

# プロダクトライン開発のバリエーション管理について

Managing Variability in System and SW Architecture

pure::variants - Variant Management & Enterprise Architect



はじめに、いくつかの用語を説明します

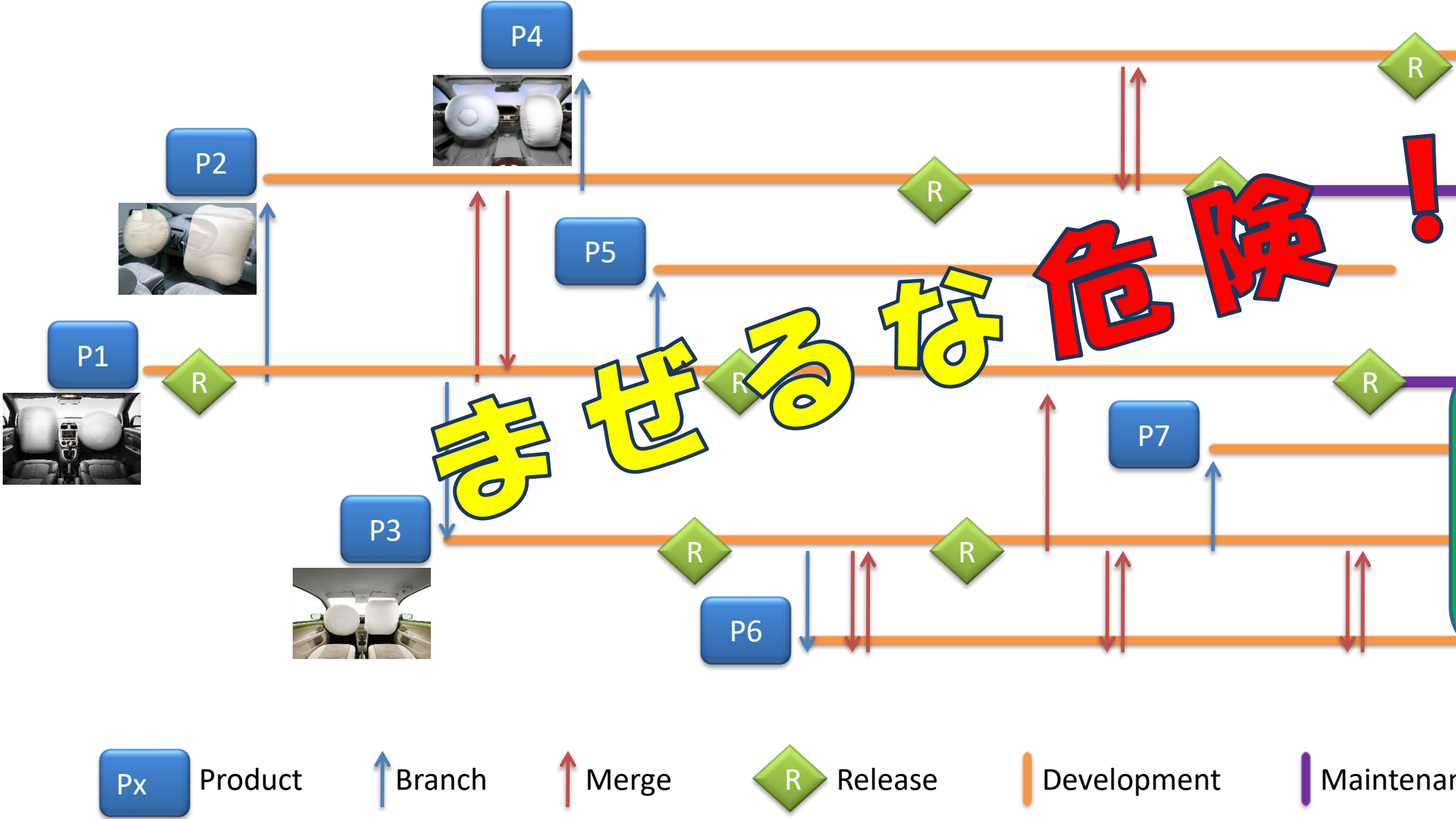


複数の派生製品からなる一連の製品群  
プロダクトライン = 製品系列

プロダクトライン開発では  
各製品のことをバリエーション  
バリエーション間の違いはバリエーション  
バージョンと混同しないこと

なぜ、混同してはいけないか？

# バージョンとバリエーションを混同して管理



まぜるな危険

再利用資産は異なる  
チームごとで変更され、  
メンテナンスや大規模  
な再利用は非常に困難。  
修正を全ての異なるコ  
ピーに反映できない

こうならないように、  
派生開発で継続  
して行われる改善  
や工夫は、上手に  
情報管理して役  
立てたい、



Px Product    ↑ Branch    ↑ Merge    R Release    | Development    | Maintenance    | Integration

そこで登場するのが、  
プロダクトライン開発です





Product Line Engineering (PLE) は、  
再利用資産を運用する技術的な取り組みであり

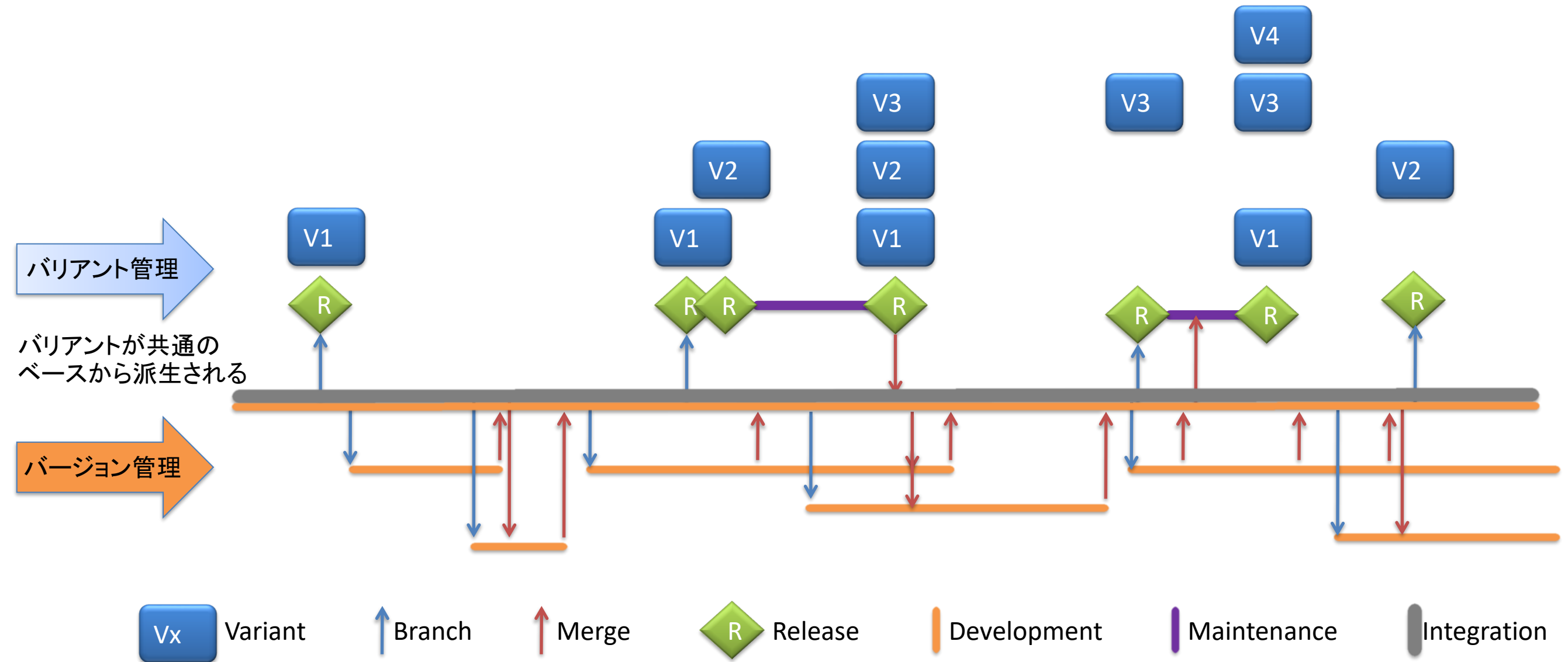
製品（バリエーション）の進化に柔軟に応じることのできる、  
開発プロセスや手法を伴う全体的なアプローチが求められる



