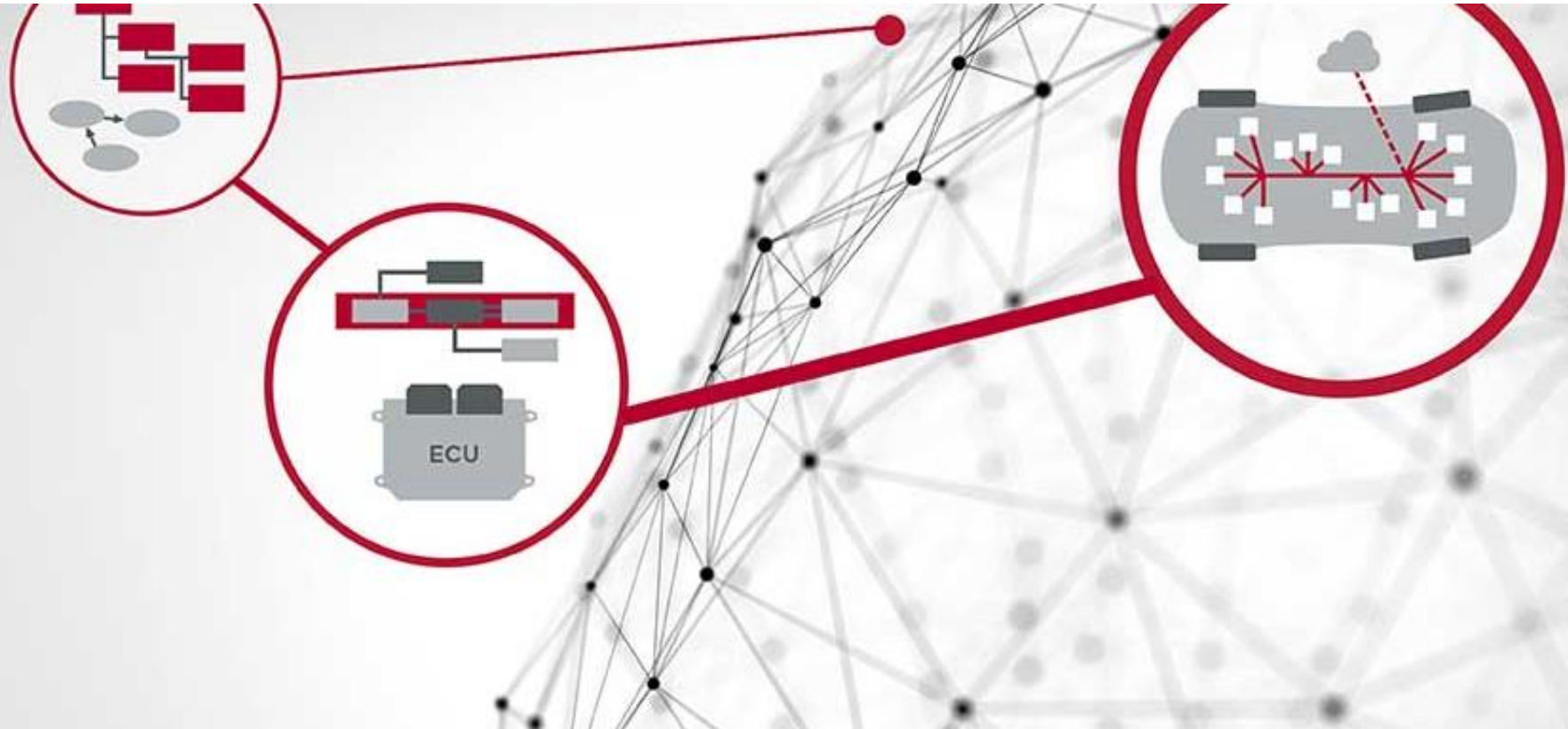




PREvisionによるMBSE

Agenda

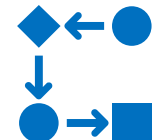
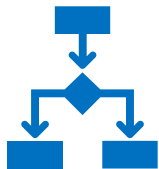
- ▶ **モデルベースシステムズエンジニアリング(MBSE)の基礎**
PREEvisionによるモデルベースシステムズエンジニアリング
まとめ

システムとは?



