

製品群開発におけるXDDPの適用と3つの課題

株式会社エクスマーション 吉元 崇



<https://www.exmotion.co.jp/>

お問い合わせ先：info@exmotion.co.jp



1 製品群開発現場を悩ませる困りごと

2 XDDPを適用しても発生する問題

3 3つの課題 / 3つの取り組み

4 取り組みの効果・効能

5 まとめ



1

製品群開発現場を悩ませる困りごと

2

XDDPを適用しても発生する問題

3

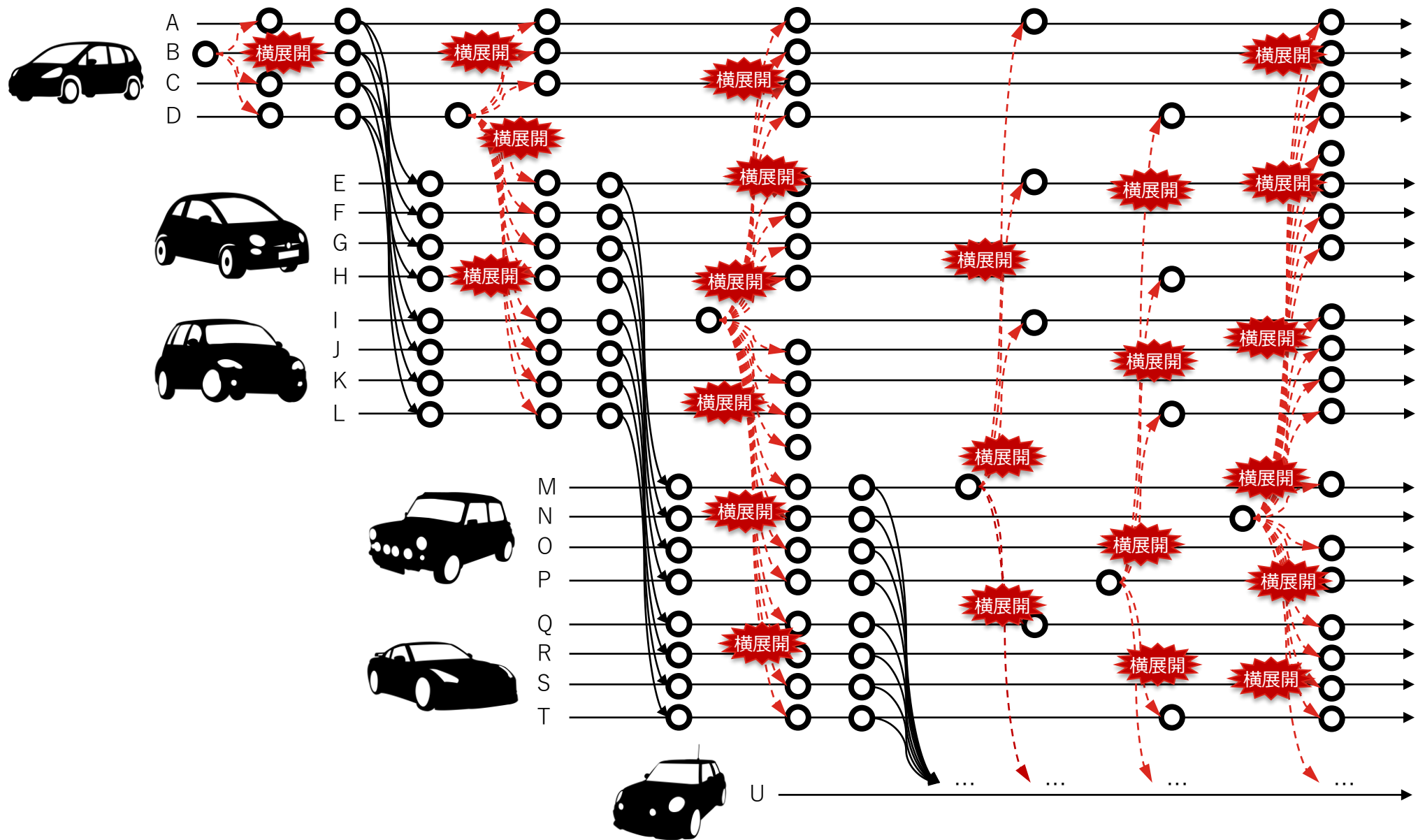
3つの課題 / 3つの取り組み

4

取り組みの効果・効能

5

まとめ





横展開対応に
苦勞されていませんか？

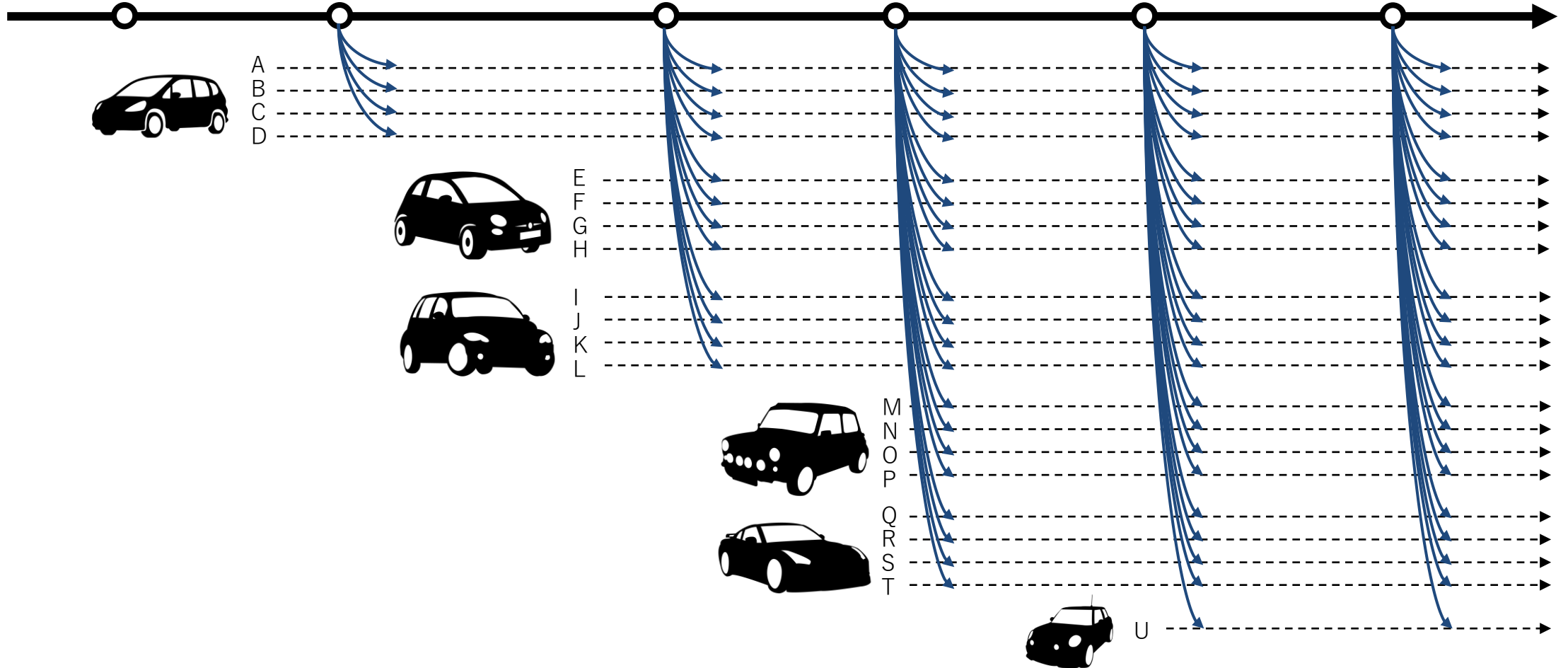


横展開対応

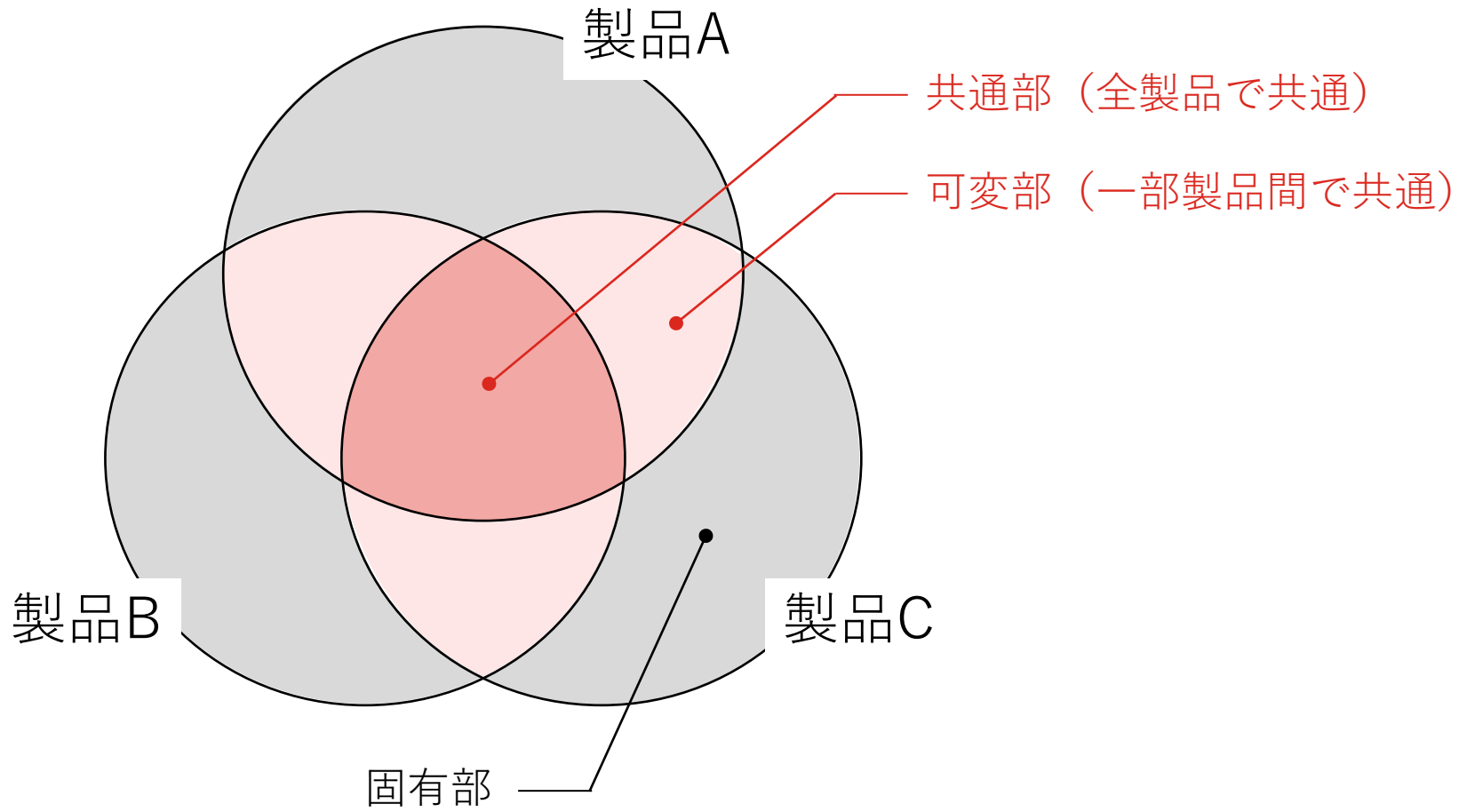


SPL開発で解決！

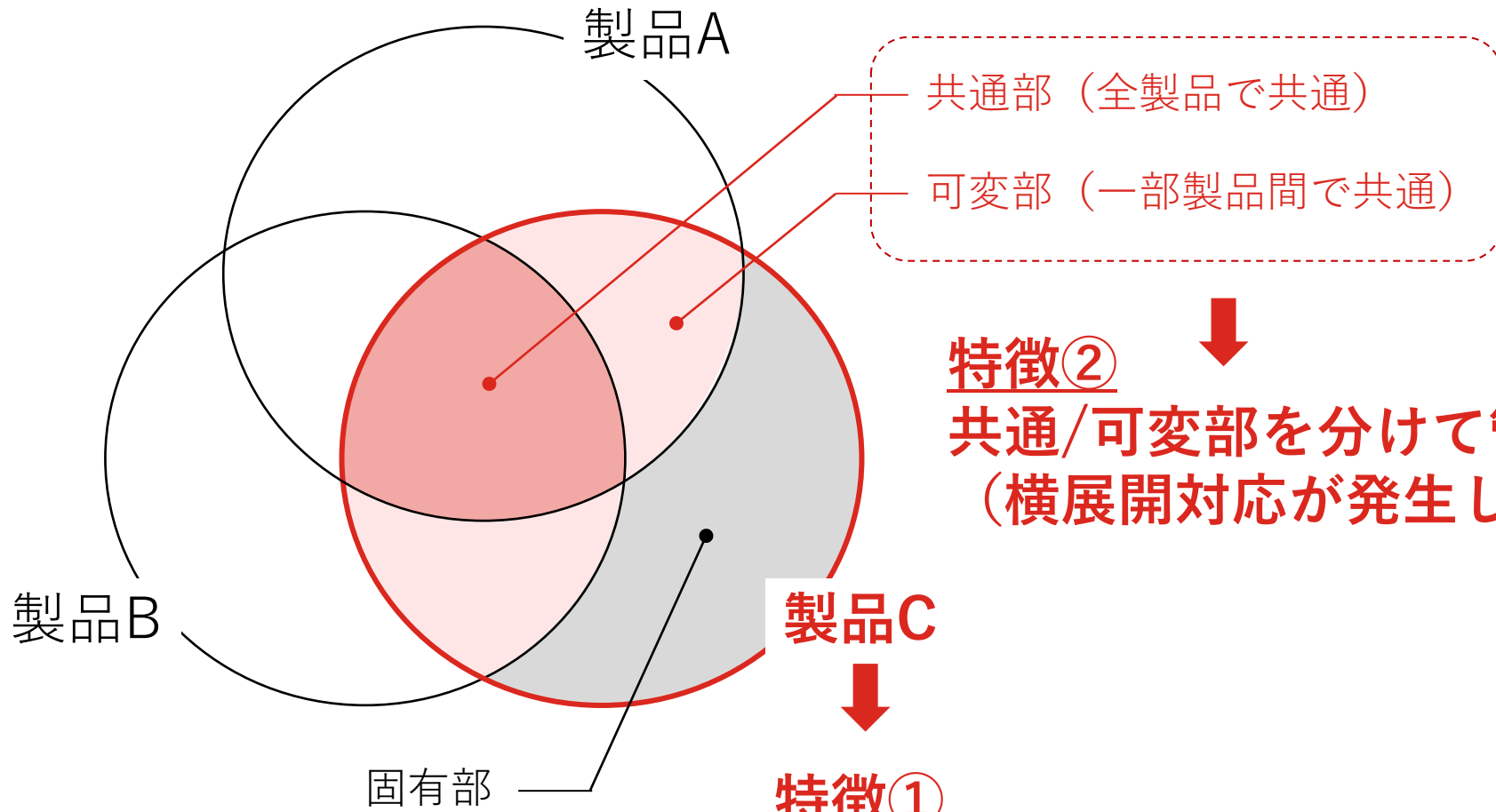