そのために必要なことは、



# バージョン管理に適正なバリエント管理ツールを統合



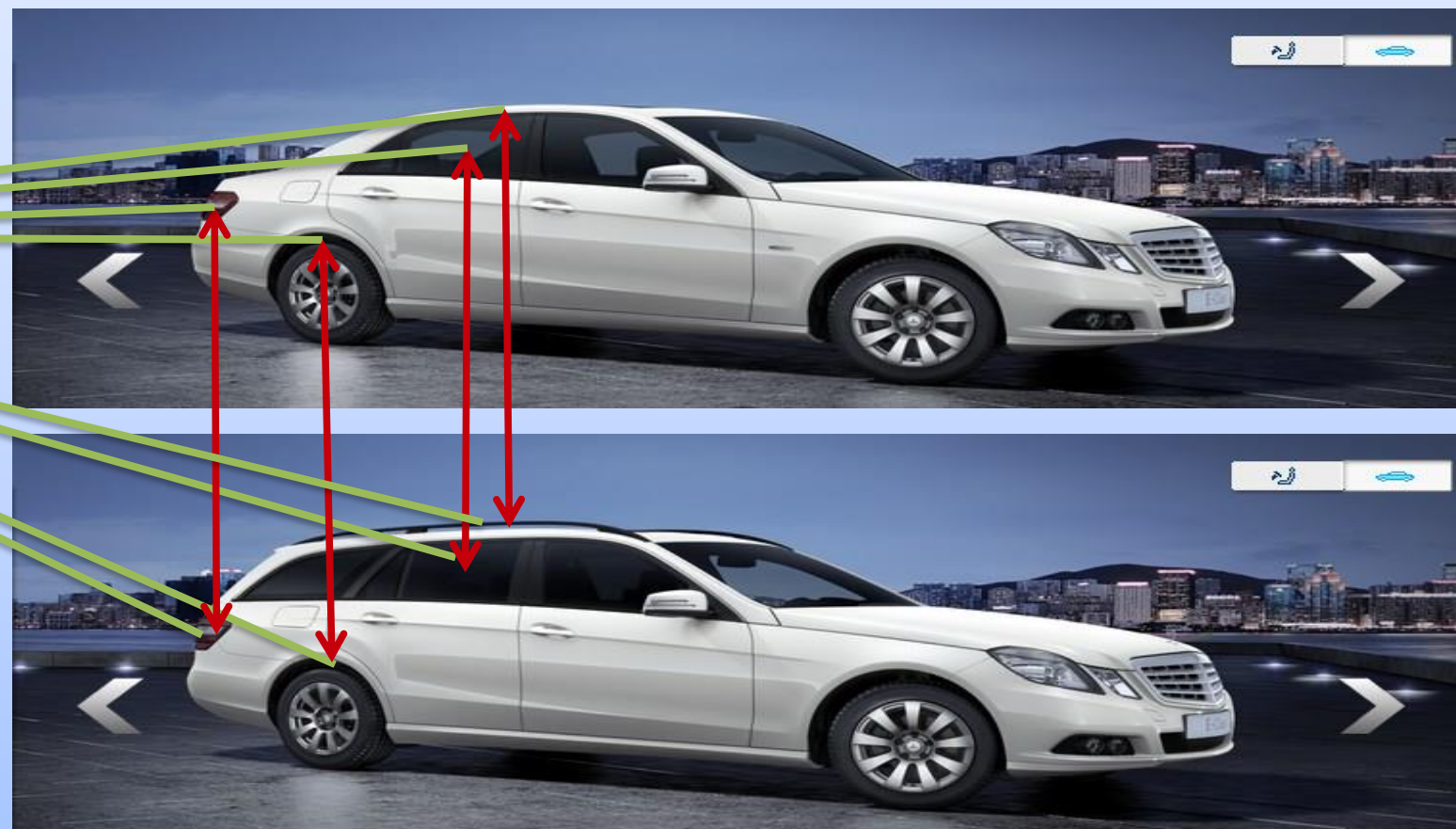
では次に、  
バリエーション管理ツールの説明ですが、  
その前に大切なキーワードである、  
バリエーションポイント

# バリエーションポイント

## 問題空間



## 解決空間



問題空間上のバリエーションポイントは、解決空間上のテクニカルなそれと結びつき、バリエーションの複雑さを軽減してバリエーションの決定項目を削減できる

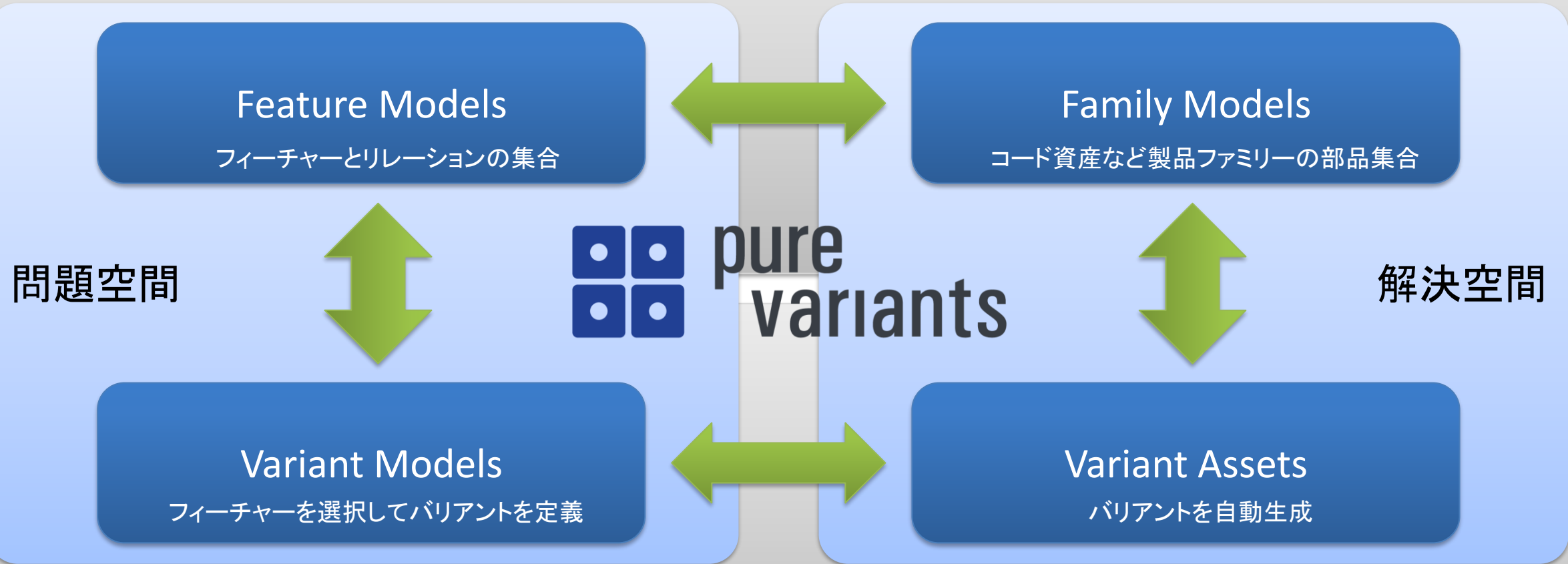
# Variant Management Solution for Systems & Software Engineering





