

# JASPIC 第21回 SPI トワイライトフォーラム

## 派生開発における ベースシステムからコア資産への進化

2018年03月22日

JASPIC プロダクトライン分科会

# 発表者

- 筒井 賢 (オムロン ソフトウェア)
- 林 好一 (S R A)
- 遠藤 潔 (日立ソリューションズ)

以上、プロダクトライン分科会

# アジェンダ

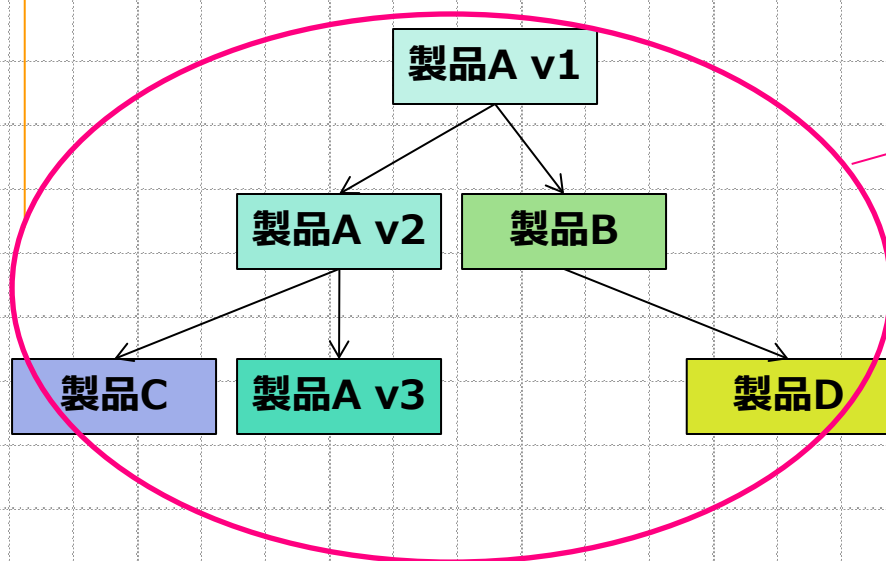
- 派生開発によって得られた資産を、複数システム(製品)に再利用できるコア資産に進化させるためには、どうしたら良いか
  - この課題に対する考え方を、例示を含めながら解説し、その後に実適用に供するための議論を参加者と深める
- \*コア資産: 繰り返し再利用するためのソフトウェア関連成果物

**18:30-19:45 解説**

**19:45-20:30 討論**

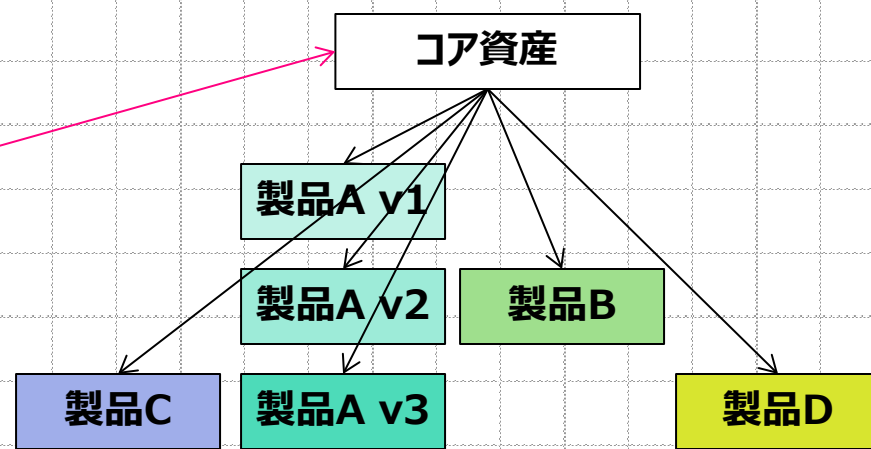
# 派生開発とプロダクトライン開発（PLE）

## 派生開発の典型例



- 過去のシステムを基に再利用を検討する
  - 作ったものを再利用する

## PLEの基本形



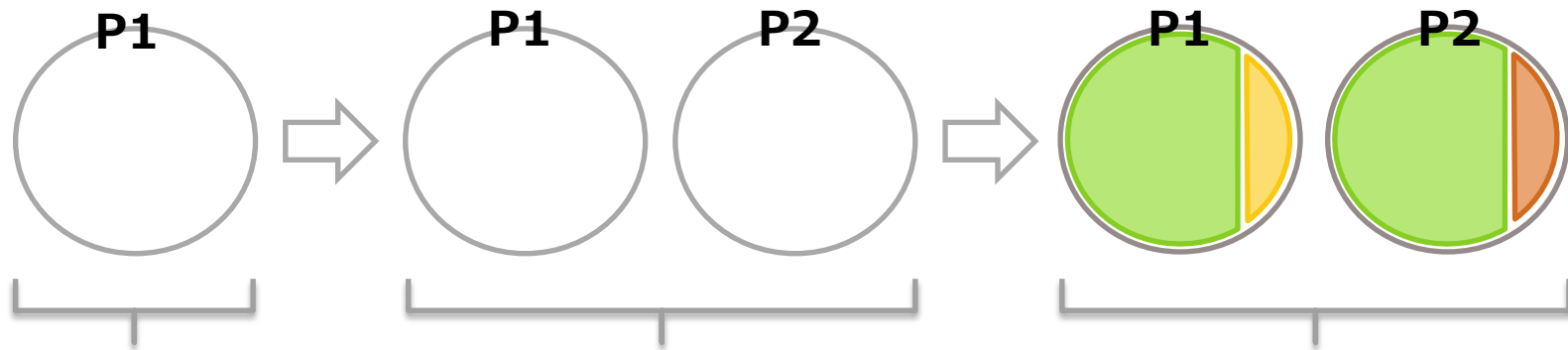
- 将来のシステムを基に再利用を検討する
  - 再利用するものを作る

# 派生をベースとしたシステムを コア資産に進化させるストーリー

# 派生開発時に共通性を把握する

## 製品モデル

(物理の組み合わせモデル)