A system is a combination of interacting elements organized to achieve one or more stated purposes. *1

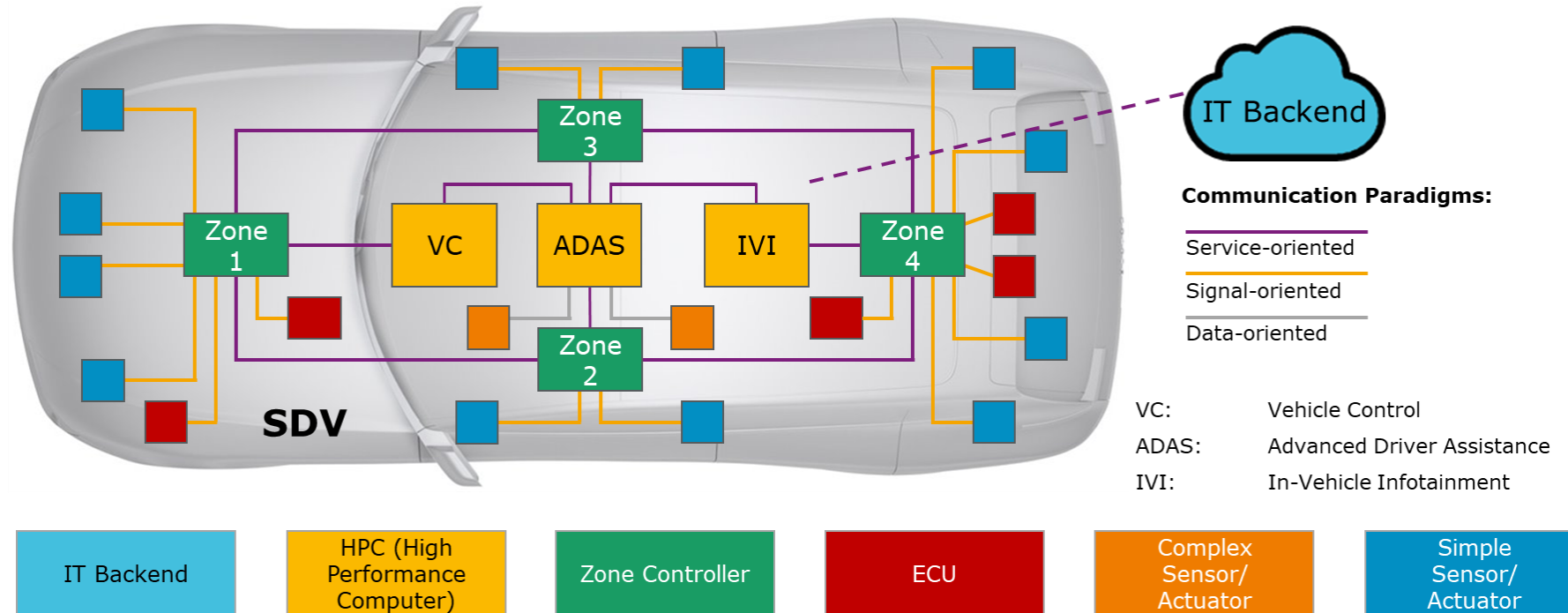
システムとは、1つまたは複数の目的を達成するために組織された
相互作用する要素の組み合わせのこと



※¹ INCOSE(International Council of Systems Engineering)による定義

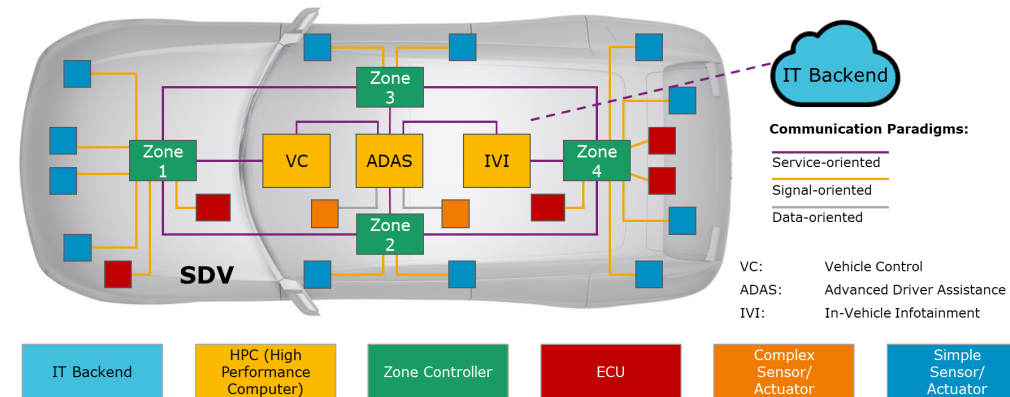
モデルベースシステムズエンジニアリング(MBSE)の基礎

自動車エンジニアリングにおけるシステムとは



自動車 E/E エンジニアリングにおける「システム」とは、HPC、IT バックエンド、ゾーン コントローラー、ECU、センサー、アクチュエーター、電源、ワイヤー ハーネスのネットワーク全体であり、システムを決定し記述する要求とアーキテクチャが含まれる