# PLEの戦略＝関心事の分離

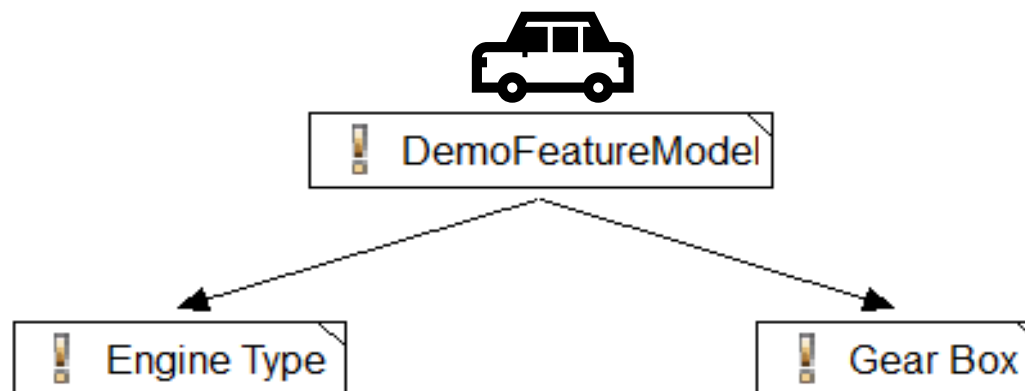
ドメインエンジニアリング



アプリケーションエンジニアリング

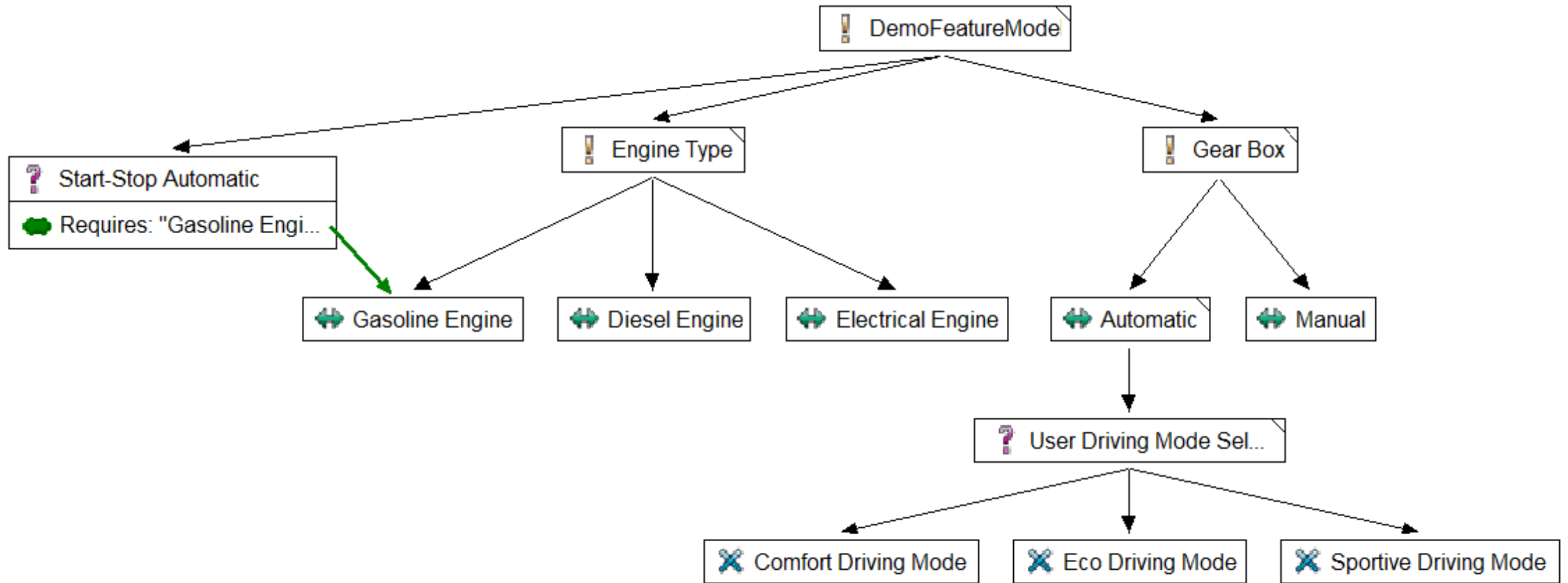
ここでツールの実演を行う前に、  
フィーチャモデルの概要を説明します

# Feature Model



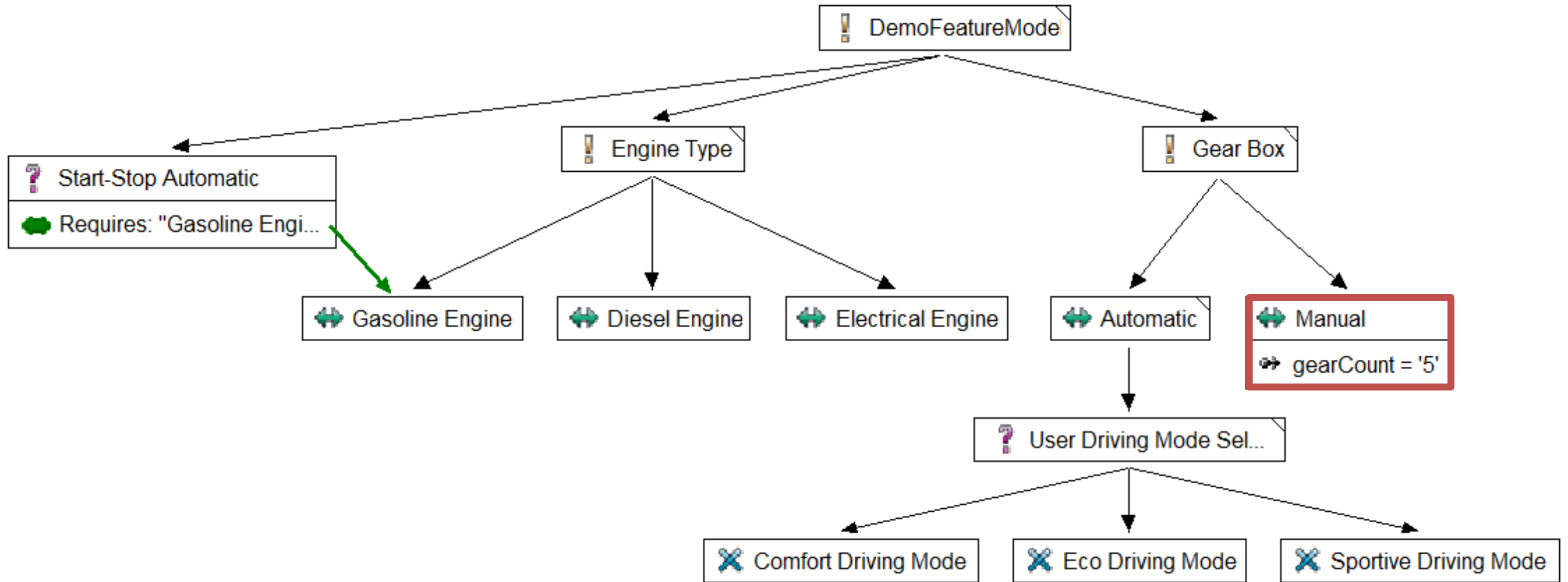
タイプ :  = Mandatory (必須)  = Optional (選択自由)  = Alternative (どれか一つ)  = Or (少なくとも一つ)

# Feature Model - Dependencies



タイプ : ! = Mandatory (必須) ? = Optional (選択自由) ↔ = Alternative (どれか一つ) ✕ = Or (少なくとも一つ)