単に「製品がある」  
というレベルの管理

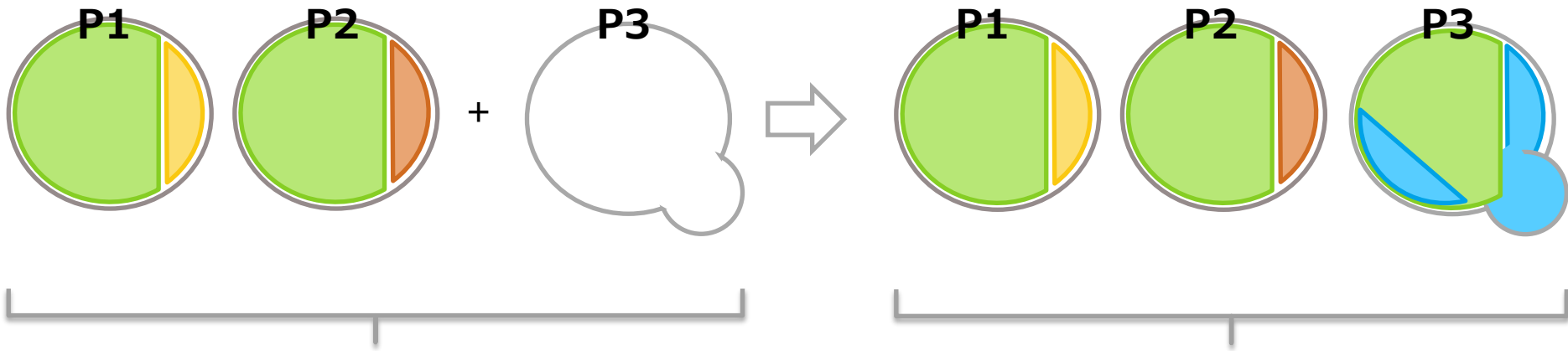
単に「二つの異なる製品がある」  
というレベルの管理

「二つの異なる製品があり、そ  
れらの間には共通の部分とそう  
でない部分がある」というレ  
ベルの管理

# 3機種目の改造で共通部分の非共通化を把握する

## 製品モデル

(物理の組み合わせモデル)



共通部分を持つ二つの製品に三つ目の製品を加える

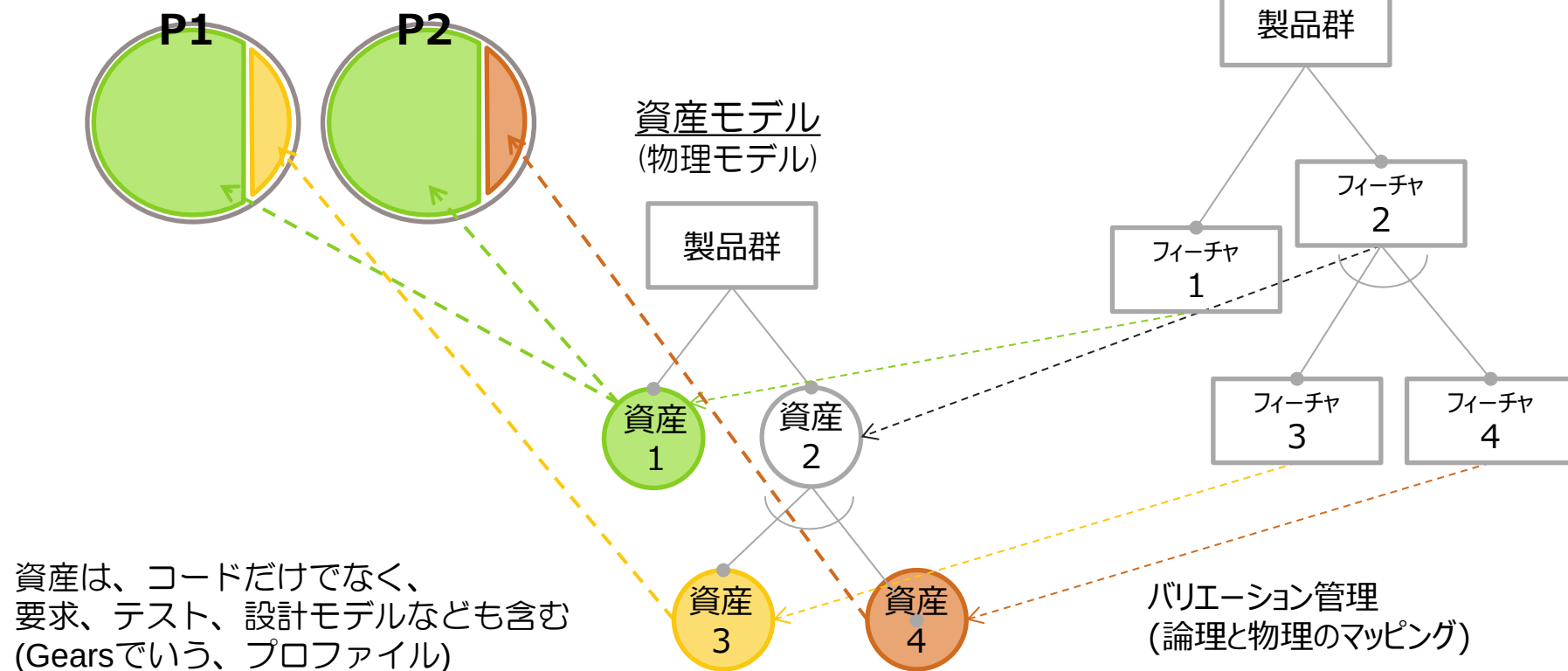
P1とP2の間の共通部分の、  
一部は三製品間で共通部分となり、  
一部はP3では異なる部分となった。  
またP3には、P1・P2のどちらにも該当  
しない部分が追加された