SPL開発とはどんな開発か？



SPL開発とはどんな開発か？



SPL開発とはどんな開発か？



特徴②

共通/可変部を分けて管理するので重複がない
(横展開対応が発生しない)

特徴①

必要なパーツを組み合わせて製品を導出できる



横展開対応



SPL開発で解決！

とは言うものの…



現場からはこんなご意見が

SPL開発なんてできませんよ。
横展開対応は力技で解決しています。

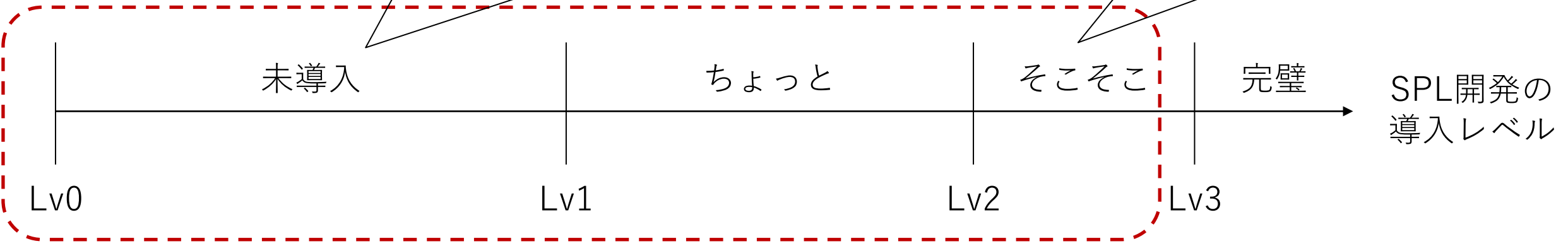
SPL開発やってても、
横展開対応ありますよ。

今日は、このような方々に向けたお話です



SPL開発なんてできませんよ。
横展開対応は力技で解決してます

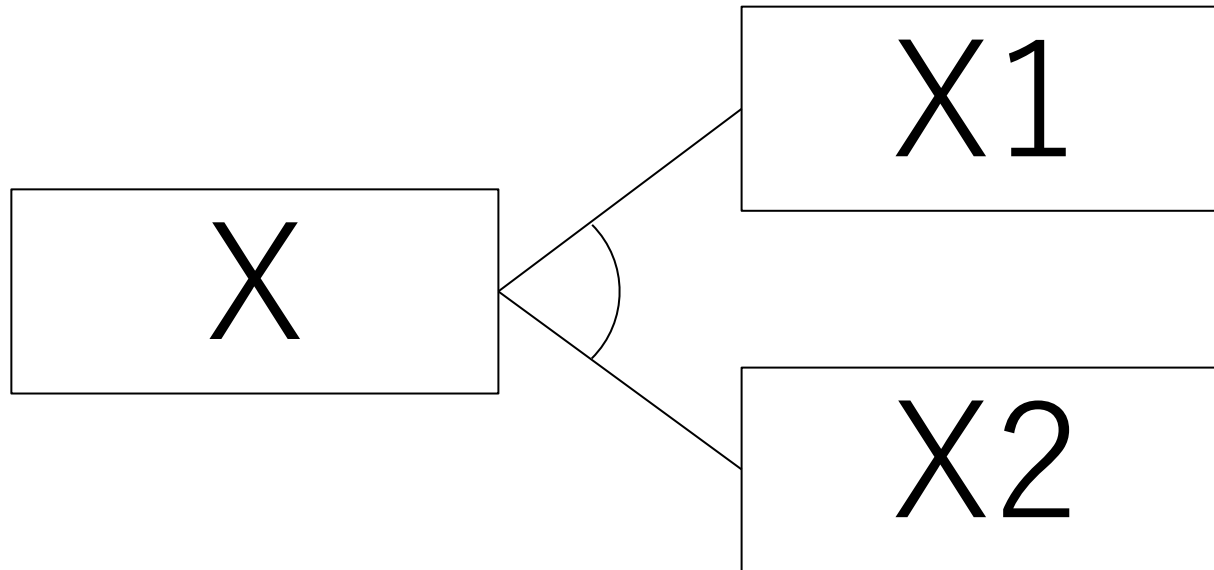
SPL開発やってても、
横展開対応ありますよ。



横展開に苦労される方々



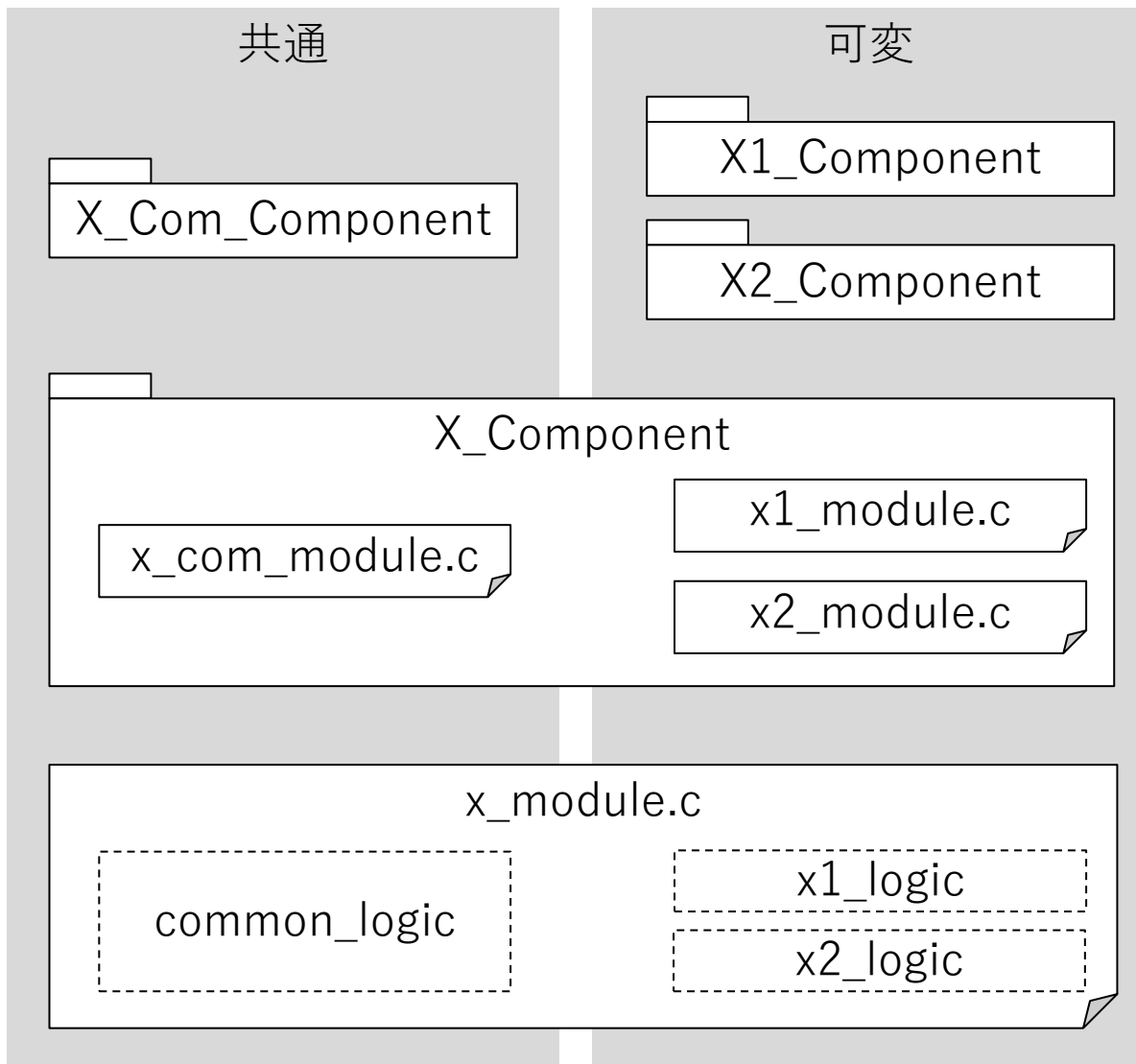
Q.



