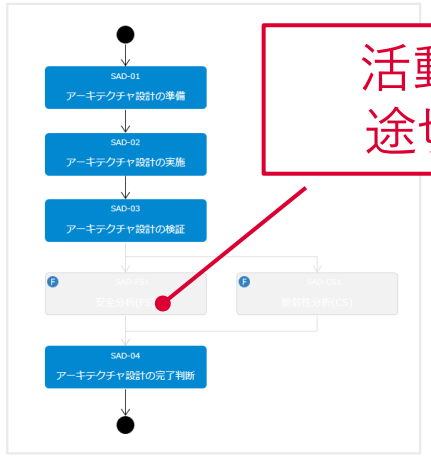
完全性の確保

- テーラリングパターン選択（プロダクト適用）後、テーラリングされたプロセスが完全な状態になるように、プロセス定義を加筆する。

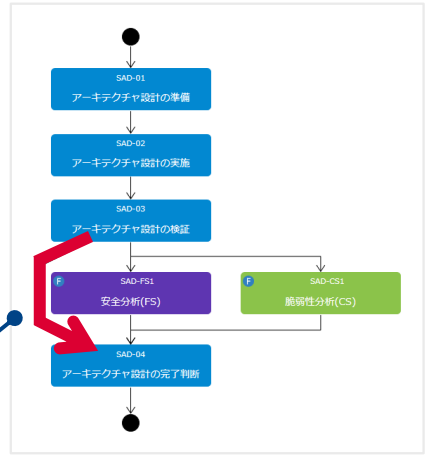
テーラリング前



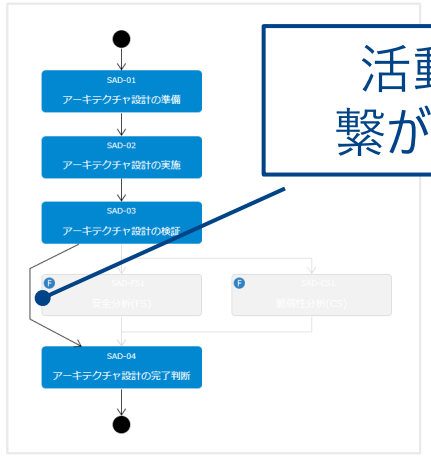
テーラリングパターン選択後（例：QM）



活動間が