# 製品間共通性の把握との関係

## 用語の定義

製品モデル  
(物理の組み合わせモデル)

フィーチャモデル  
(論理モデル)

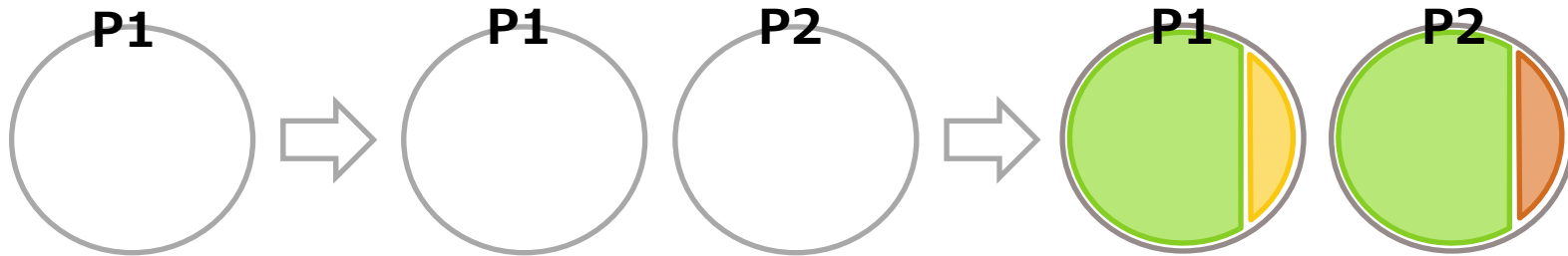




# 2機種目の開発からフィーチャモデルを作り始める

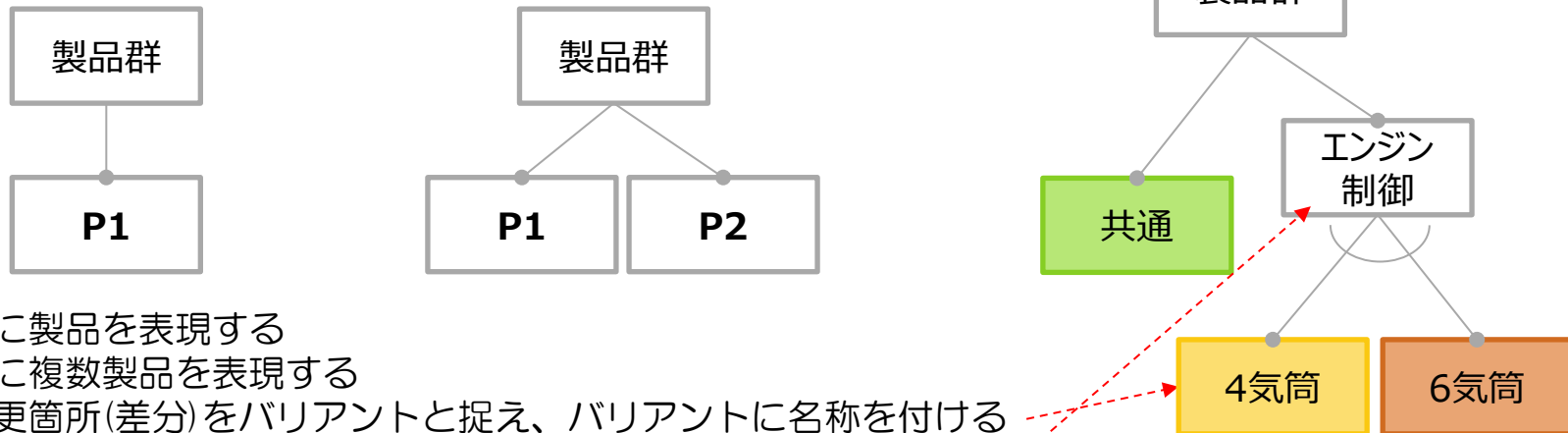
## 製品モデル

(物理の組み合わせモデル)