# Feature Model - attribute

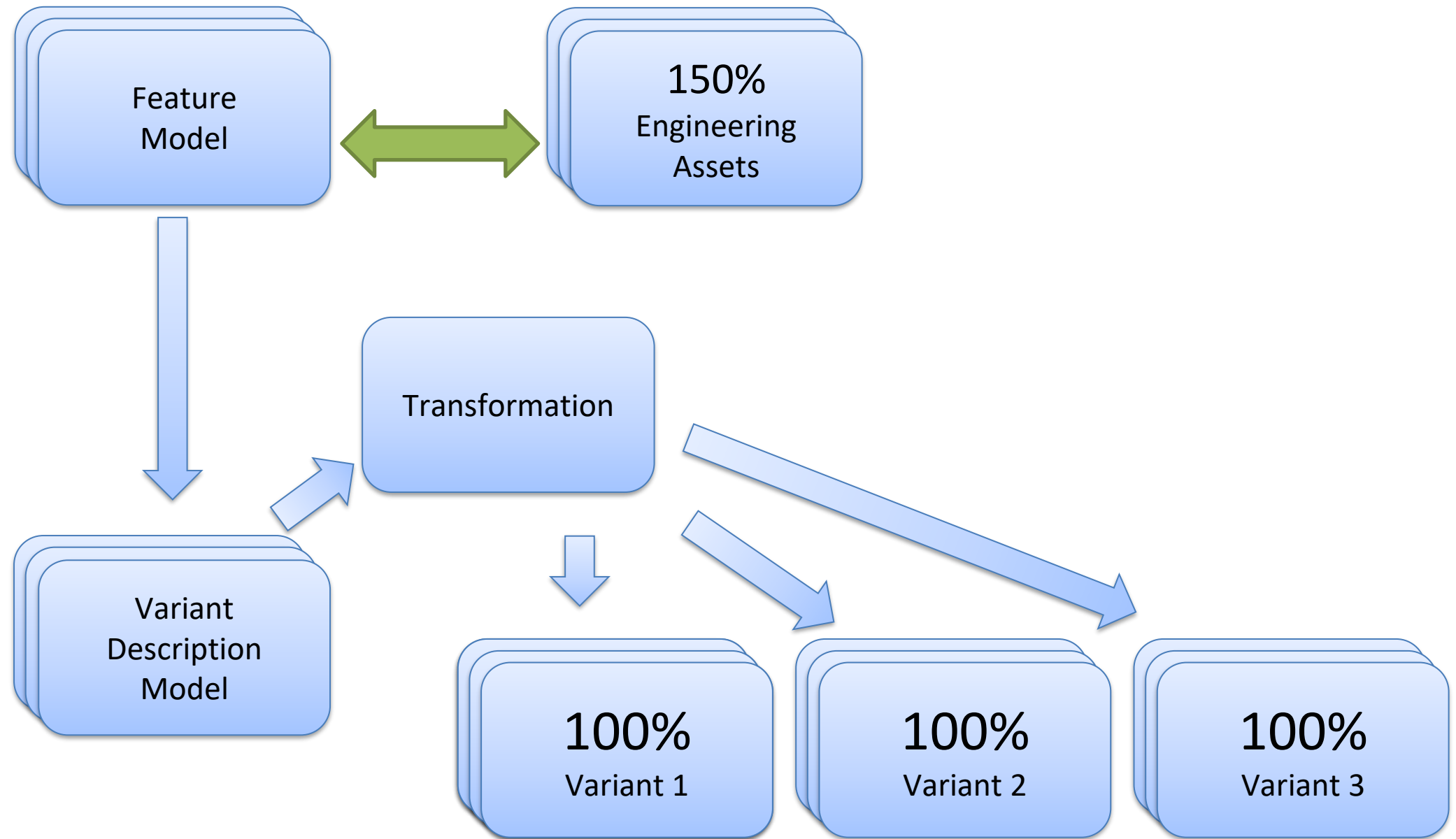


タイプ : ! = Mandatory (必須) ? = Optional (選択自由) ↔ = Alternative (どれか一つ) ✕ = Or (少なくとも一つ)



それではバリエーション管理ツールを、  
**Enterprise Architect** との連携を絡めて実演します

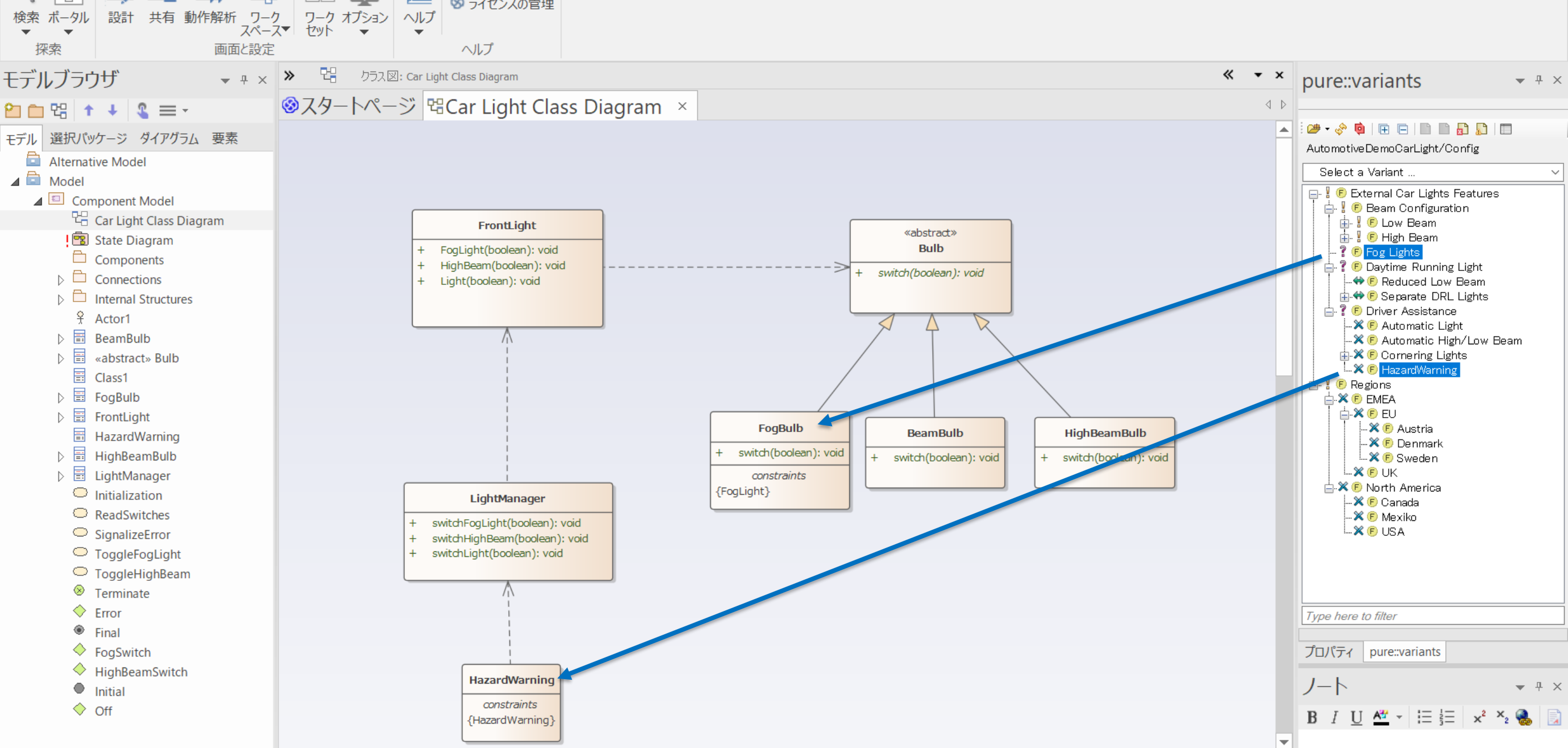
# Feature Model-based Variant Management



# Enterprise Architect 連携

The screenshot displays the Enterprise Architect interface. The main window shows a class diagram with an abstract class **Bulb** and two subclasses, **FogBulb** and **BeamBulb**. The **Bulb** class has a method `switch(boolean): void`. The **FogBulb** class also has a `switch(boolean): void` method and a `constraints {FogLight}` block. The **BeamBulb** class has a `switch(boolean)` method. A '既定値' (Default) dialog box is open, showing the 'p::v Restriction' type selected in the '定義済みの制約の種類' (Defined constraint types) list. The dialog also shows fields for '制約' (Constraint), '説明' (Description), and 'ノート' (Note). The '説明' field contains 'p::v Restriction'. The 'ノート' field is empty. The dialog has buttons for '新規(N)' (New), '保存(S)' (Save), and '削除(D)' (Delete). The '既定値' dialog box is also open, showing the 'p::v Restriction' type selected in the '定義済みの制約の種類' (Defined constraint types) list. The dialog also shows fields for '制約' (Constraint), '説明' (Description), and 'ノート' (Note). The '説明' field contains 'p::v Restriction'. The 'ノート' field is empty. The dialog has buttons for '新規(N)' (New), '保存(S)' (Save), and '削除(D)' (Delete). The '既定値' dialog box is also open, showing the 'p::v Restriction' type selected in the '定義済みの制約の種類' (Defined constraint types) list. The dialog also shows fields for '制約' (Constraint), '説明' (Description), and 'ノート' (Note). The '説明' field contains 'p::v Restriction'. The 'ノート' field is empty. The dialog has buttons for '新規(N)' (New), '保存(S)' (Save), and '削除(D)' (Delete).