途切れる



QMで有効な
関連を追加



活動間が
繋がっている

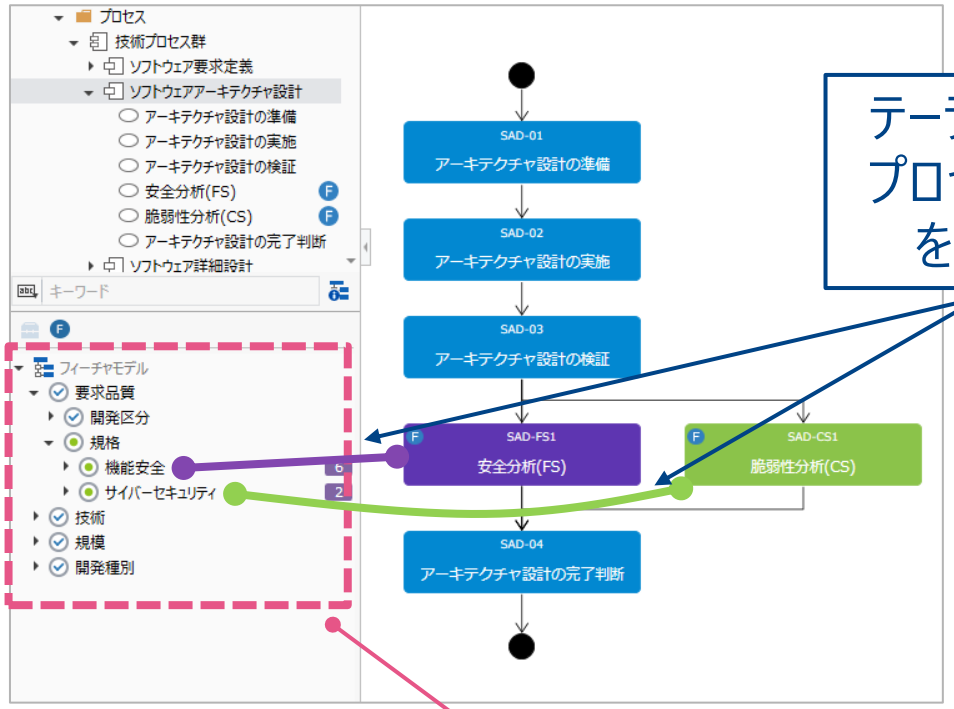
解決策の実践：【Step4】プロセス定義情報へのフィーチャ割り当て

CONFIDENTIAL
関係者外秘

■ フィーチャの割り当て

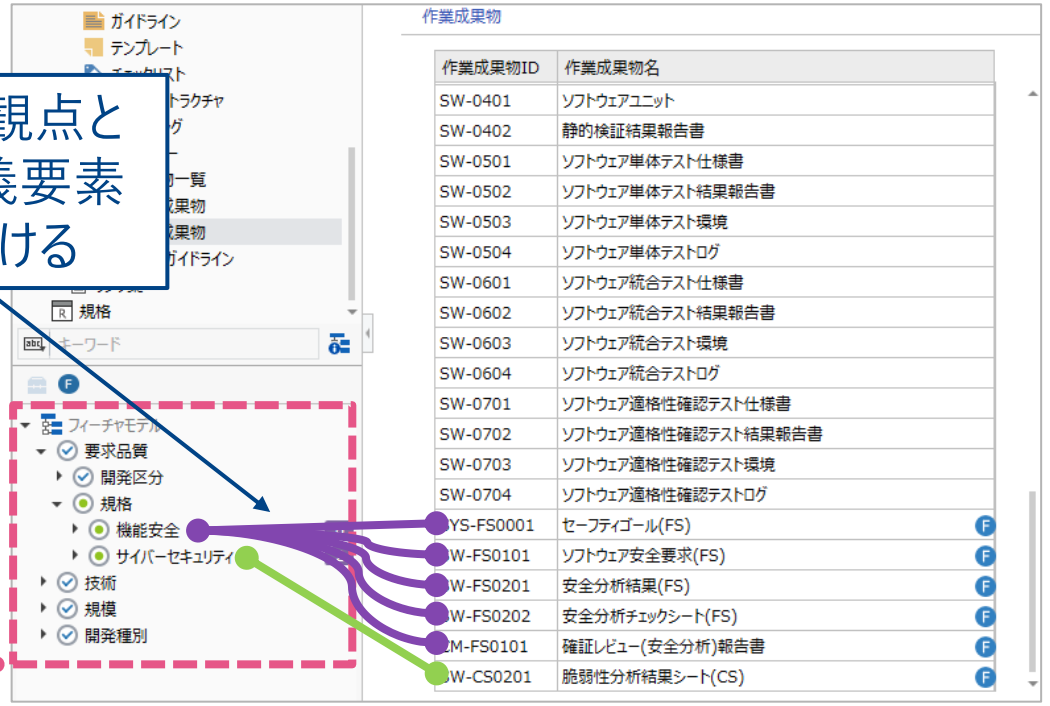
- テーラリング観点（フィーチャ）と関係するプロセス定義情報を特定し、関連付ける。