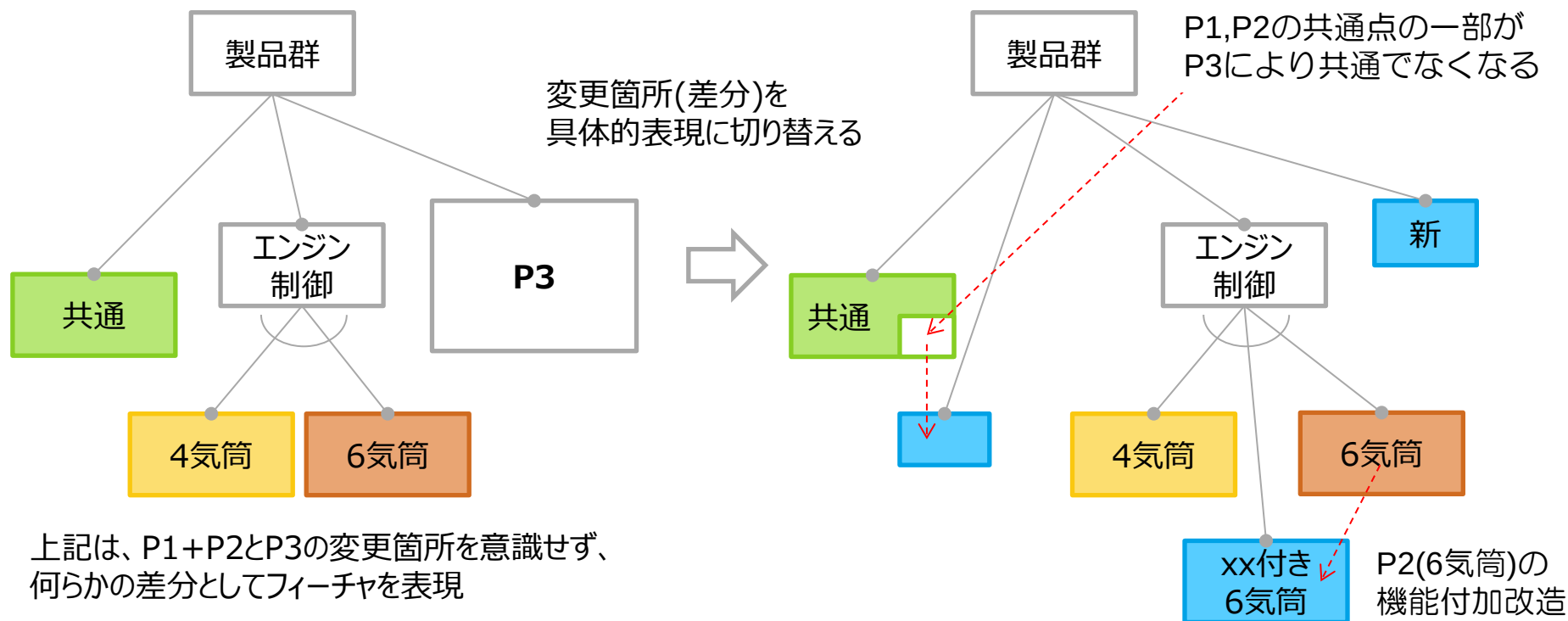
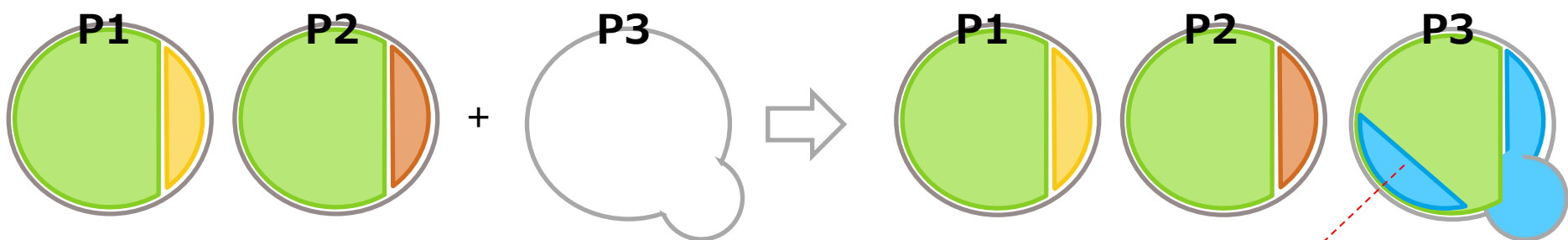
## フィーチャモデル

(論理モデル)



1. 単に製品を表現する
  2. 単に複数製品を表現する
  3. 変更箇所(差分)をバリエーションと捉え、バリエーションに名称を付ける (4気筒, 6気筒)
  4. バリエーションの上位概念を考え、名称を付ける (エンジン制御)
- ※この時点で、注目すべきは物理モデルから論理モデルに替わっている