共通資産の構造、こうなっていないませんか？

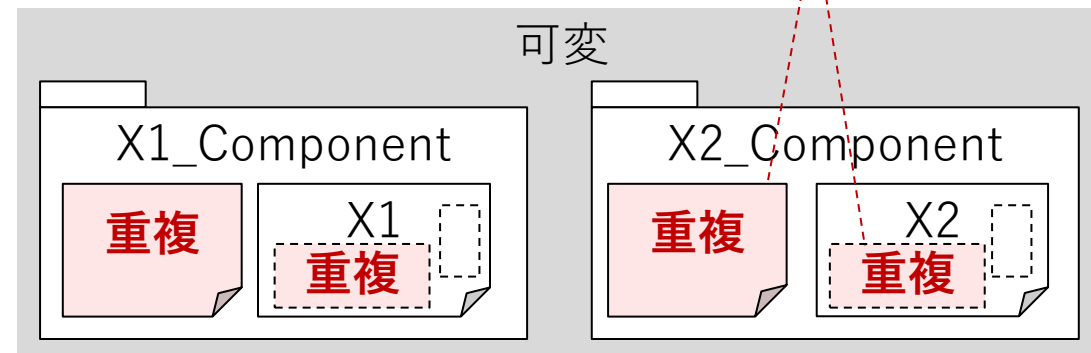


理想

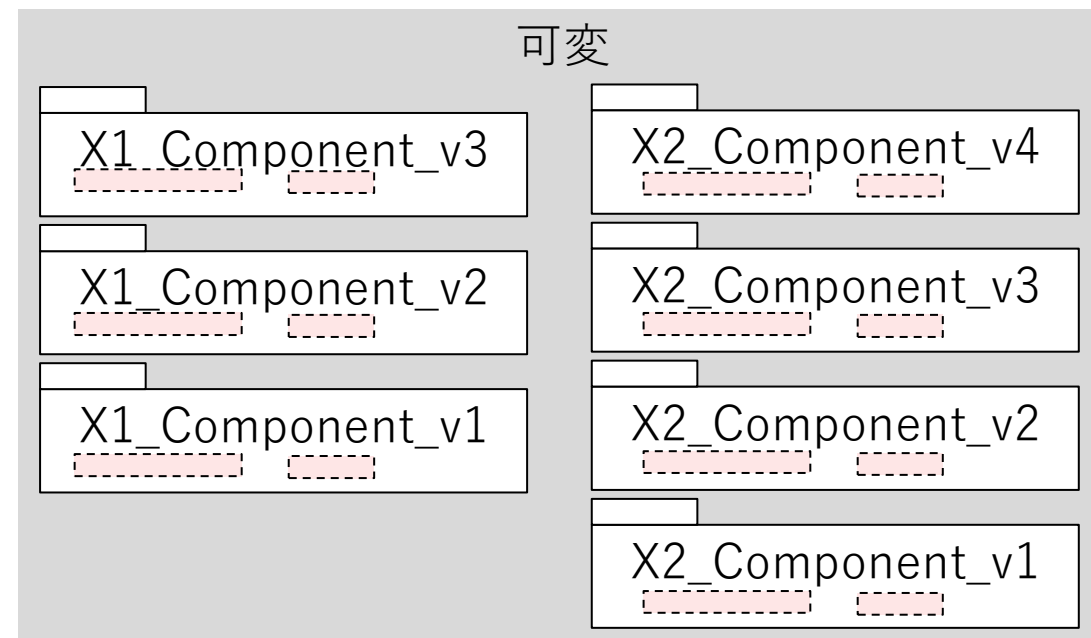


現実解①

理由) 品質が担保された既存資産をわざわざ
共通/可変に再構造化するメリットがない



現実解②



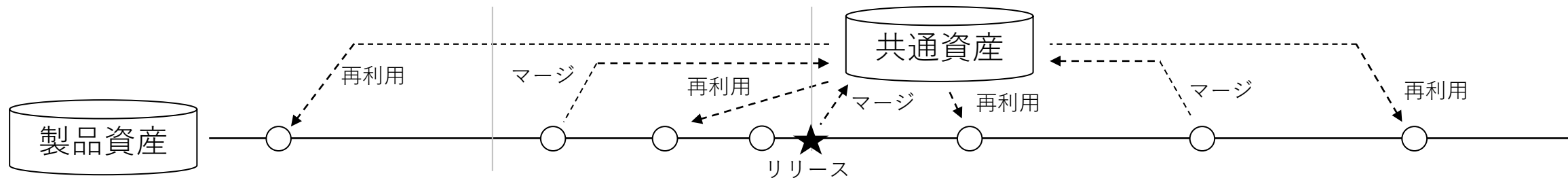
こんな資産管理、やっていませんか？（1：派生開発型）

凡例) ○ 変更



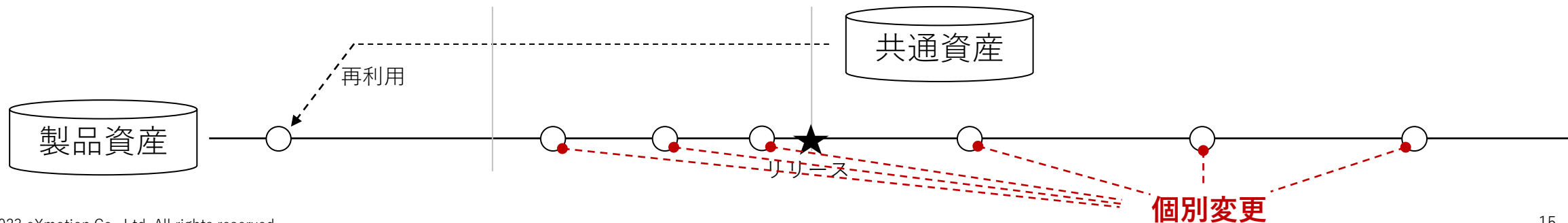
理想（SPL開発）

| | 開発前 | 開発中 | リリース後（保守） |
|---------|----------|-----------|-----------|
| 製品資産の扱い | 共通資産を再利用 | 共通資産と同期する | |



現実解（派生開発型）

| | 開発前 | 開発中 | リリース後（保守） |
|---------|----------|-------------------|-----------|
| 製品資産の扱い | 共通資産を再利用 | 共通資産とは同期せず、個別管理する | |



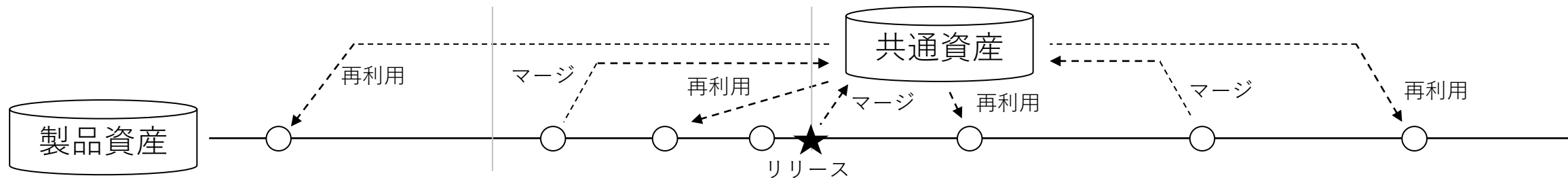
こんな資産管理、やっていませんか？（2：統合開発型）

凡例) ○ 変更



理想（SPL開発）

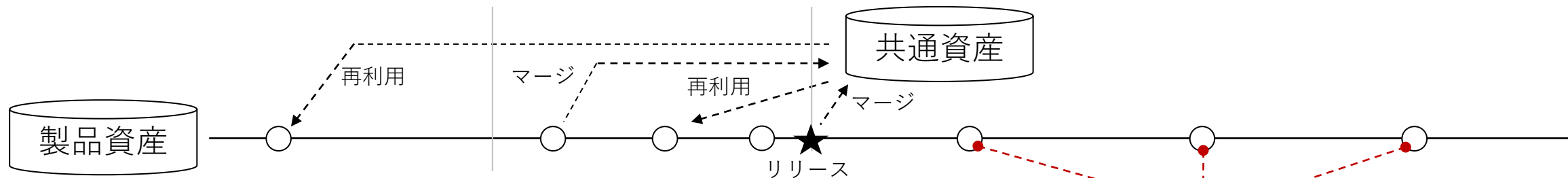
| | 開発前 | 開発中 | リリース後（保守） |
|---------|----------|-----------|-----------|
| 製品資産の扱い | 共通資産を再利用 | 共通資産と同期する | |



現実解（統合開発型）

理由) 対象顧客に関係ない共通資産の変更を、即時反映することは現実的に厳しい

| | 開発前 | 開発中 | リリース後（保守） |
|---------|----------|-----------|-------------------|
| 製品資産の扱い | 共通資産を再利用 | 共通資産と同期する | 共通資産とは同期せず、個別管理する |