例）アクティビティの割り当て



テーラリング観点と
プロセス定義要素
を関連付ける

例）作業成果物の割り当て



テーラリング観点（フィーチャ）

解決策の実践：【Step5】プロジェクトでのプロセステーラリング（１）

CONFIDENTIAL
関係者外秘

① テーラリングパターン選択

- プロジェクトの特性に応じて、プロジェクトに**適用するテーラリングパターン（プロダクト）を選択**する。

テーラリング前（150%モデル）

テーラリングパターン選択
（プロダクト適用）

テーラリングパターン選択後

解決策の実践：【Step5】プロジェクトでのプロセステーラリング（２）

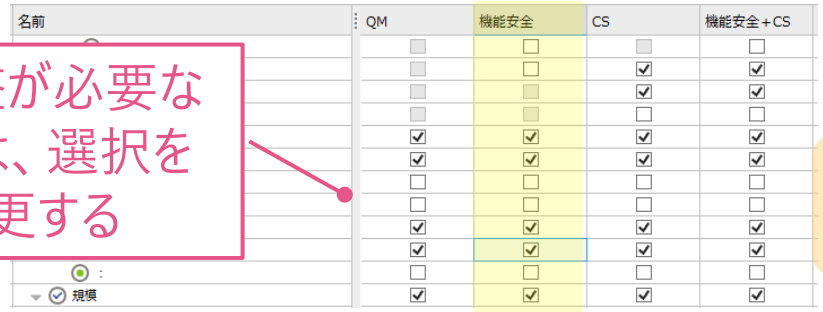
CONFIDENTIAL
関係者外秘

② テーラリングパターンからの小修整

- テーラリングパターンからの小修整は、**テーラリング観点（フィーチャ）を更に選択**することで実現する。
- この**コンフィグレーション結果がテーラリング記録**となる。

テーラリングパターン選択後

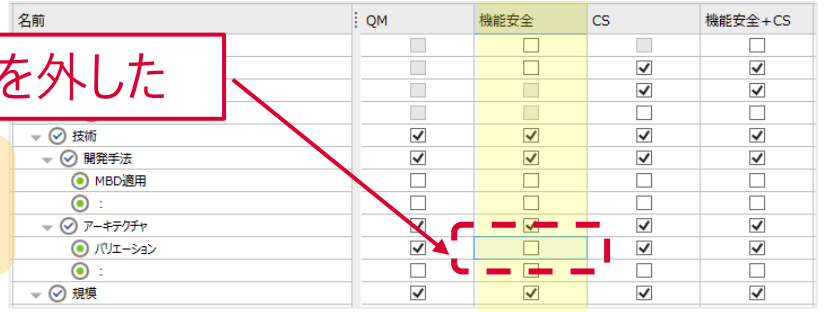
小修整が必要な場合は、選択を変更する



チェックのON/OFFで小修整

テーラリングパターンからの小修整後

チェックを外した



不要なプロセスが消える

解決策の評価（１）

改善点に対し、まず仕組み面の課題解決に取り組んだので、仕組み面を評価する。

■ 仕組みに対する要件は達成できたか

1. **標準的なテーラリング観点（＝プロジェクト特性）をカバーできること**
   - テーラリング観点のフィーチャモデル化では、あらかじめ定義できるテーラリング観点に制約はでなかった。

2. **用意できるテーラリングパターンに制約がでないこと**
   - もともと用意できていた代表的なテーラリングパターンは維持できた。また、仕組み上の制約もない。

3. **省略以外の修整ケースも実現できること**
   - 150%モデルとしてあらかじめプロセス定義しておくことで、”簡略化・統合・代替” を実現できた。
   - “追加” など標準にない修整内容は、プロセス定義書に手を加えることができる仕組みになっている。

4. **プロジェクトプロセスの完全性を確保できること**
   - 150%モデルの定義により、テーラリングされたプロセスの完全性が確保できるようになった。

5. **テーラリング結果の記録が手軽であること**
   - コンフィグレーション結果がテーラリング記録となり、チェックボックスの操作で記録を残せるようになった。

解決策の評価（２）

■ プロセス定義の内容や標準プロセスの適用対象範囲は維持できたか

- 本アプローチを実践した標準プロセスでも、実践前と同じプロセス定義内容を維持できた。
- また、テーラリングパターンによる適用対象範囲の変化はなく、同じプロジェクト・製品を維持できた。