クラス図: Car Light Class Diagram: 作成日: 2011/09/26 9:45:56 更新日: 2020/09/14 20:11:15 100% 780 x 1130



フィーチャモデルにプロダクトラインの問題空間のバリエーションを表現して  
解決空間上のあらゆる資産と紐付けられる

検索 ポータル 設計 共有 動作解析 ワークスペース ワーク オプション ヘルプ

探索 画面と設定 ヘルプ

モデルブラウザ

モデル 選択パッケージ ダイアグラム 要素

- Alternative Model
- Model
  - Component Model
    - Car Light Class Diagram
    - State Diagram
    - Components
    - Connections
    - Internal Structures
    - Actor1
    - BeamBulb
    - «abstract» Bulb
    - Class1
    - FogBulb
    - FrontLight
    - HazardWarning
    - HighBeamBulb
    - LightManager
    - Initialization
    - ReadSwitches
    - SignalizeError
    - ToggleFogLight
    - ToggleHighBeam
    - Terminate
    - Error
    - Final
    - FogSwitch
    - HighBeamSwitch
    - Initial
    - Off

クラス図: Car Light Class Diagram

スタートページ Car Light Class Diagram

```
classDiagram
    class FrontLight {
        +FogLight(boolean): void
        +HighBeam(boolean): void
        +Light(boolean): void
    }
    class «abstract» Bulb {
        +switch(boolean): void
    }
    class FogBulb {
        +switch(boolean): void
        constraints {FogLight}
    }
    class BeamBulb {
        +switch(boolean): void
    }
    class HighBeamBulb {
        +switch(boolean): void
    }
    class LightManager {
        +switchFogLight(boolean): void
        +switchHighBeam(boolean): void
        +switchLight(boolean): void
    }
    class HazardWarning {
        constraints {HazardWarning}
    }

    FrontLight ..> Bulb
    FrontLight ..> LightManager
    LightManager ..> FrontLight
    LightManager ..> HazardWarning
    Bulb <|-- FogBulb
    Bulb <|-- BeamBulb
    Bulb <|-- HighBeamBulb
```

pure::variants

AutomotiveDemoCarLight/Config

HighLight\_US.vdm

- External Car Lights Features
  - Beam Configuration
    - Low Beam
    - High Beam
    - Fog Lights
  - Daytime Running Light
  - Reduced Low Beam
  - Separate DRL Lights
  - Driver Assistance
  - Automatic Light
  - Automatic High/Low Beam
  - Cornering Lights
  - HazardWarning
- Regions
  - EMEA
    - Austria
    - Denmark
    - Sweden
    - UK
  - North America
    - Canada
    - Mexiko
    - USA

Type here to filter

プロパティ pure::variants

ノート

クラス図: Car Light Class Diagram: 作成日: 2011/09/26 9:45:56 更新日: 2021/09/19 2:03:38 100% x 1130

モデル内アドイン

バリエーションモデルのプレビュー機能  
150%モデル内のバリエーションを取捨選択

# フィーチャ間の依存・排他関係

The image displays a feature management interface with two feature trees and a relation editor.

**Functional\_Features.xfm**

- External Car Lights Features
  - Beam Configuration
  - Fog Lights
  - Daytime Running Light
- Driver Assistance
  - Automatic Light
    - Automatic High/Low Beam
  - Cornering Lights
    - Adaptive Forward Lighting
    - Static Cornering Lights
      - Requires: "Fog Lights"
  - HazardWarning

**Regions.xfm**

- Regions
  - EMEA
    - EU
      - Austria
      - Denmark
      - Sweden
        - Requires: "HazardWarning"
      - UK
    - North America
      - Canada
      - Mexiko
      - USA

**Edit 'Sweden'**

Edit Relations...

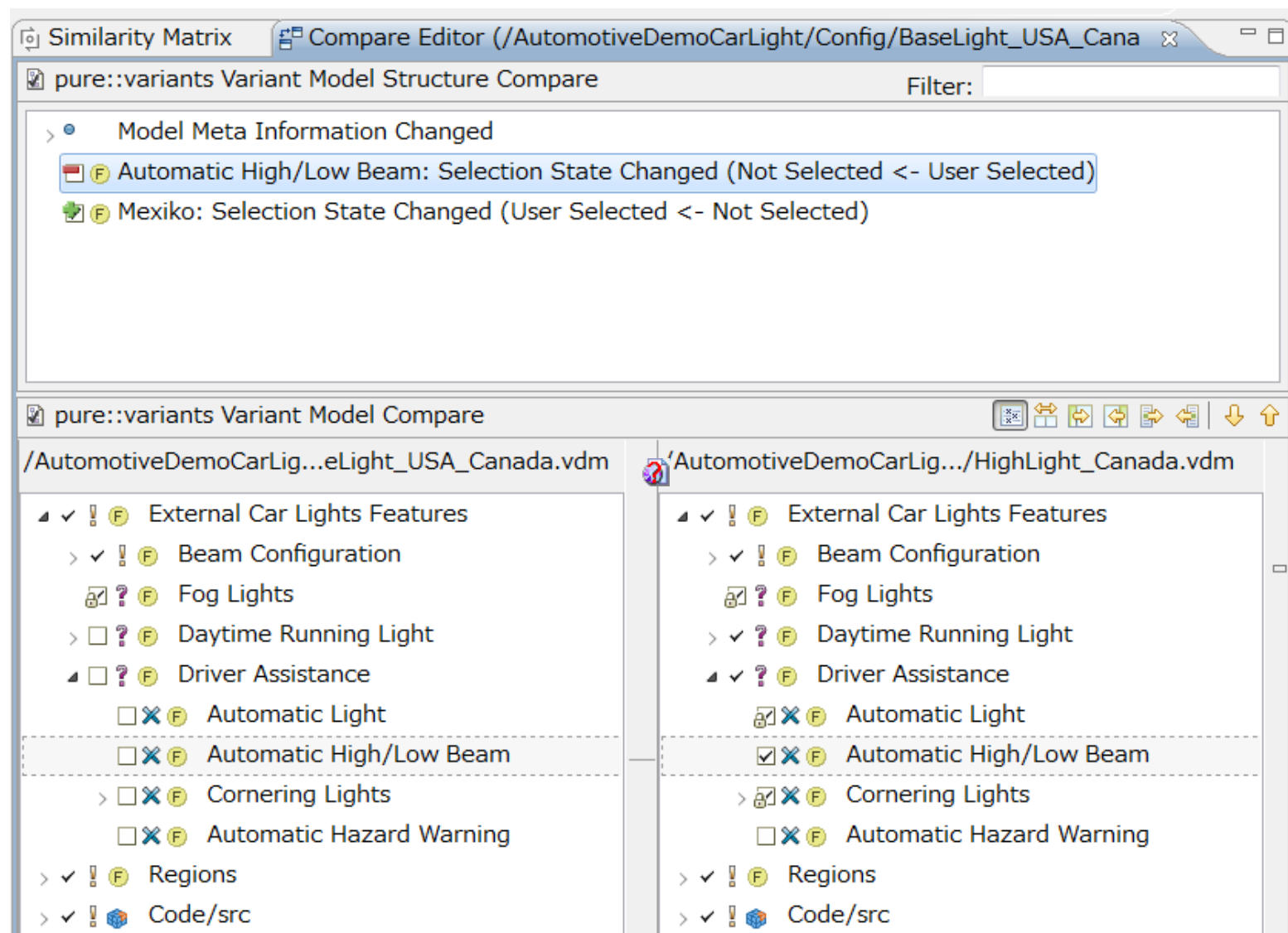
| General              | Relations | Attributes | Restrictions | Constraints |
|----------------------|-----------|------------|--------------|-------------|
| <b>Type</b>          |           |            |              |             |
| ps:requires          |           |            |              |             |
| ps:defaultProvider   |           |            |              |             |
| ps:discourages       |           |            |              |             |
| ps:discouragesAny    |           |            |              |             |
| ps:equalsAll         |           |            |              |             |
| ps:equalsAny         |           |            |              |             |
| ps:exclusiveProvider |           |            |              |             |
| ps:expansionProvider |           |            |              |             |
| ps:influences        |           |            |              |             |
| ps:provides          |           |            |              |             |
| ps:recommendedFor    |           |            |              |             |
| ps:recommendedForAll |           |            |              |             |
| ps:recommends        |           |            |              |             |
| ps:recommendsAll     |           |            |              |             |
| ps:requestsProvider  |           |            |              |             |
| ps:requiredFor       |           |            |              |             |
| ps:requiredForAll    |           |            |              |             |
| ps:requires          |           |            |              |             |