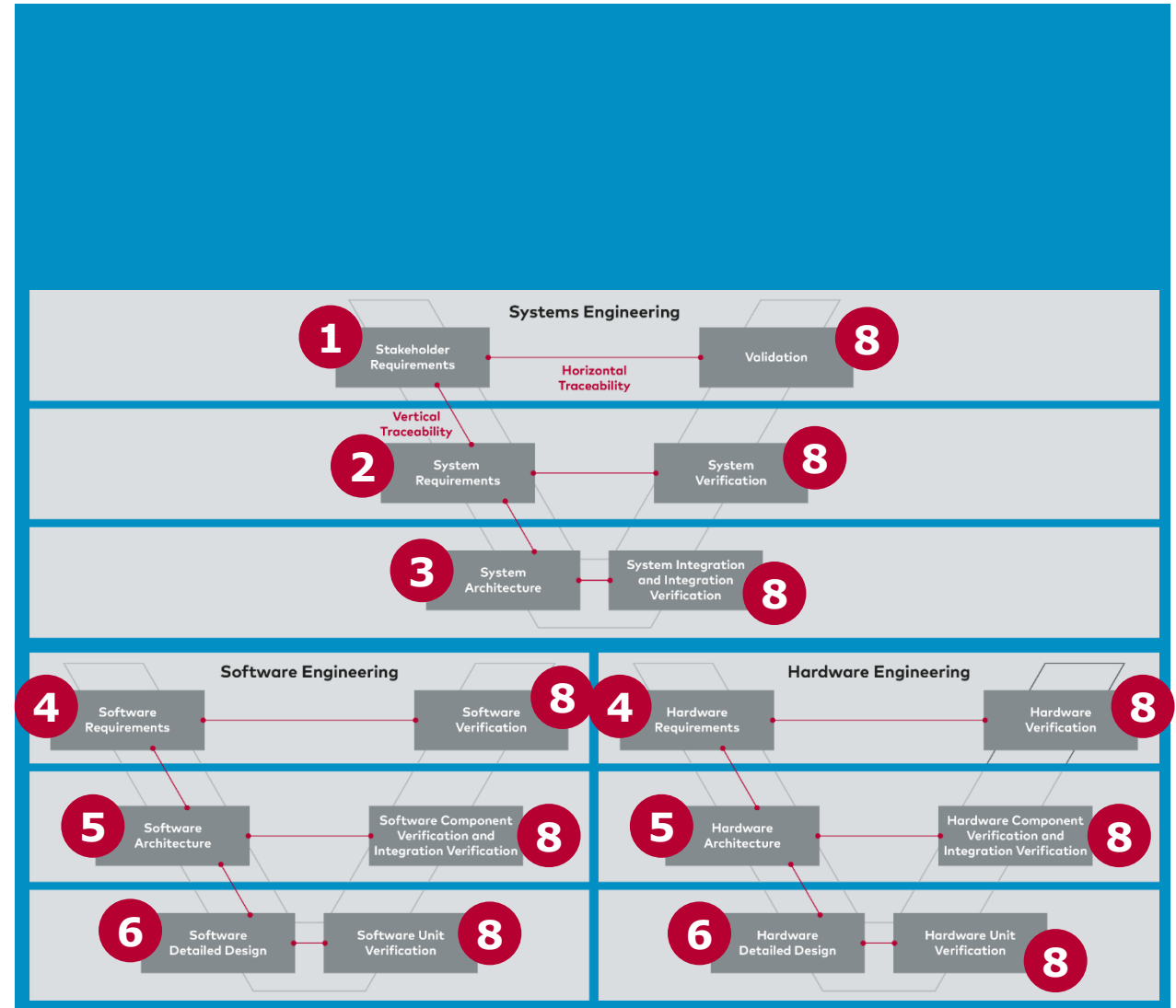
ドキュメントベースに比したモデルベースのシステムズエンジニアリング(MBSE)の優位性