■ 本アプローチは十分運用可能であるか

プロジェクト

- プロセステーラリングを**プロダクト適用**および**コンフィグレーション（チェックボックスの選択）**で実現でき、効率的かつ適正に実施できるようになった。また、操作も容易であった。

SEPG

- ツールのサポートもあり、フィーチャ割り当て・テーラリングパターン定義の工数は、プロセス定義の総工数の中で無視できる程度であった。
- あらかじめプロセス定義を補完（150%モデル定義）しておく工数は必要であった。
  - ✓ 標準プロセス定義として本来必要な工数だったと認識している。

プロセステーラリングへの SPLE 適用は有効かつ運用可能と考える

まとめ と 今後の取り組み

CONFIDENTIAL
関係者外秘

■ まとめ

- **プロセステーラリングをより適正かつ効率良く実施したい**、という改善点があった。
- まずは、**テーラリングの仕組み面の課題解決**に取り組んだ。
- 課題解決のアプローチとして、**組織標準プロセスのテーラリングへのソフトウェアプロダクトラインエンジニアリング（SPLE）の適用**を実践した。
- 具体的な実現方法として、**プロジェクト特性をフィーチャモデル化**し、**プロセス定義のデータ構造を定義**した上で記述し、**プロセス定義情報へフィーチャを割り当て**、**テーラリングパターンを定義**した。
- 本アプローチを実践した結果、**期待通りの効果を確認**することができた。
  本アプローチはプロセステーラリングの有用な実現手段の一つであると判断している。

■ 今後の取り組み

- テーラリング実施者（プロジェクトマネージャ）の**テーラリングスキルの向上／プロセス理解を促進**し、
  今回の仕組みとの相乗効果を図る。
- 仕組み面としての改良として、テンプレートやチェックリストも含めたプロセス資産全体を連動できるようにし、プロジェクトの利便性の向上を図る。

DENSO

Crafting the Core

参考文献

CONFIDENTIAL
関係者外秘

[1] 林好一、ソフトウェアプロダクトラインエンジニアリングをプロセステーラリングに応用する、第25回ソフトウェア品質シンポジウム 2006
[2] 池永直樹、DXアプローチによるプロセス記述 ~プロセス記述の効率化、品質および使用性向上に向けた取り組み~、4th NSPICE Conference
[3] 村上孝,小寺浩司,加藤拓也、複合規格に対応する標準プロセスの構築、5th NSPICE Conference
[4] 2009年度 SQiP研究会 第1分科会、プロセスは定着していますか Part2 ～テーラリングガイド作成の手法の提案～
[5] 2013年度 SQiP研究会 第1分科会、現場作業に密着したテーラリングを実現する「標準プロセスの構造及びプロセス定義」
[6] 相澤武、プロセス定着への取り組み ~テーラリングからはじまるプロセス改善~、SPI Japan 2008
[7] 西村隆,山路厚,伊藤善博,原健三、「メタモデルによる設計情報定義とマルチビューを活用したトレーサビリティ記録方式の提案」、ソフトウェア品質シンポジウム 2021
[8] 堀内一， 大林正晴， 藤川泰之,「メタモデル標準化の意義と最新動向， 前編：-基本的概念と歴史的経過-， 後編：-MOF， XMI 仕様と応用-」， 2002， 情報処理 Vol. 43 No.11,No.12,（社）情報処理学会
[9] Peter Haumer, An Overview to Eclipse Framework Composer Part1:Key Concepts
[10] OMG, Software & Systems Process Engineering Meta-Model Specification Version.2.0
[11] Method Park, Stages Documentation, https://doc.stagesasaservice.com/doku.php/start
[12] OMG, Meta-Modeling and the OMG Meta Object Facility (MOF)
[13] Next Design, https://www.nextdesign.app/