# 3機種目改造によるフィーチャのさらなる分離

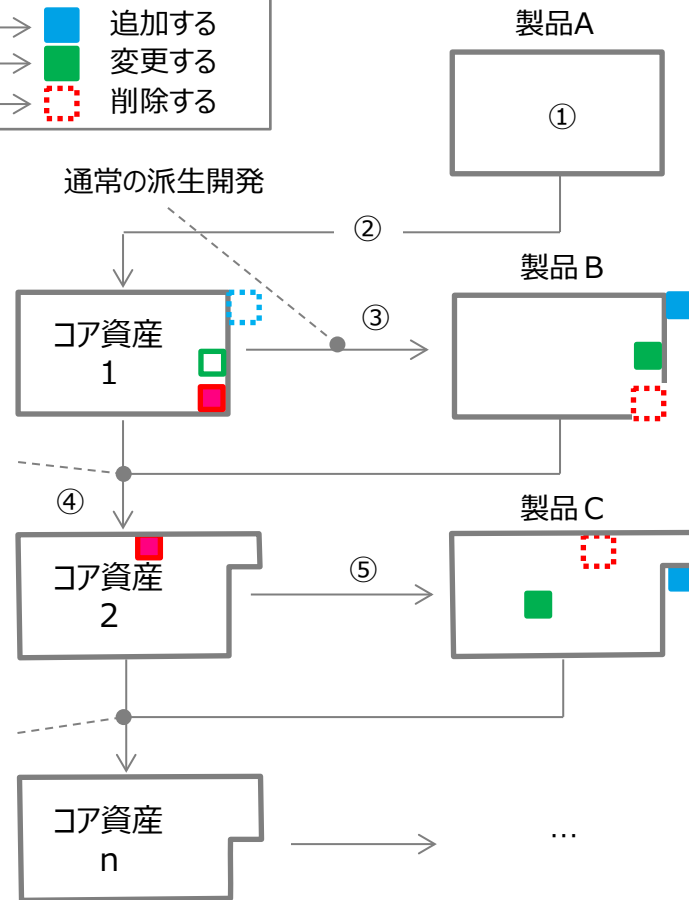
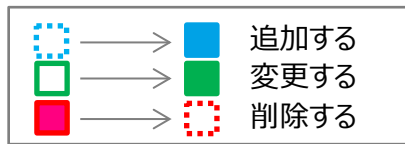


# コア資産に進化させる ストーリーの例示

# 派生開発のタイミングで意識することから始まる

□変更するときに、以下の3点を意識しておくことが重要

1)「どこに」 2)「なにを」 3)「どうした(追加/変更/削除)」

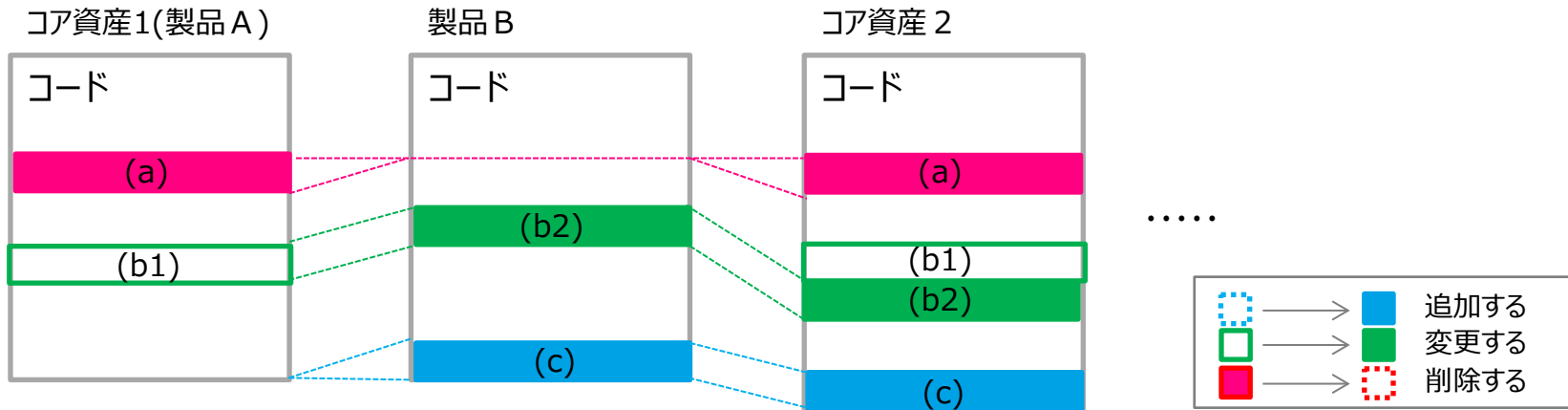


- ① 製品Aを新規開発する
- ② 製品Aをコア資産 1とする
- ③ 製品Bを、コア資産 1 から派生開発する
- ④ 製品Bでコア資産から追加/変更/削除されたものをコア資産 1 に統合してコア資産 2とする
- ⑤ 製品Cをコア資産 2 から派生開発する
- ...

コア資産 nとは、繰り返して製品開発したものを継続的に統合し、次の開発ができる派生元となるものを指す

# 繰り返す開発で差分をためていく(コード資産)

## □製品 A、B…を網羅するコード(たとえば #ifdef)が組み込まれる



### コア2のコード (コア1と製品Bの統合)

```
#ifdef 製品=A
(a) ■
#endif
#ifdef 製品 = A
(b1) □
#elif 製品= B
(b2) ■
#endif
#ifdef 製品=B
(c) ■
#endif
```

- コア資産1 (製品Aと同一) から製品Bを派生開発した際の差分 (追加/変更/削除) をコア資産1に統合してコア資産2を作る

# 繰り返す開発で差分をためていく(要求資産)

## □コア資産に製品Bを表現する要求仕様が組み込まれる

コア1 (製品A)