| Model Elements |                     | Level   | BaseLight                | BaseLight_EMEA | BaseLight_USA_C...       | Demo                     | HighLight                | HighLight_Canada         | HighLight_EMEA           | HighLight_US             |
|----------------|---------------------|---------|--------------------------|----------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| [-]            | CarLightFeatures    |         |                          |                |                          |                          |                          |                          |                          |                          |
| [-]            | Car Lights Features |         | ✓                        | ✓              | ✓                        | ✓                        | ✓                        | ✓                        | ✓                        | ✓                        |
| [-]            | Regions             | 1       | ✓                        | ✓              | ✓                        | ✓                        | ✓                        | ✓                        | ✓                        | ✓                        |
| [-]            | EMEA                | 1.1     | <input type="checkbox"/> | ✓              | ✗                        | <input type="checkbox"/> | <input type="checkbox"/> | ✗                        | ✓                        | ✗                        |
| [-]            | EU                  | 1.1.1   | <input type="checkbox"/> | ✓              | ✗                        | <input type="checkbox"/> | <input type="checkbox"/> | ✗                        | ✓                        | ✗                        |
|                | Austria             | 1.1.1.1 | <input type="checkbox"/> | ✓              | ✗                        | <input type="checkbox"/> | <input type="checkbox"/> | ✗                        | <input type="checkbox"/> | ✗                        |
|                | Denmark             | 1.1.1.2 | <input type="checkbox"/> | ✓              | ✗                        | <input type="checkbox"/> | <input type="checkbox"/> | ✗                        | ✓                        | ✗                        |
|                | UK                  | 1.1.1.3 | <input type="checkbox"/> | ✓              | ✗                        | <input type="checkbox"/> | <input type="checkbox"/> | ✗                        | ✓                        | ✗                        |
| [-]            | North America       | 1.2     | <input type="checkbox"/> | ✗              | ✓                        | <input type="checkbox"/> | <input type="checkbox"/> | ✓                        | ✗                        | ✓                        |
|                | Canada              | 1.2.1   | <input type="checkbox"/> | ✗              | ✓                        | <input type="checkbox"/> | <input type="checkbox"/> | ✓                        | ✗                        | <input type="checkbox"/> |
|                | Mexiko              | 1.2.2   | <input type="checkbox"/> | ✗              | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | ✗                        | <input type="checkbox"/> |
|                | USA                 | 1.2.3   | <input type="checkbox"/> | ✗              | ✓                        | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | ✗                        | ✓                        |
| [-]            | Beam Configuration  | 2       | ✓                        | ✓              | ✓                        | ✓                        | ✓                        | ✓                        | ✓                        | ✓                        |
| [-]            | Low Beam            | 2.1     | ✓                        | ✓              | ✓                        | ✓                        | ✓                        | ✓                        | ✓                        | ✓                        |
|                | Xenon               | 2.1.1   | <input type="checkbox"/> | ✗              | ✗                        | <input type="checkbox"/> | ✓                        | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
|                | Halogen             | 2.1.2   | <input type="checkbox"/> | ✗              | ✗                        | <input type="checkbox"/> | ✓                        | <input type="checkbox"/> | ✗                        | ✗                        |
| [-]            | High Beam           | 2.2     | ✓                        | ✓              | ✓                        | ✓                        | ✓                        | ✓                        | ✓                        | ✓                        |
|                | Matrix              | 2.2.1   | <input type="checkbox"/> | ✗              | ✗                        | <input type="checkbox"/> | ✓                        | <input type="checkbox"/> | ✗                        | ✗                        |

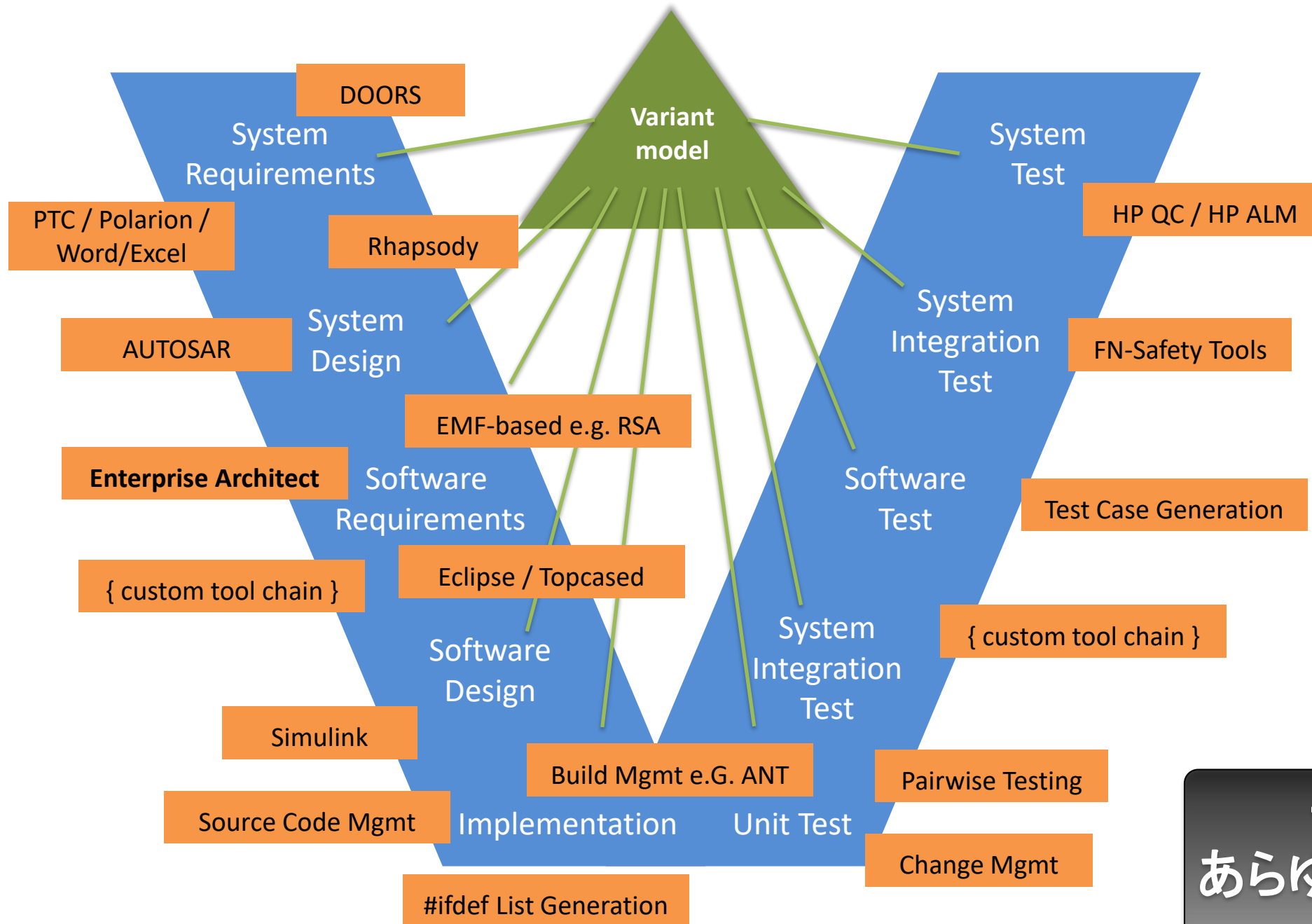
バリエーションごとで搭載する機能を比較  
フィルターやソートでバリエーション間の違いや同一性を分析できる

# バリエーション間の比較や旧バージョンとの比較もできる

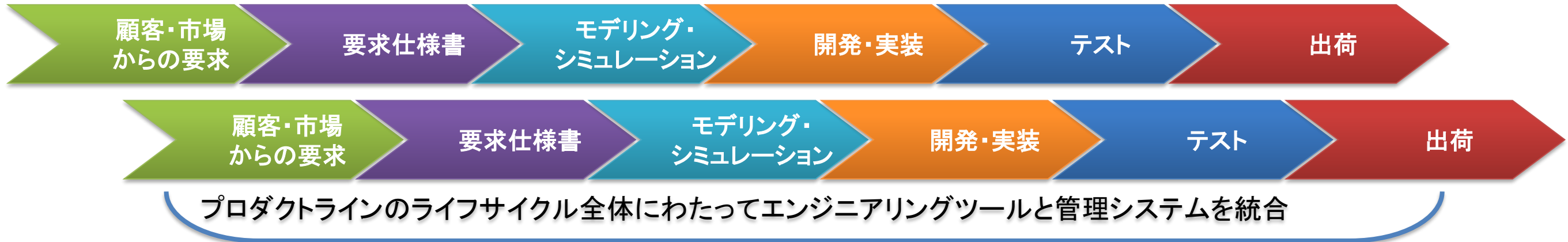


<https://www.fuji-setsu.co.jp/products/purevariants/tutorials.html#CompareMarge>

# pure::variants Enterprise Integrations



フィーチャモデルと  
あらゆる資産を紐付けられる



DOORS 9

DOORS Next

Polarion Requirements

Rhapsody

RTC

Rational Quality Manager

C/C++/Java

MS Word, Excel

AUTOSAR

EMF

Enterprise Architect

Simulink

MagicDraw

...