個別変更

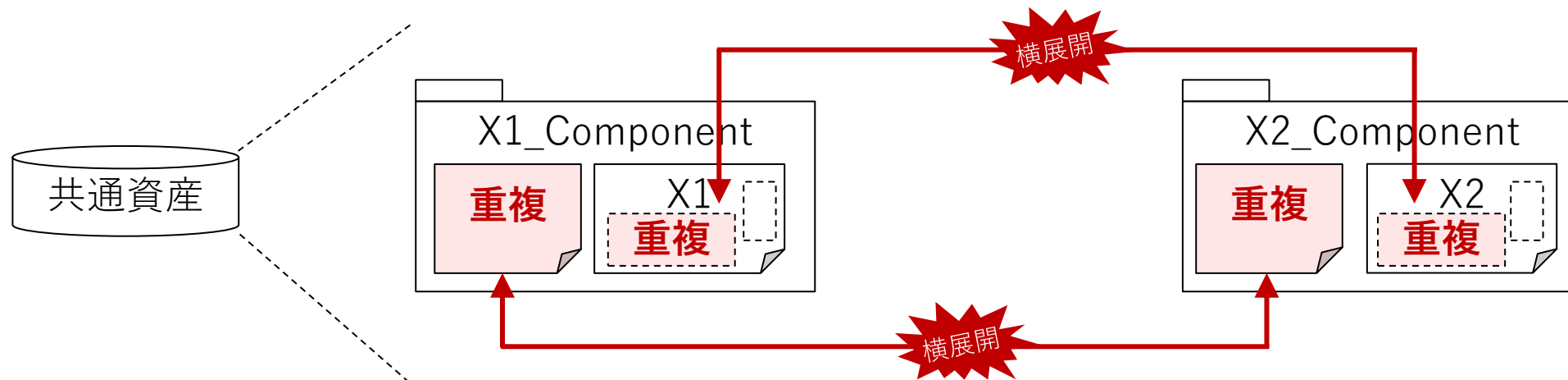
やむを得ない横展開対応

凡例) ○ 変更

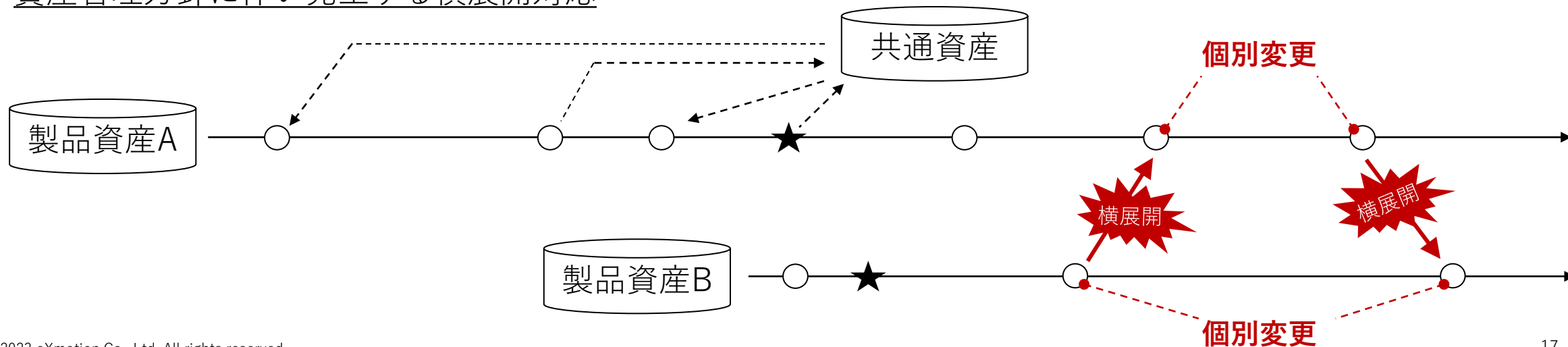


- このような開発方針だと、共通資産の中でも、製品資産間でも、横展開が発生してしまう

共通資産の構造に伴い発生する横展開対応



資産管理方針に伴い発生する横展開対応





1

製品群開発現場を悩ませる困りごと

2

XDDPを適用しても発生する問題

3

3つの課題 / 3つの取り組み

4

取り組みの効果・効能

5

まとめ

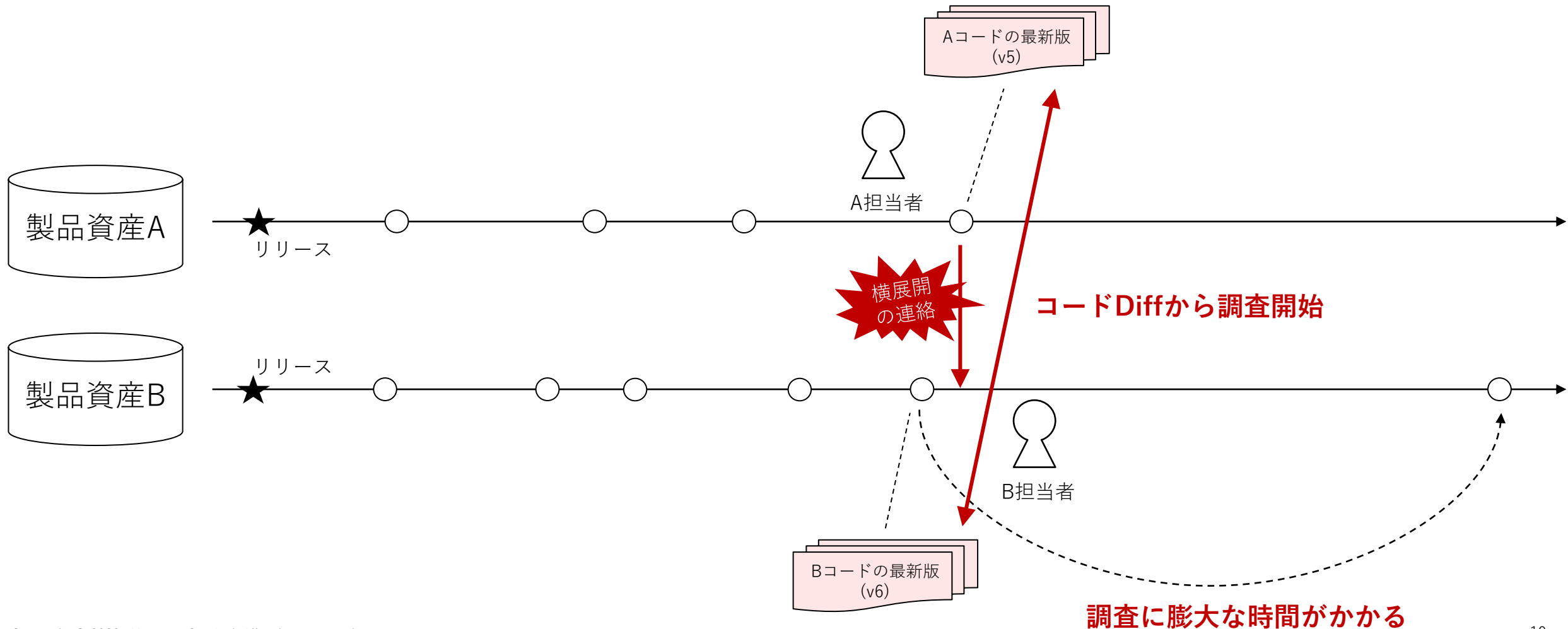
横展開対応時の差分調査に時間がかかる（1）

凡例) ○ 変更

調査対象



- XDDPを適用していなければ最悪で、差分調査に膨大な時間がかかる



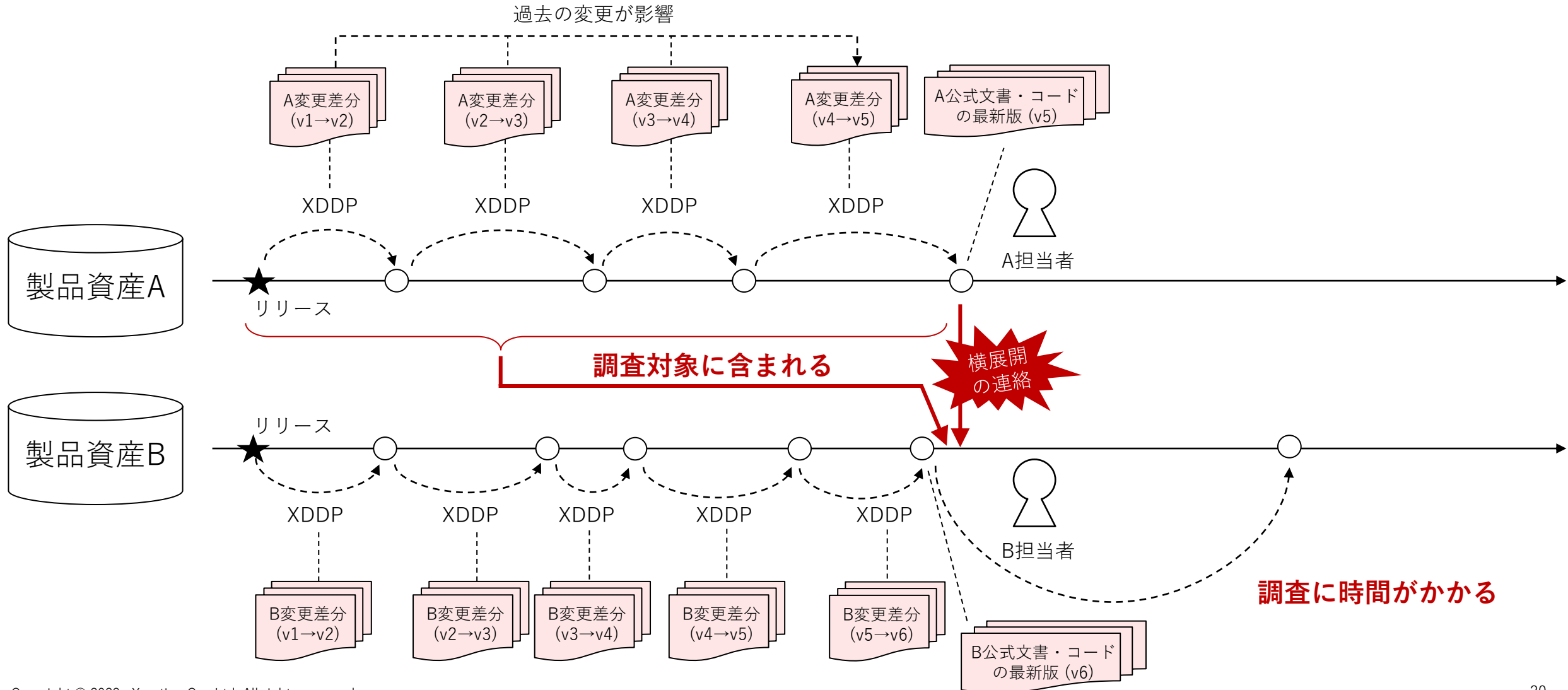
横展開対応時の差分調査に時間がかかる (2)