製品B

コア2

|                                                                                     |
|-------------------------------------------------------------------------------------|
| USDM コア1→B<br>before / after                                                        |
| <span style="color: red;">■</span> 要求を削除                                            |
| <span style="color: green;">□</span> 要求を <span style="color: green;">■</span> 要求に変更 |
| <span style="color: blue;">■</span> 要求を追加                                           |

- コア資産1 (製品Aと同一) から製品Bを派生開発した際の差分 (追加/変更/削除) をコア資産1に統合してコア資産2を作る

コア資産 (統合製品)

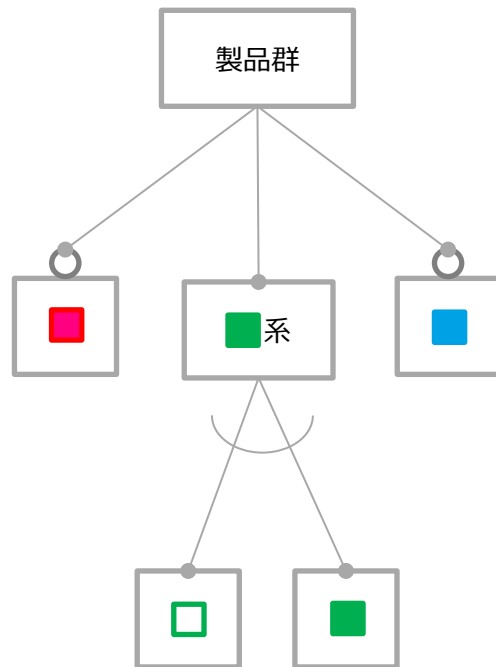


コア2の要求 (コア1と製品Bの統合)

|                                          | 製品A | 製品B |  |
|------------------------------------------|-----|-----|--|
| <span style="color: red;">■</span> 要求    | ○   |     |  |
| <span style="color: green;">■</span> 系要求 | ○   |     |  |
|                                          |     | ○   |  |
| <span style="color: blue;">■</span> 要求   |     | ○   |  |
|                                          |     |     |  |

# どの場合もフィーチャモデルで対応が表現できる

## □追加・変更・削除は「可変性」であり、統一的にモデリングできる



凡例

- 製品によって実現したりなかったりするオプションフィーチャを表わす
- 製品によってどれかを実現する選択枝フィーチャを表わす

コア2のコード（コア1と製品Bの統合）

```
#ifdef 製品=A
(a) ■
#endif
#ifdef 製品 = A
(b1) □
#elif 製品= B
(b2) ■
#endif
#ifdef 製品=B
(c) ■
#endif
```

コア2の要求（コア1と製品Bの統合）

|      |   | 製品A | 製品B |  |
|------|---|-----|-----|--|
| ■要求  |   | ○   |     |  |
| ■系要求 | □ | ○   |     |  |
|      | ■ |     | ○   |  |
| ■要求  |   |     | ○   |  |
|      |   |     |     |  |

- 先のコードの例でも要求の例でも、同じ可変性表現を充てることができる
- 可変性の表現には「フィーチャモデル」が使える
- フィーチャモデルでは、製品等具体的な変更結果ではなく、「何が何故変わるのか」を表現して、一般性を持たせる（再利用性を高める）

□ 「□」が「■」に変わるのは、「製品A」と「製品B」だからではなく、例えば、「ネットワーク」という上位概念があり、その種類として「CAN」と「Ether」があるからだ、という風に

# どこからでも誰でも始められるフィーチャモデリング

## □ 同じ製品群のあらゆる種類の資産は、ひとつのフィーチャモデルと対応

要求…仕様

USDM記述、ユースケース、etc

|      | 製品A | 製品B |  |
|------|-----|-----|--|
| ■要求  | ○   |     |  |
| ■系要求 | ○   |     |  |
| ■要求  |     | ○   |  |
|      |     |     |  |

設計

テスト資産

...

マニュアル

(何でも)

- どの役割の人でもどのタイミングでもフィーチャモデルを作れる
- そのフィーチャモデルをどの役割の人にも展開できる
- どの役割の人でもできるなら、大元の元である要求の可変性から整理しよう！