Zuken

medini analyze

14:25～15:15 (50分)

## コマツがたどりついたシステムの作り方 ― その2 ツール活用編 ―

コマツ

開発本部 システム開発センタ メカトロ制御第一グループ

技師 上義樹様

コマツでは2013年度から設計意図を残すことをとりかかりとしてMBSEを導入してきた。

9月6日にオージス総研様のセミナーにて、MBSEとプロダクトライン開発を組合せる取り組みについて紹介する。本発表では、その続きとして、**ツールをどのように使ってソフトウェアを開発しているかを**中心に報告する。

### 第3回 Enterprise Architect 事例紹介セミナー

現在の状態 **お申し込み受付中です。**

日付 **2019年10月23日(水)**

時間 **13:30～17:20 (13:10 受付開始)**

会場 **富士ソフト アキバプラザ (JR・地下鉄秋葉原駅近く)**

「[gihyo.jp](https://gihyo.jp)」に記事公開 => <https://gihyo.jp/news/report/2019/11/1112>

MBSE, プロダクトライン開発を実践し要求を正しく満たすソフトウェアをスピード感をもって実装できる体制を整備

# Customer Example



# ダンフォス社: 周波数コンバータ業界2位



異なる市場向けの製品群:

- 小型～大規模システムまで様々なバリエーション
- 工業用、水のポンプ、ビルの空調、特定用途

主な動機:

- 類似製品の増加
- ソフトウェア資産のより良い管理を模索
- 資産を再利用して品質を向上させる

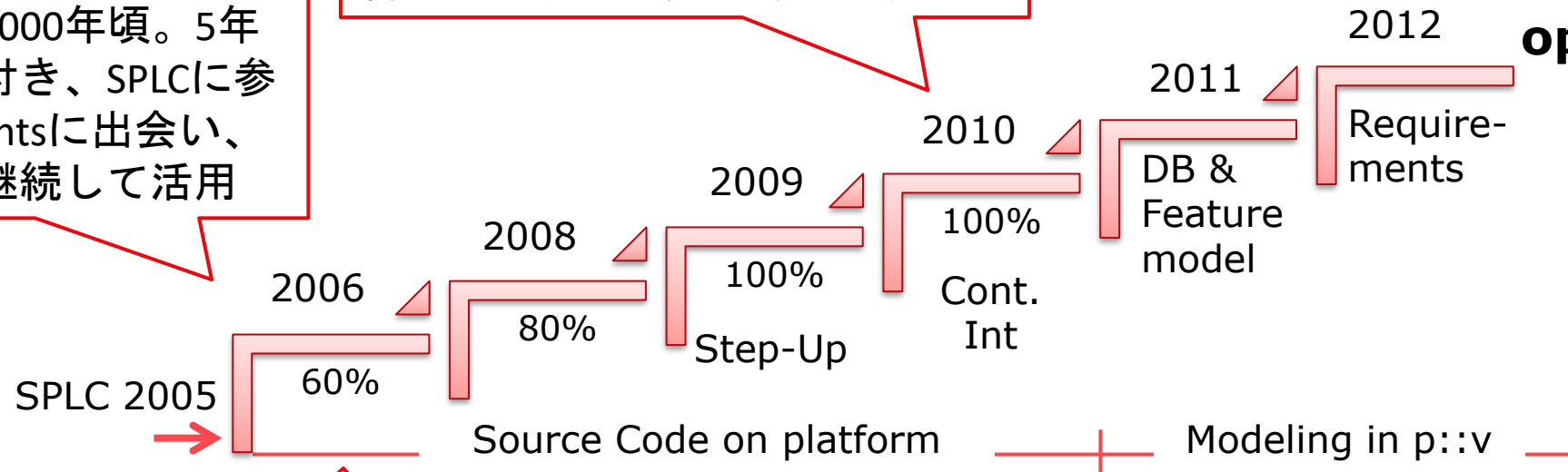


# The Journey

2006年にPLEを開始。プラットフォームは2000年頃。5年たって課題に気づき、SPLCに参加してpure::variantsに出会い、現在に至るまで継続して活用

100%までの期間は長く見えるが、全ての既存製品を継続開発しながら、PLEのアプローチにマイグレートする必要がある、また各製品は、従来通り6カ月ごとの更新に対処する必要があった。

その後範囲も拡大した。製品ごとのパラメーターデータベースを1つの共通パラメーターファミリーデータベースにして、最後に要件も対象に。通常は要件から始めることが多いが、当初要件管理のマネージャがPLEに賛同しなかった。



当初はSWだけに。再利用の範囲は60%（2カ月で達成、残りの40%は各プロジェクトに固有）、80%（2年後）、100%（また1年後）と段階的に広げた。

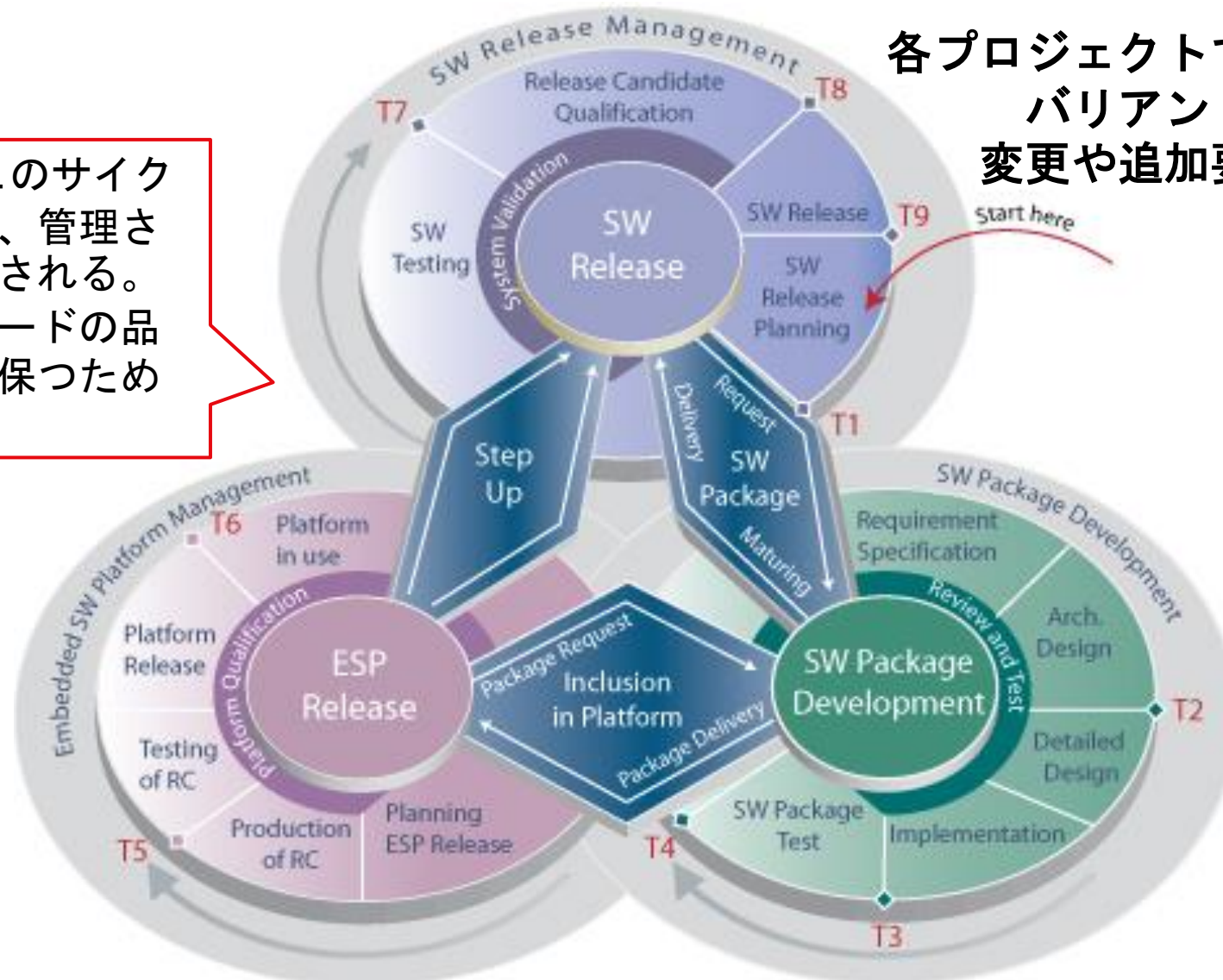
2つの製品を選び、これらのマージから始めた。これが上手くいくことを見届けてから、残りの10製品とのギャップを調査。マージされたものと同等のコードが有った場合は、それをマージされた方のコードに置き換えた。これで最初の60%の達成は比較的容易であった。複雑なものは手を付けないようにした。ただそのあとは複雑なコードにも対処したので、時間も費やした。一度に100%を目指さないことが良かったといえる。