凡例) ○ 変更

調査対象



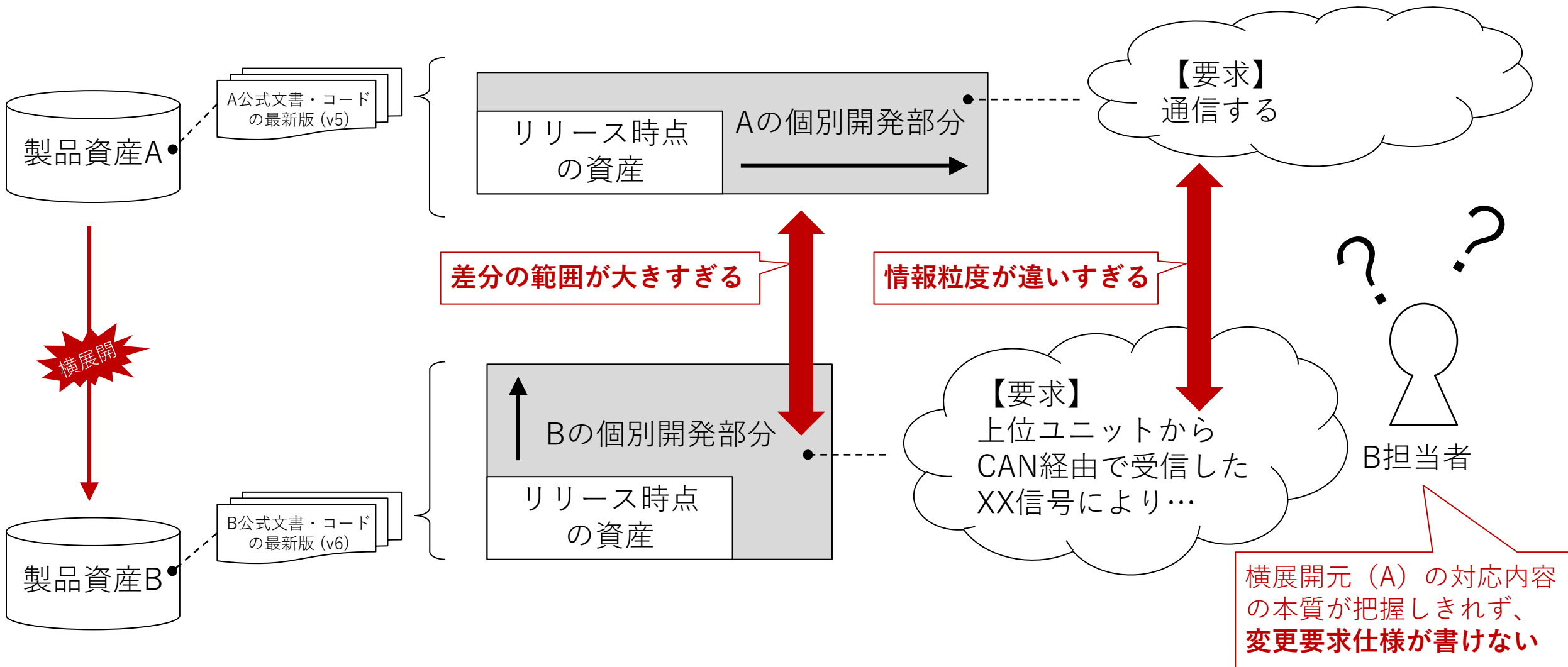
- XDDPを適用していても、差分調査に時間がかかる





差分調査に時間がかかる原因

- 例え公式文書があったとしても、差分範囲が大きく、情報粒度が異なるため、調査に難航する





事例（本当にあった怖い話）

- 事例 1) 1機種 の試作で変更開発に追われ、2年トータル の投入工数は**120人月**

| ドメイン | コード規模 | 製品数 | 共通資産 | XDDP | 変更頻度 | 変更規模 | 変更内容 |
|--------|-------|----------|------------|------|-------|--------|-----------|
| パワトレ制御 | 20万行 | 過渡期は20以上 | 担当者の頭の中に存在 | 適用なし | 1件/3日 | 500行/件 | 主にパラメータ調整 |

- 事例 2) 量産直前に横展開が発生し、全製品トータル の投入工数は**300人月**

| ドメイン | コード規模 | 製品数 | 共通資産 | XDDP | 変更規模 | 変更内容 |
|--------|-------|------|-------------|------|-------------------------|--------|
| 駆動部品制御 | 10万行 | 10以上 | 存在する(重複を許容) | 適用あり | チケット総数10,000件(急な横展開が発生) | ロジック変更 |

- 他にも、これよりひどい現場も…



1 製品群開発現場を悩ませる困りごと

2 XDDPを適用しても発生する問題

3 3つの課題 / 3つの取り組み

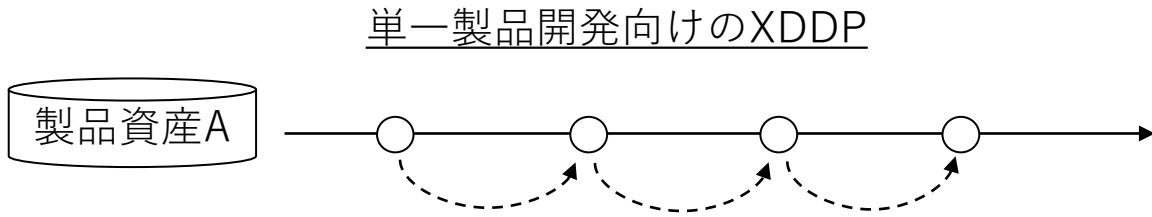
4 取り組みの効果・効能

5 まとめ

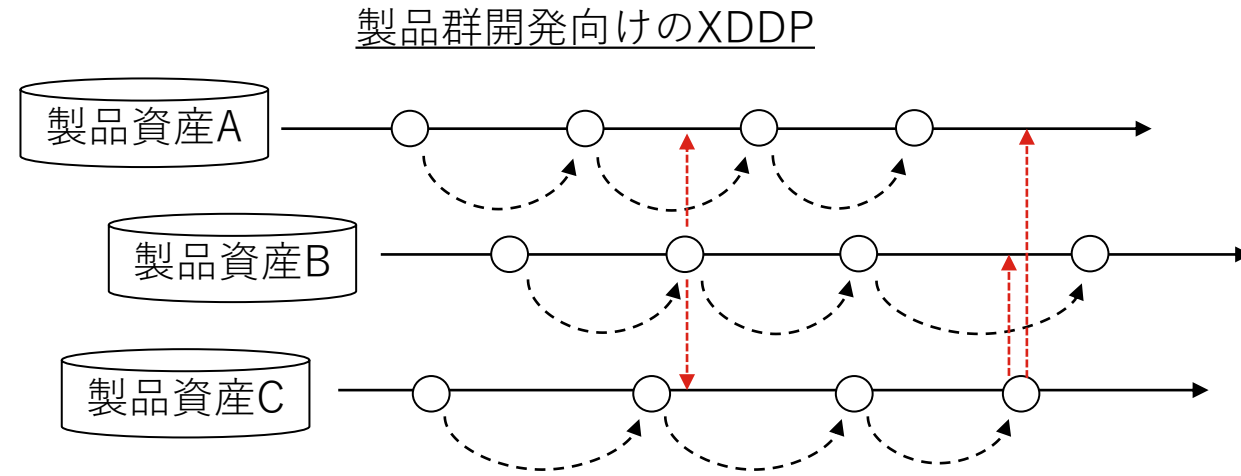


XDDPに+αする観点

- 横展開対応を進めるために、まずはXDDPの適用が「必要」
- 製品群開発向けのXDDPでは、単一製品開発向けのXDDPに、観点を追加する必要がある



時間軸の
差分のみ管理



時間軸 × 空間軸の
差分を管理

製品群開発におけるXDDPの適用に伴う3つの課題



① 製品「間」の差分と理由の可視化

- 開発資産の分岐後に、製品差分の暗黙知が大きくなるようにする

② 差分を共有・抑制する文化の醸成

- 開発資産の分岐後に、製品差分が大きくなるように設計する場を設ける
- 個別最適ではなく全体最適の考え方で横展開対応を進める文化を根付かせる
- 製品差分の可視化をやりやすくするための取り組みを議論する場を設ける

③ 製品全体目線での構成管理

- 製品群全体で統一されたプロセス・ルールで開発を進める
- 時間軸×空間軸で関係性を持つ膨大な開発資産を整理整頓して、検索しやすくする

差分の範囲が大きすぎる

解決

解決

解決

そのために

そのために

解決

情報粒度が違いすぎる

解決

課題に対する3つの取り組み (= SPL開発一歩前の取り組み)



| 課題 | XDDP (+ α) の取り組み | SPL開発の取り組み |
|------------------|---|--------------------|
| ①製品「間」の差分と理由の可視化 | <p style="text-align: center;">製品差分解説書</p> | <p>フィーチャモデル</p> |
| ②差分を共有・抑制する文化の醸成 | <p style="text-align: center;">製品差分共有会</p> | <p>PLコントロール委員会</p> |
| ③製品全体目線での構成管理 | <p style="text-align: center;">製品差分トレースビュー</p> | <p>PLアーキテクチャ</p> |

製品差分解説書



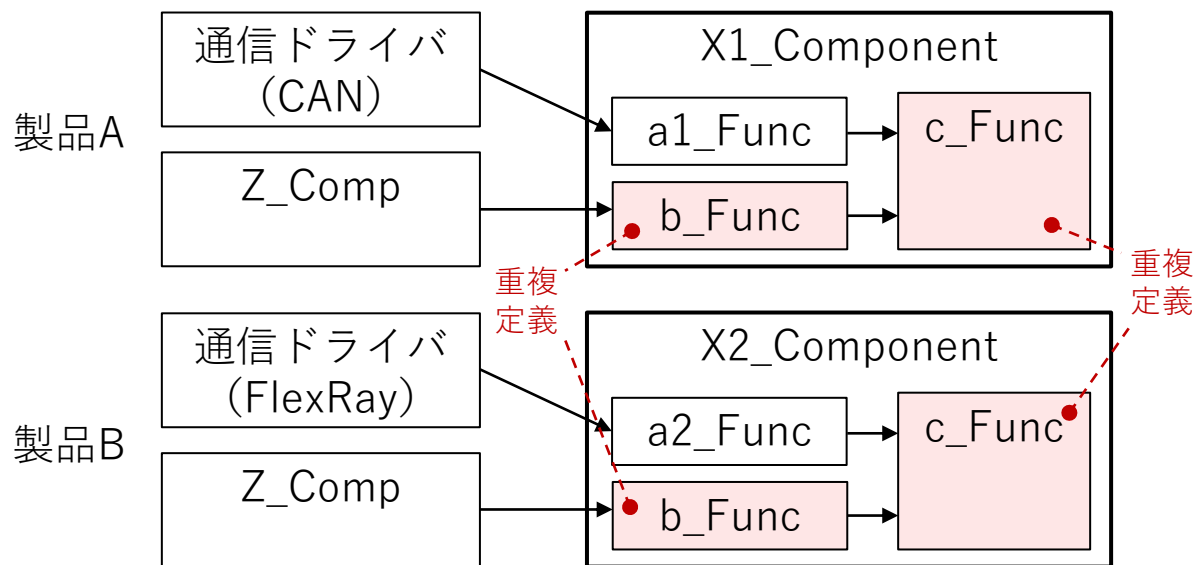
製品差分（概要）と理由の可視化

製品差分（目次）の可視化

製品差分解説書

フィーチャモデル

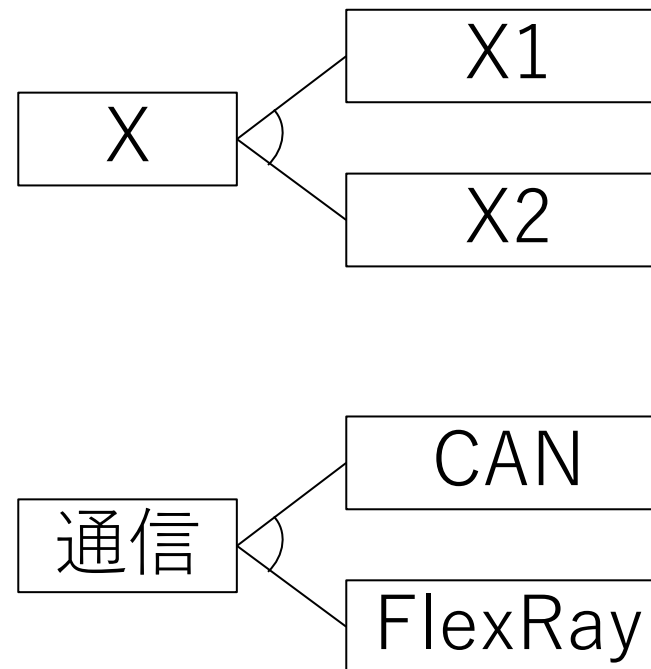
■製品差分の全体像



■なぜこのような作りになっているのか

時期 2019/10/05

- 理由
- ・後発車種Bで顧客から通信形式の変更が要求された際に開発資産が分岐し、個別調整され始めた（差分理由）
 - ・本当は共通部と可変部を綺麗に分離したかったが、Bの開発計画を優先し、重複管理を許容した（構造理由）
 - ・関連チケットURL (<https://www...>)