# 要求の再構成 1/2

□ 複数の要求をコア資産に統合する際、再構成が必要になる場合がある

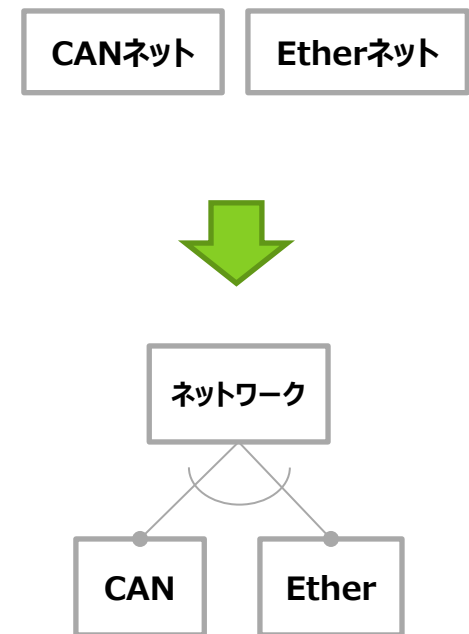
□ 例1：以下の二つの要求の統合

- CANネットを介して通信すること
- Etherネットを介して通信すること

□ 例1の場合には、共通の上位概念をくり出して以下のように再構成する

- ネットワークを介して通信すること（共通要求）
- CANネットを介して通信すること（選択肢要求）
- Etherネットを介して通信すること（選択肢要求）

例1：



## 要求の再構成 2/2

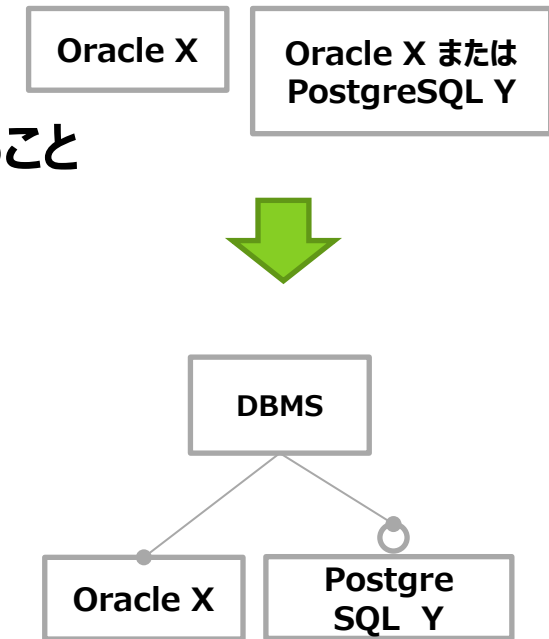
### □ 例2：以下の二つの要求の統合

- Oracle X に接続できること
- Oracle X または PostgreSQL Y に接続できること

### □ 例2の場合には、共通の上位概念と、さらに共通の要素をくり出して、以下のように再構成する

- **DBMS** に接続できること（共通要求）
- **Oracle X** に接続できること（共通要求）
- **PostgreSQL Y** に接続できること（オプション要求）

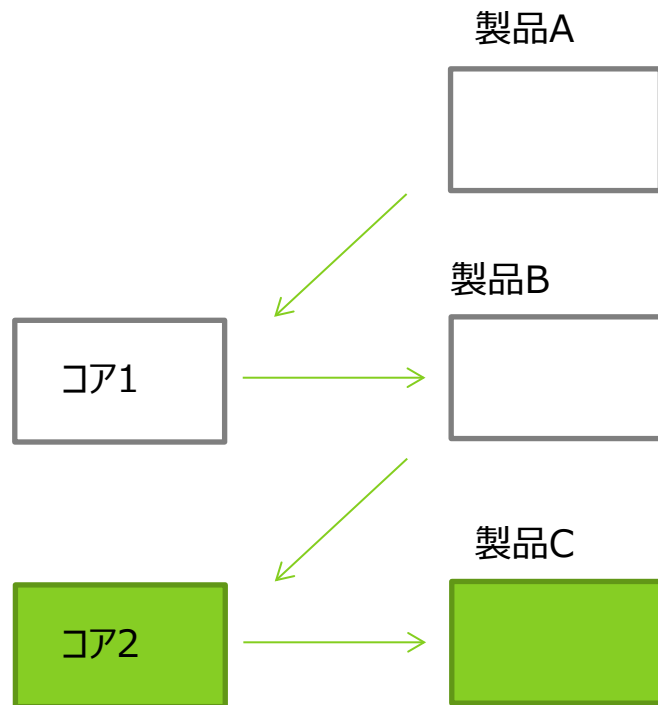
例2：



# 派生開発のタイミングで意識することから始まる

□ 変更するときに、以下の3点を意識しておくことが重要

①「どこに」 ②「なにを」 ③「どうした(追加/変更/削除)」



- 製品Aを新規開発する
- 製品Aをそのままコア資産1とする
- 製品Bを、コア資産1から派生開発する
- 製品Bでコア資産1から**追加/変更/削除されたもの**を、コア資産1に統合してコア資産2とする
- 製品Cをコア資産2から派生開発する
- ...

# むすび

- 膨大な過去資産から共通点をまとめるのはしんどい
- まず重要な二個の製品の差異（物理的な違い）を捉える
  - \* なるべく似ているものを対象にする
- それをフィーチャモデル（論理的な違い）で捉え直す
- 以降、次のステップを続けていく
  - 追加、変更、削除（物理差異）を捉える
  - 物理差異をフィーチャモデルで表現（論理差異）する
    - \* 必要に応じて、フィーチャモデル（論理差異）を他の種類の成果物にも関連づける
- このように差異を整理し続けることで、再利用容易なソフトウェア関連成果物（＝コア資産）が充実していく
  - \* できたところから次の開発で活用していける