# ダンフォス社 PLEベースの開発プロセス

全てのSWがこのサイクルで開発され、管理され、リリースされる。全ての製品コードの品質と一貫性を保つために行われる。

各プロジェクトで利用して  
バリエーションをテスト  
変更や追加要求を出す



The 3 Circles

SW Release cycle  
SW Package cycle  
ESP release cycle

The Toll Gates

T1 – Release Planning is in place

T2 – Arch design acceptable

T3 – Implementation acceptable

T4 – Test & platform elements acceptable

T5 – ESP Release Candidate available

T6 – ESP Official Release available

T7 – Planned SW Testing completed

T8 – Final Release Candidate Approved

T9 – Official SW Release is completed

ESP = Embedded Software Platform  
ここで資産をPLEに管理  
矛盾や一貫性など品質面も評価

PLEの資産を開発  
新たなパッケージなどもここで開発

# ダンフォス社:プロダクトライン開発の成果

- プロダクトライン開発は、  
業界6位から2位に押し上げた要因の一つ
- プロダクトライン開発は2004年に開始され、  
当初期待された寿命は10年でしたが、  
2016年現在、さらに10年間使用できることを確認
- 製品数は10倍に増えたが(20製品と60のオプション)  
担当者数は5倍増で済んだ

各プロジェクトのソースコードは約100万行。PLE前は10プロジェクト有ったので、1000万行の管理が必要ということになる。PLE導入後は20プロジェクトに増えたが総計400万行である。少ないコードで多くの製品を生み出している。結果的に品質向上にも貢献。同様に要件も50%くらい減っている。このように以前との比較でPLEの効果を測ることも出来る。

顧客固有のシステムを受注するためのプロトタイプを作成は、PLE導入以前は構成要素の変更が容易ではなく2～3ヶ月必要であったが、PLE導入後は1～2週間でできるようになった。

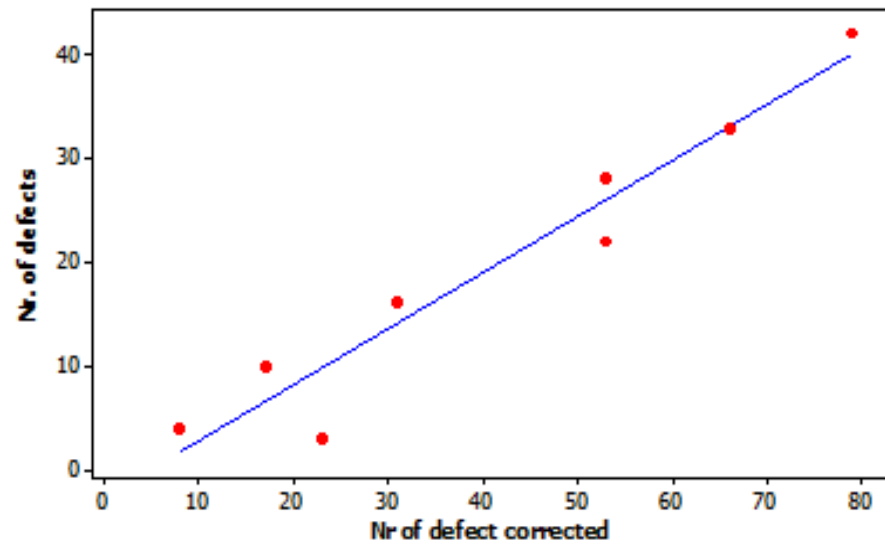
論文: Ten Years of Product Line Engineering at Danfoss: Lessons Learned and Way Ahead

[https://www.researchgate.net/publication/308816044\\_Ten\\_years\\_of\\_product\\_line\\_engineering\\_at\\_Danfoss\\_lessons\\_learned\\_and\\_way\\_ahead](https://www.researchgate.net/publication/308816044_Ten_years_of_product_line_engineering_at_Danfoss_lessons_learned_and_way_ahead)

# PLEのビフォーアフター/欠陥相関

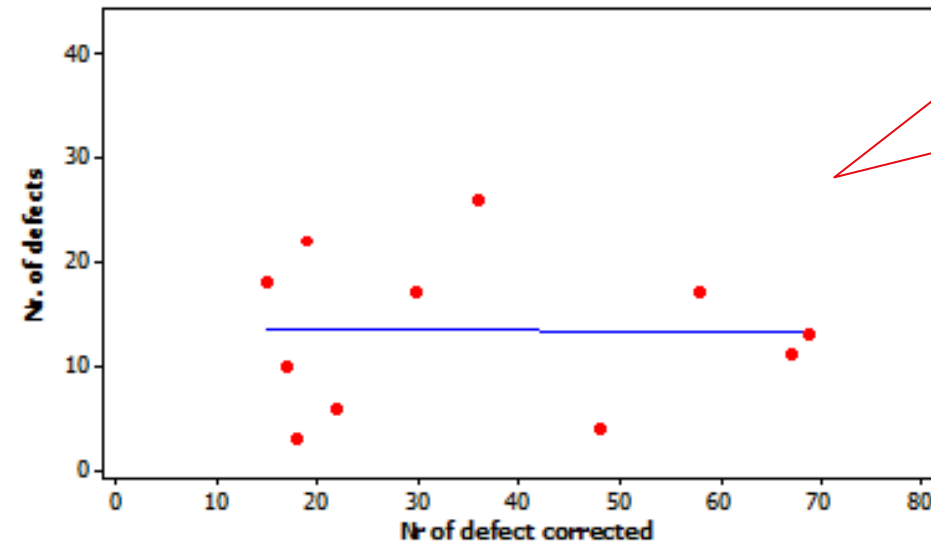
新たに検出されたバグの数

Before PLE Process



バグ修正の数

After PLE Process



バグ修正の数

PLE後はバグの修正によって  
新たなバグが出ない傾向であることもわかった！

SixSigma Study



## PLE成功事例における共通性は？：

- ・初期段階で多くの時間はかかっていない。  
製品にある程度の成熟があり、またプロジェクトは50名以上。
- ・得られる効果と負担は共通。失敗もある。それを受け入れられるか？
- ・他のチームと上手くやれる？ 変えることに伴う痛みに耐えられる？
- ・「現状に満足？」そもそも問題があったので取り組みを考えたはず。  
エンジニアだけでなくマネージャを含めた組織内で皆が共通の認識を持つことが重要。ただいつでも皆の賛同が得られるというわけでは無い。  
組織が十分な柔軟性を持つことができることが大きな成功要素。  
また作業を継続する中で学びもあるはずで、それを生かして改善を加えていけることも大切。





pure::variants について

<https://www.fuji-setsu.co.jp/products/purevariants/>

Dr.Danilo Beuche の実践的なプロダクトライン開発について

[https://www.fuji-setsu.co.jp/products/purevariants/Danilo\\_Blog.html#PLE](https://www.fuji-setsu.co.jp/products/purevariants/Danilo_Blog.html#PLE)

富士設備工業(株) 電子機器事業部

<https://www.fuji-setsu.co.jp>