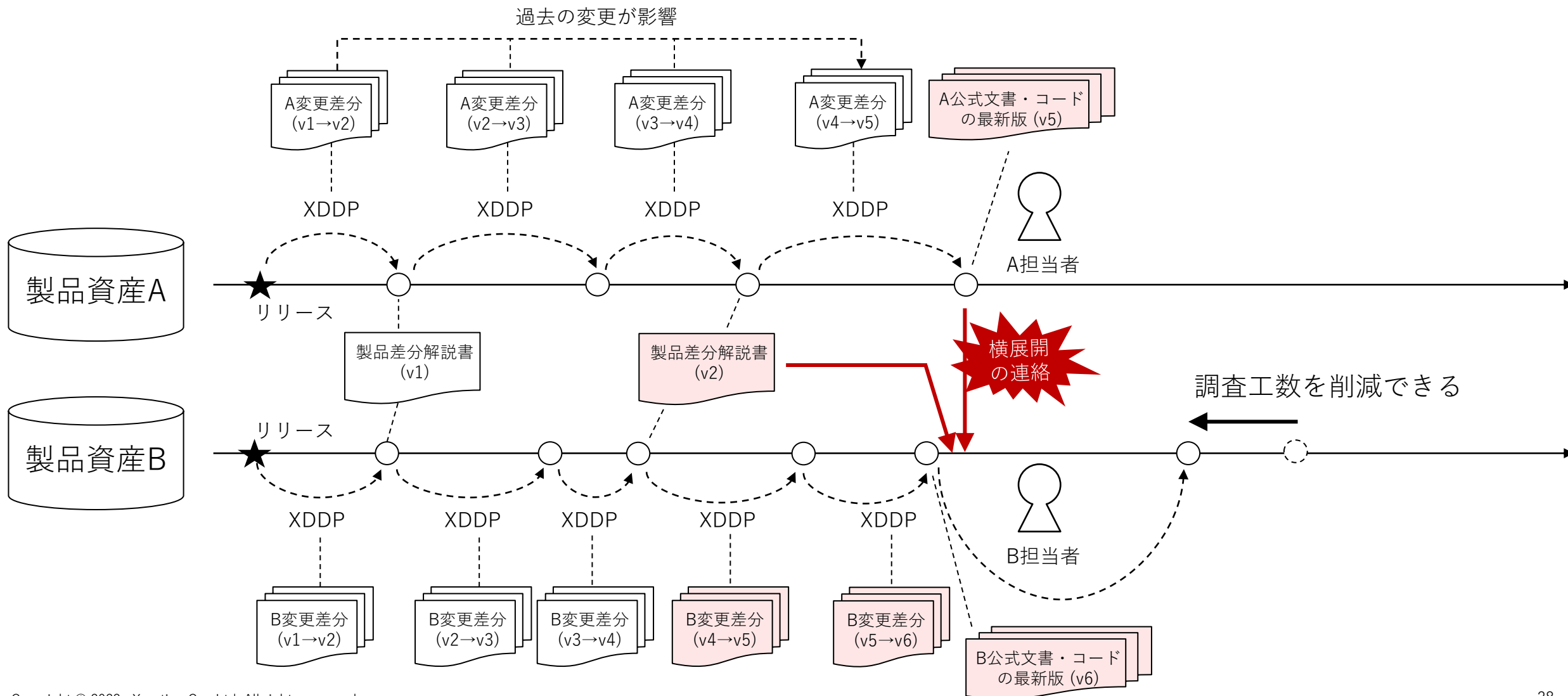
定期的に製品差分を可視化しておく

凡例) ○ 変更

調査対象



- 各製品資産の変更タイミング、または定期的に製品差分を可視化し、記録する



製品差分共有会



製品差分の可視化と横展開の戦略を議論

共通化に向けた戦略を議論

製品差分共有会

■目的

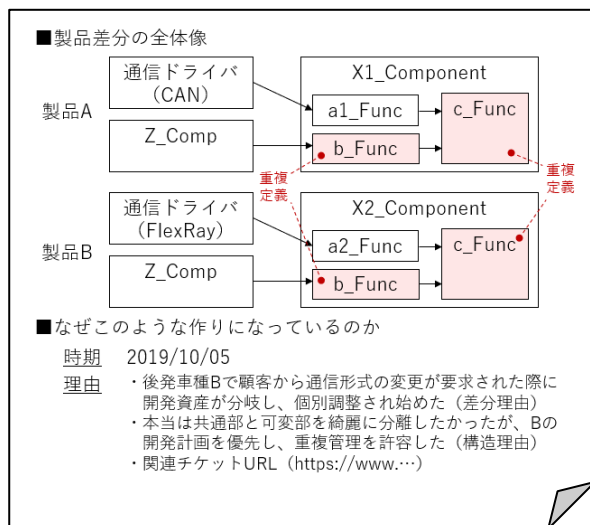
- ・横展開対応の事前対策

■主な活動内容

- ・製品差分と理由の可視化
- ・各製品担当から自身の担当製品の情報を共有
- ・新規開発要件の横展開要否を判断
- ・横展開の進め方の調整
- ・開発プロセス・成果物形式の統一

■アウトプット

- ・議事メモ
(製品差分の共有結果)



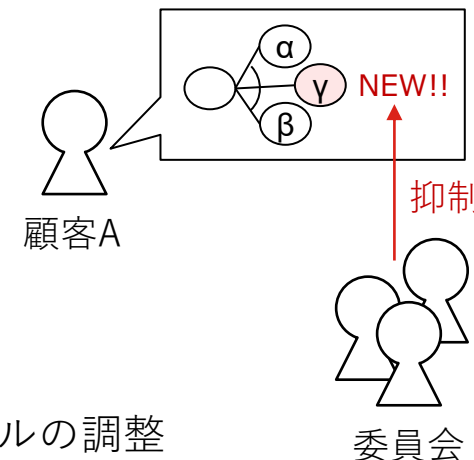
PLコントロール委員会

■目的

- ・SPL開発の推進・維持

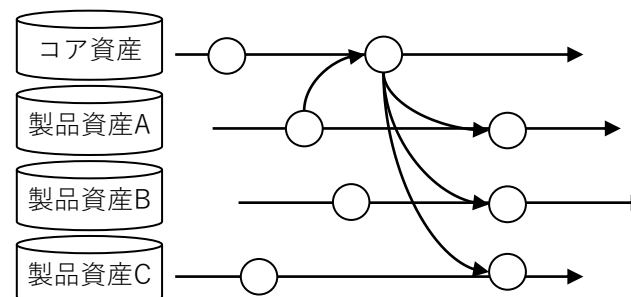
■主な活動内容

- ・製品差分の抑制
- ・共通資産化する範囲の見極め
- ・重複開発の監視
- ・PLアーキテクチャの妥当性評価
- ・再利用資産/製品開発スケジュールの調整

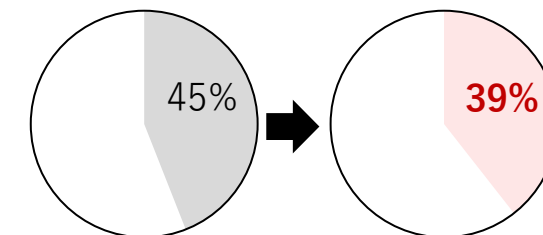


■アウトプット

- ・再利用戦略定義書

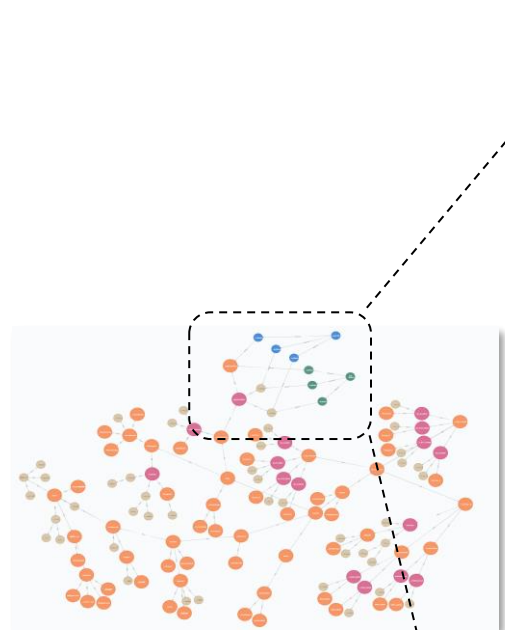


ブランチ戦略 (リリース/マージ計画)

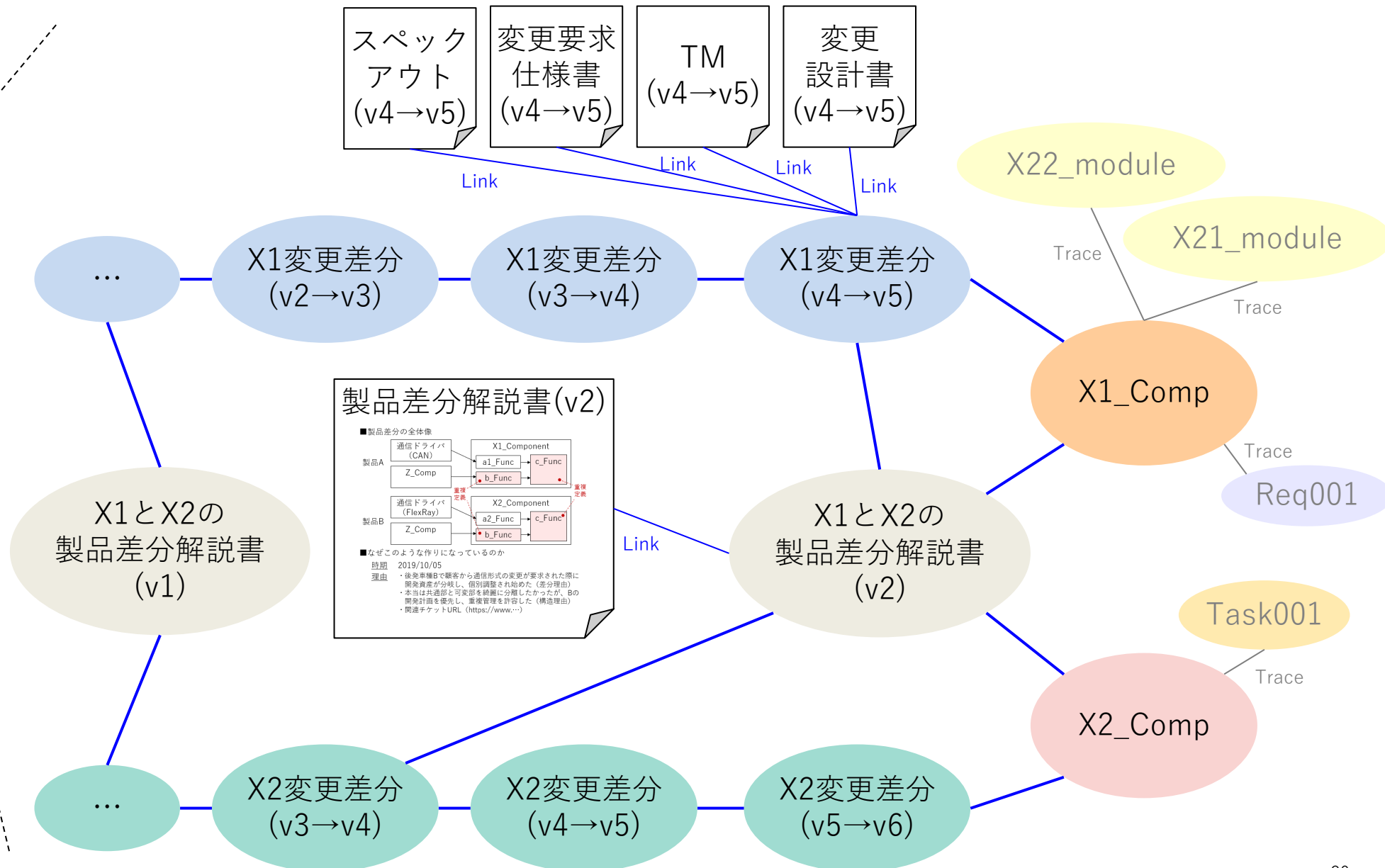


前回からの再利用率の推移

製品差分トレースビュー



開発資産を分析してビューを作成 (HTML)