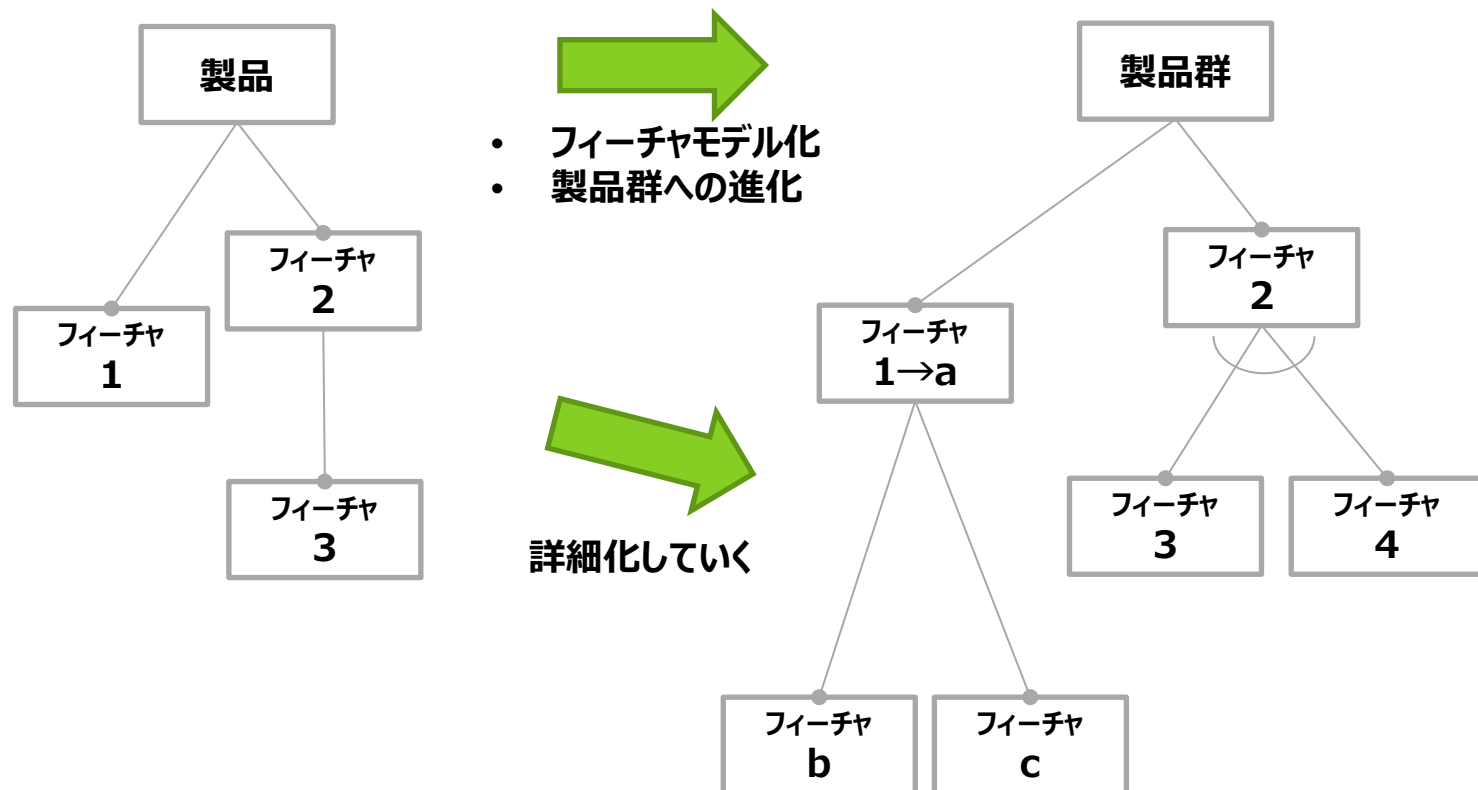
# おまけ

# フィーチャモデルの進化のさせ方

□ 製品を主要機能で表現する

□ 製品群が発展するに伴い、機能が自然と分割されてくる

□ 初期には明らかでなかった非機能フィーチャが見えてくる(性能など)



# 要求コア資産の進化のさせ方

## □ コア資産のUSDMから個別製品の開発に使うUSDMを作る

コア2

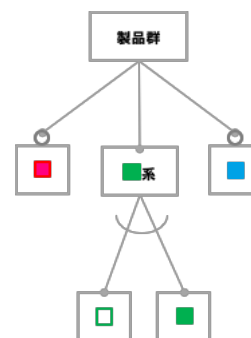
製品C

### USDM コア2 → 製品C

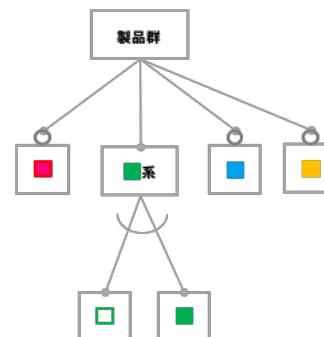
既に存在する要求の取捨選択  
(流用および削除)に加えて  
before / after(追加および変更)  
を行なう

|      | before/after |
|------|--------------|
| ■要求  | ← (流用)       |
| ■系要求 | □ - (非選択)    |
|      | ■ ○ (選択)     |
| ■要求  | 削除           |
| ■要求  | 追加           |
|      |              |

## □ コア資産のUSDMと今回開発したUSDM(before / after)を統合して新たなコア資産を作る



|      | 製品A | 製品B |  |
|------|-----|-----|--|
| ■要求  | ○   |     |  |
| ■系要求 | □   | ○   |  |
|      | ■   | ○   |  |
| ■要求  |     | ○   |  |
|      |     |     |  |



|      | 製品A | 製品B | 製品C |  |
|------|-----|-----|-----|--|
| ■要求  | ○   |     | ○   |  |
| ■系要求 | □   | ○   | ○   |  |
|      | ■   |     | ○   |  |
| ■要求  |     | ○   |     |  |
| ■要求  |     |     | ○   |  |
|      |     |     |     |  |

**トワイライトフォーラムへご参加いただき、  
ありがとうございました**