1 製品群開発現場を悩ませる困りごと

2 XDDPを適用しても発生する問題

3 3つの課題 / 3つの取り組み

4 **取り組みの効果・効能**

5 まとめ



効果（事例1：パワトレ制御）

- メンバーが一丸となりフルセットの改善を実施し、大きな改善効果を得た
 - 共通資産の構築・整理から変更プロセスの抜本的な改善、プロセス支援ツールの構築まで実施
 - ここで記載している内容以外の改善も実施（例：ロールの見直し、教育、…）
 - 今まで **2年かかっていた制御開発が1ヵ月半で回せるようになった**

| | 共通資産 | XDDP | +α | | | 効果 |
|--------|-----------------|------|-------------------------|----------------|-----------------|-------|
| | | | 製品差分 解説書 | 製品差分 共有会 | 製品差分 トレースビュー | |
| Before | 担当者の頭の中に存在する | 適用なし | 適用なし | 適用あり (形式だけ) | 適用なし | 120人月 |
| After | 存在する (重複を許容) | 適用あり | 一部適用あり (フィーチャ管理のみ実施) | 適用あり | 適用あり | 7.5人月 |

改善の流れ

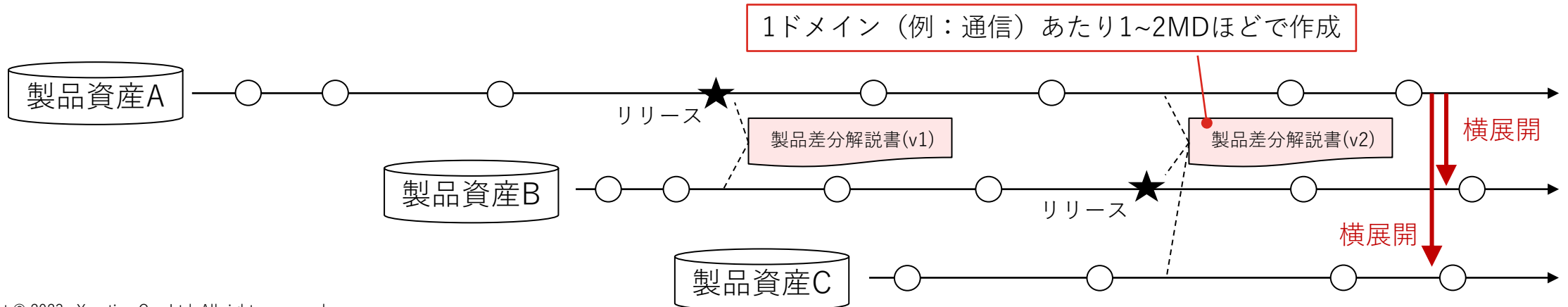




効果（事例2：駆動部品制御）

- メンバー間で製品差分の共有を徹底し、差分調査工数の削減を実現
 - 製品差分解説書をラフに作成（標準化会議の議事メモをそのまま活用）
 - 製品差分共有会を、メジャーリリースイベント毎に開催
 - 効果計測開始時点のサンプル数で、平均して横展開時に1製品あたりの対応工数が**30%程度改善**

| | 共通資産 | XDDP | +α | | | 効果 |
|--------|-----------------|------|---------|---------|-------------|-------------------|
| | | | 製品差分解説書 | 製品差分共有会 | 製品差分トレースビュー | |
| Before | 存在する (重複を許容) | 適用あり | 適用なし | 適用なし | 適用なし | 変更1件あたりの対応工数0.6人日 |
| ↓ | | | | | | |
| After | 存在する (重複を許容) | 適用あり | 適用あり | 適用あり | 適用なし | 変更1件あたりの対応工数0.4人日 |

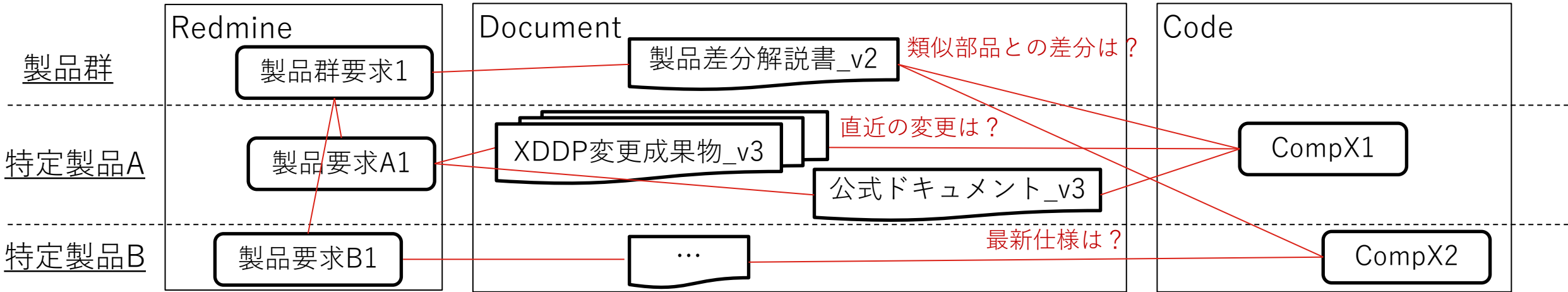




効果（事例3：先進運転制御）

- メンバーが欲しいビューを定義し、設計差分を検索しやすくした
 - 製品差分トレースビューで開発の全体像を可視化
 - 効果計測開始時点のサンプル数で、平均して横展開時に1製品あたりの調査工数が**17%改善**

| | 共通資産 | XDDP | + α | | | 効果 |
|--------|-----------------|------|-------------------------|-------------|-----------------|-------------------|
| | | | 製品差分 解説書 | 製品差分 共有会 | 製品差分 トレースビュー | |
| Before | 存在する (重複を許容) | 適用あり | 一部適用あり (フィーチャ管理のみ実施) | 適用なし | 適用なし | 変更1件あたりの調査工数4時間 |
| After | 存在する (重複を許容) | 適用あり | 適用あり | 適用なし | 適用あり | 変更1件あたりの調査工数3.3時間 |

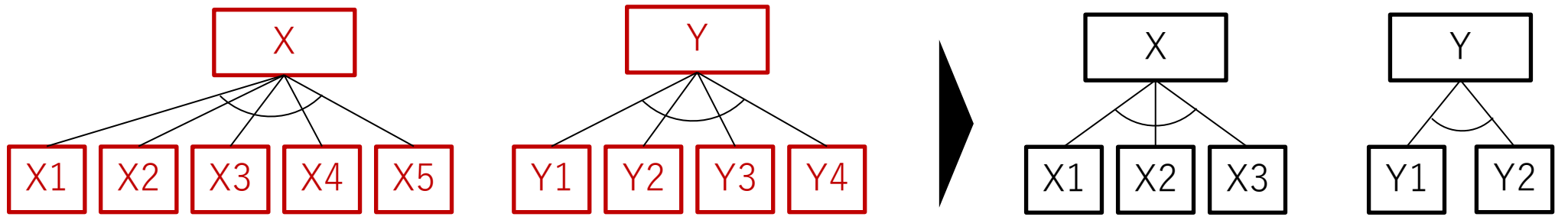




効果（事例4：ボデー制御）

- メンバーが長年悩んでいた制御の複雑さを紐解き、バリエーション数を削減
 - 制御バリエーションが点在しており、冗長定義した制御の変更に苦勞し、不具合の温床となっていた
 - 新規開発のタイミングで、制御構造の見直しを議論する会議体を設け、継続的に共通化を議論
 - 不要な制御バリエーションの削除を検討（PLコントロール委員会としての意味合いが強い）
 - 対応前後で再利用率（共通化できた制御の割合）が**30%程度向上**。バリエーション数は40%削減。

| | 共通資産 | XDDP | + α | | | 効果 |
|--------|--------------------|------|-------------------------|-------------|-----------------|----------|
| | | | 製品差分 解説書 | 製品差分 共有会 | 製品差分 トレースビュー | |
| Before | 存在する (重複を許容) | 適用あり | 一部適用あり (フィーチャ管理のみ実施) | 適用なし | 適用なし | 再利用率 55% |
| ↓ | | | | | | |
| After | 存在する (共通/可変を分離) | 適用あり | 一部適用あり (フィーチャ管理のみ実施) | 適用あり | 適用なし | 再利用率 85% |



今回ご紹介した取り組みが効く現場



- 製品間で開発資産の粒度にムラがある
 - 粒度が異なる製品資産間の差分の調査結果が記録されることで、後の横展開対応が楽になるだけでなく、後の共通資産化もやりやすくなる
- 開発中/リリース後共に、横展開対応の頻度が高い
 - 今までの横展開対応1回あたりにかかっていた膨大な分析・調査時間を削減できる
 - 変更が積み重なるほど、削減効果が増える
- リリースした製品の品質が安定していない恐れがある
 - 業界初の機構を搭載したシステムなど、リリース後に横展開対応が発生する恐れがある場合は、事前対策として今回ご紹介した取り組みをしておくことで、後の緊急/大規模な横展開が発生した場合の炎上を抑えることができる



1 製品群開発現場を悩ませる困りごと

2 XDDPを適用しても発生する問題

3 3つの課題 / 3つの取り組み

4 取り組みの効果・効能

5 **まとめ**



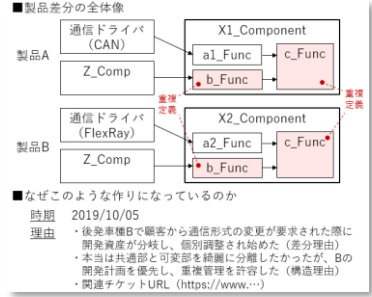
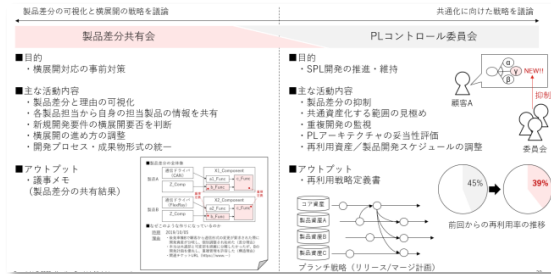
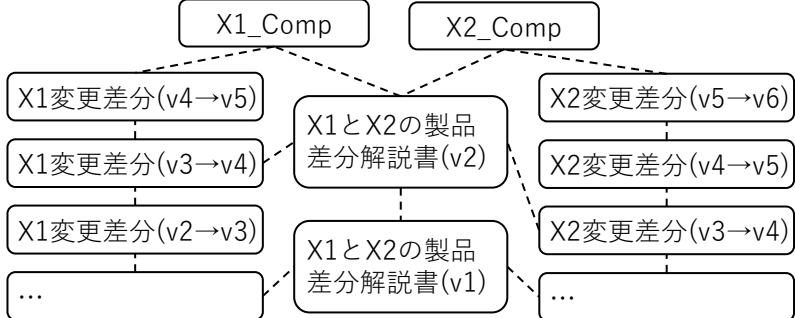
- 製品群開発の現場の多くでは「やむを得ない横展開対応」に苦勞している
 - SPL開発を適用していても、適用していなくても発生する問題

- 大規模・高頻度の横展開対応が発生したら、多くの現場は炎上不可避の状況
 - XDDPを適用していないと最悪
 - XDDPを適用していても、製品間の差分調査に苦勞する

- まずはXDDP適用が先決で、適用後に + α で3つの観点を追加する
 - ① 製品「間」の差分と理由の可視化
 - ② 差分を共有・抑制する文化の醸成
 - ③ 製品全体目線での構成管理

- + α を適用すると、現場によって様々だが、何かしらかの良い効果が出る
 - 例：1件当たりの横展開対応工数が30%削減

今回ご紹介した取り組みをすれば、SPL開発の導入がやりやすくなる

| 課題 | XDDP (+ α) の取り組み | SPL開発の取り組み |
|------------------|---|---|
| ①製品「間」の差分と理由の可視化 | <p style="text-align: center;">製品差分解説書</p>  <p>■製品差分の全体像</p> <p>■なぜこのような作りになっているのか</p> <p>時期 2019/10/05</p> <p>理由</p> <ul style="list-style-type: none"> 後発車種Bで顧客から通信形式の変更が要求された際に開発資産が分岐し、個別調整され始めた(差分理由) 本当は共通部と可変部を綺麗に分けたかったが、Bの開発計画を優先し、重複管理を許容した(構造理由) 関連チケットURL (https://www...) | <p style="text-align: center;">フィーチャモデル</p> <div style="border: 2px solid red; padding: 5px; margin: 10px auto; width: fit-content;"> <p>SPL開発導入をご検討の現場の方は、是非、参考にしてみてください</p> </div> |
| ②差分を共有・抑制する文化の醸成 | <p style="text-align: center;">製品差分共有会</p>  <p>製品差分の可視化と複製率の戦略を議論</p> <p>PLコントロール委員会</p> <p>■目的</p> <ul style="list-style-type: none"> PL開発の推進・維持 <p>■主な活動内容</p> <ul style="list-style-type: none"> SPL開発の抑制 製品差分の共有 共通化された範囲の見極め 重複開発の監視 PLアーキテクチャの妥当性評価 再利用戦略/製品開発スケジュールの調整 <p>■アウトプット</p> <ul style="list-style-type: none"> 再利用戦略定義書 <p>45% → 39%</p> <p>前回の再利用率の推移</p> | <p style="text-align: center;">PLコントロール委員会</p> |
| ③製品全体目線での構成管理 | <p style="text-align: center;">製品差分トレースビュー</p>  <p>X1_Comp X2_Comp</p> <p>X1変更差分(v4→v5) X1とX2の製品差分解説書(v2) X2変更差分(v5→v6)</p> <p>X1変更差分(v3→v4) X1とX2の製品差分解説書(v2) X2変更差分(v4→v5)</p> <p>X1変更差分(v2→v3) X1とX2の製品差分解説書(v1) X2変更差分(v3→v4)</p> <p>...</p> | <p style="text-align: center;">PLアーキテクチャ</p> |



ご清聴

ありがとうございました

Overview



本日の発表内容のOverview

Why

製品群開発で「横展開対応」「重複」が問題の現場は多い
 ↓
 解決手段としてSPL開発が考えられるが、完璧に導入できている現場は少ない

完璧なSPL開発とは？
 ・開発資産の共通部と可変部を綺麗に分離
 ・共通資産を直したら即座に全製品に反映
 ・製品リリース後は共通資産だけ直せば良い

既存資産で品質が担保されているから、手間をかけて分離するうま味が少ない

だから

類似資産が必要になったら、コピーして資産を作り、大部分の重複を許容して個別調整する

共通資産の変更に合わせてリリース後の製品に都度反映なんてできない

だから

製品リリース前は、共通資産と可能な限り同期することで横展開対応を防ぐけど

なぜなら

顧客に関係ないソフトの変更を反映する旨を理解してもらうことが難しいから

リリース後製品は個別管理するので、他製品の変更の横展開対応の必要がある

重複箇所が発生する横展開対応は**力技で解決する**

What

(希望的観測)
 OTAが一般的になり、リリース時期の足並みを揃えられる世界になれば、この問題は解決可能

そもそも上手に横展開対応できてますか？
 (3つの課題)

- ①製品「間」の差分と理由の可視化
- ②差分を共有・抑制する文化の醸成
- ③製品全体目線での構成管理

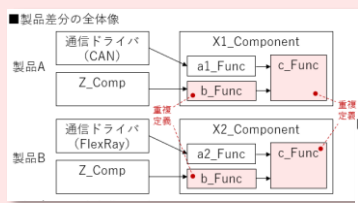
Whatの前提

製品A→Bの横展開でXDDPしても、毎回一から分析して時間がかかる
 ↓
 なぜなら
 製品AとBで要求、設計資料、開発タスクの情報粒度・精度が共に異なるため

これで満足する現場が多いけど…
横展開 300人月

How

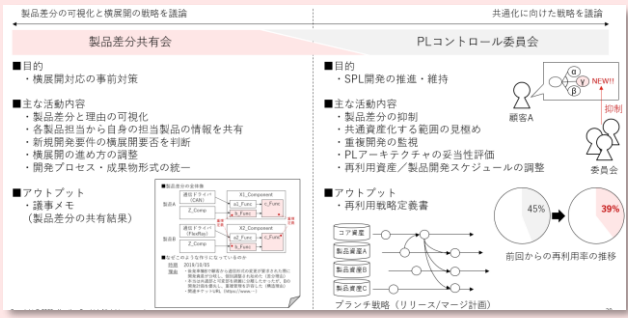
構造・振る舞いの差分



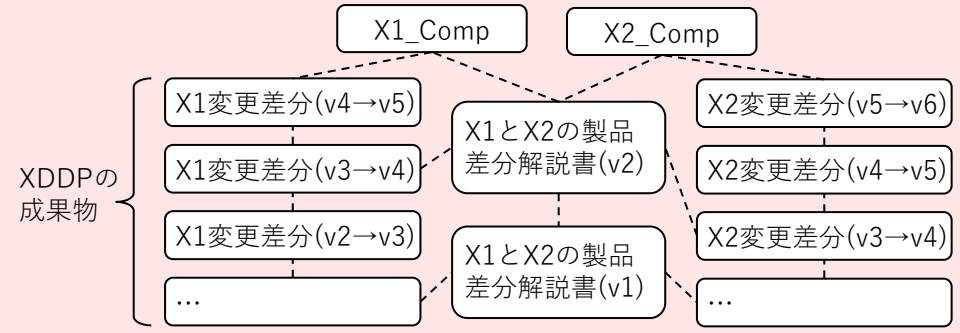
製品差分解説書 (≠フィーチャモデル)

現在の作りになっている理由
 ■なぜこのような作りになっているのか
 時期 2019/10/05
 理由
 ・後発機種Bで顧客から通信形式の変更が要求された際に開発資産が分岐し、個別調整され始めた(差分理由)
 ・本当は共通部と可変部を綺麗に分離したかったが、Bの開発計画を優先し、重複管理を許容した(構造理由)
 ・関連チケットURL (https://www...)

製品差分共有会



製品差分トレースビュー



まとめ

現場を苦しめているのは「やむを得ない横展開対応」 → 今までは横展開対応を力技で乗り切れたが、そろそろ限界の現場が増えている → 上手に横展開するにはXDDPに+αが必要(3つの課題) → 製品差分の共有と抑制を狙った3つの取り組み → 新しい取り組みを始めることによる追加工数も発生するが、トータルで20%程度は横展開対応工数の削減に期待が持てる

よくある質問



Q1. 従来の変更要求仕様書はどう変わるの？

- A. 従来の変更要求仕様に、横展開対応を進めやすくするための要求が、新たに追加される
 - 製品差分共有会で議論した結果と対応した、特定製品に対する横展開対応の要求や、製品差分を共有しやすくしたり、差分拡大を抑えるために必要な要求を新たに定義する。

横展開対応の要求
(製品差分共有会の
議事とリンク)

| | | |
|------|-------|---|
| 変更要求 | CR-01 | 今までは通信モジュールのXXX処理をスケジューラーからコールする際には特別な考慮は必要としていなかったが、これからはXXX処理をコールする前に最低□□msのウェイト処理を入れてほしい |
| | 理由 | α社での不具合対応の横展開。システムを構成しているデバイスの都合上、デバイスの初期処理が完了するまでの間は通信をウェイトさせる必要がある旨をデバイス供給メーカーから指摘あり。 詳細は製品差分共有会_20230405 (外部リンク) 参照 |
| | 説明 | ... |

製品差分を共有しやすくするための要求

| | | |
|------|-------|--|
| 変更要求 | CR-02 | 今まではコードの関数のコメント欄に処理概要が記載されていなかったが、これからは処理概要の記載をしてほしい |
| | 理由 | 後の製品差分解析を楽にするために、α社のコーディングスタイルと統一しておきたい |
| | 説明 | ... |

製品差分拡大を抑制するための要求

| | | |
|------|-------|---|
| 変更要求 | CR-03 | 今までは通信モジュールのYYY処理で、通信データのチェック処理をβ社固有で実装していたが、これからはα社のチェック処理を適用してほしい |
| | 理由 | リリースして品質保証済みのα社の処理に寄せておくことで、製品差分拡大を抑制するため |
| | 説明 | ... |



Q2. 製品差分解説書ってどんな単位で作るの？

- A. 最終的にはコンポーネントくらいの単位が管理上は楽。
 - 製品ラインナップ数が少ない間は、1冊の資料で差分箇所を一覧管理することはできる
 - 製品ラインナップ数が増えてきたら、1冊の資料では差分を管理しきれなくなるので、適切な単位で分割して管理すると良い
 - 場合によっては、コンポーネント単位 & モジュール単位で管理することも考えられるが資料数が増えても管理し切れないので、コンポーネント単位くらいで抑えるのが現実解となる

パターン①
一冊のみ作成

A製品とB製品の差分
差分一覧

| コンポーネント | 差分概要 |
|----------|------|
| XXX Comp | ... |
| YYY Comp | ... |
| ... | ... |

差分詳細 (XXX Comp)

| モジュール | 差分詳細 |
|------------|------|
| XX1 Module | ... |
| XX2 Module | ... |
| ... | ... |

パターン②
コンポーネント単位で作成

XXX Comp差分解説書

A、B、C、Dの差分

YYY Comp差分解説書

A、B、C、Dの差分

パターン③
コンポーネント & モジュール単位で作成

XXX Comp差分解説書

A、B、C、Dの差分

XX1 Module 差分解説書

XX2 Module 差分解説書

XX3 Module 差分解説書

Q3. やること多くない？大変じゃない？



- A. 大変であるのは事実。でもやらないと「もっと大変」になる。
 - 製品差分の分析を、展開元の助力なしで独力で進めるのは困難を極める
 - 実際、横展開対応が必要な現場で、製品間の情報共有の必要性に「NO」を唱える人は少ない
 - 調査した製品差分を残さないと、以降の横展開対応の度に、再び同じ調査をやる羽目になる
 - 製品差分の調査結果は、最悪書きなぐりでも良いから、残すことが重要
 - 製品差分トレースビューを作るのは、難しくない（そこまで工数はかからない）