- ▶ “System of Systems”を扱い、複雑なシステムの全体像を見ることを支援
- ▶ システムの複雑な関係、依存関係、ライフサイクルの明確な把握
- ▶ 正確なトレーサビリティ、完全な透明性、統合された変更管理の基盤
- ▶ ステークホルダー要求などの管理、技術要求と安全要求の充足に不可欠
- ▶ ドキュメントはテキストベースの「ブラックボックス」である一方、モデルは形式的に定義された詳細なデータモデルに基づく → 一貫性があり完全に透明な、一元管理された情報源(Single Point of Truth)

モデルベースシステムズエンジニアリング(MBSE)の主なステップ

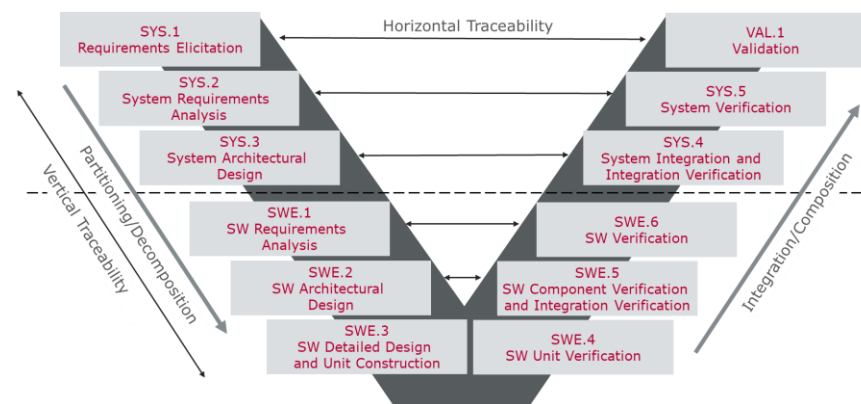
1. システムに対するステークホルダー要求の引き出し
2. 「問題領域」の分析とシステム要求仕様の導出
3. システムアーキテクチャ設計：システム全体から、充足すべき機能を導出。論理アーキテクチャでそれらを配置、文書化、モデル化
4. SW要求仕様、HW要求仕様の導出
5. ハードウェアコンポーネントとソフトウェアコンポーネントで構成されるサブシステムへのシステムの分解により、HWアーキテクチャとSWアーキテクチャを作成/モデル化
6. SW詳細設計、HW詳細設計
7. SW実装、HW実装
8. ベリフィケーション、統合、バリデーション、デリバリー



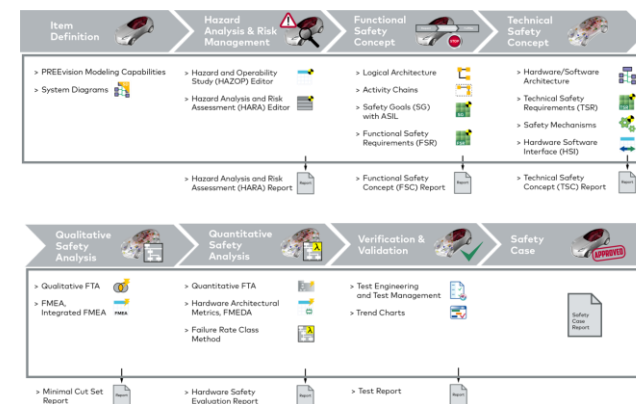
モデルベースシステムズエンジニアリング(MBSE)の基礎

MBSEフレームワーク

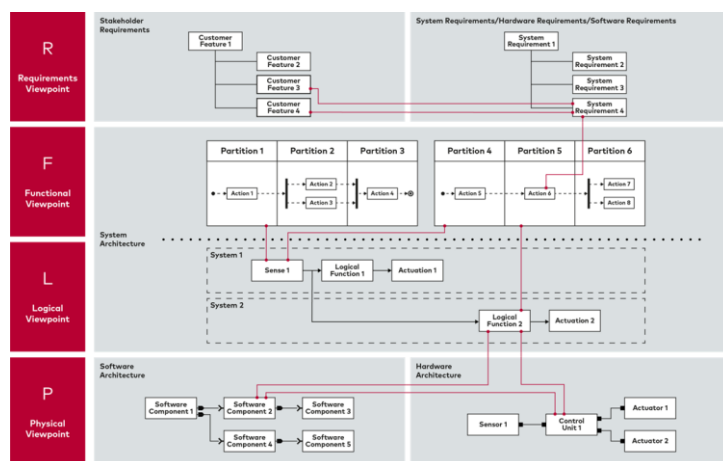
標準仕様 ➡ ASPICE



標準仕様 ➡ ISO26262



メソドロジー ➡ RFLP

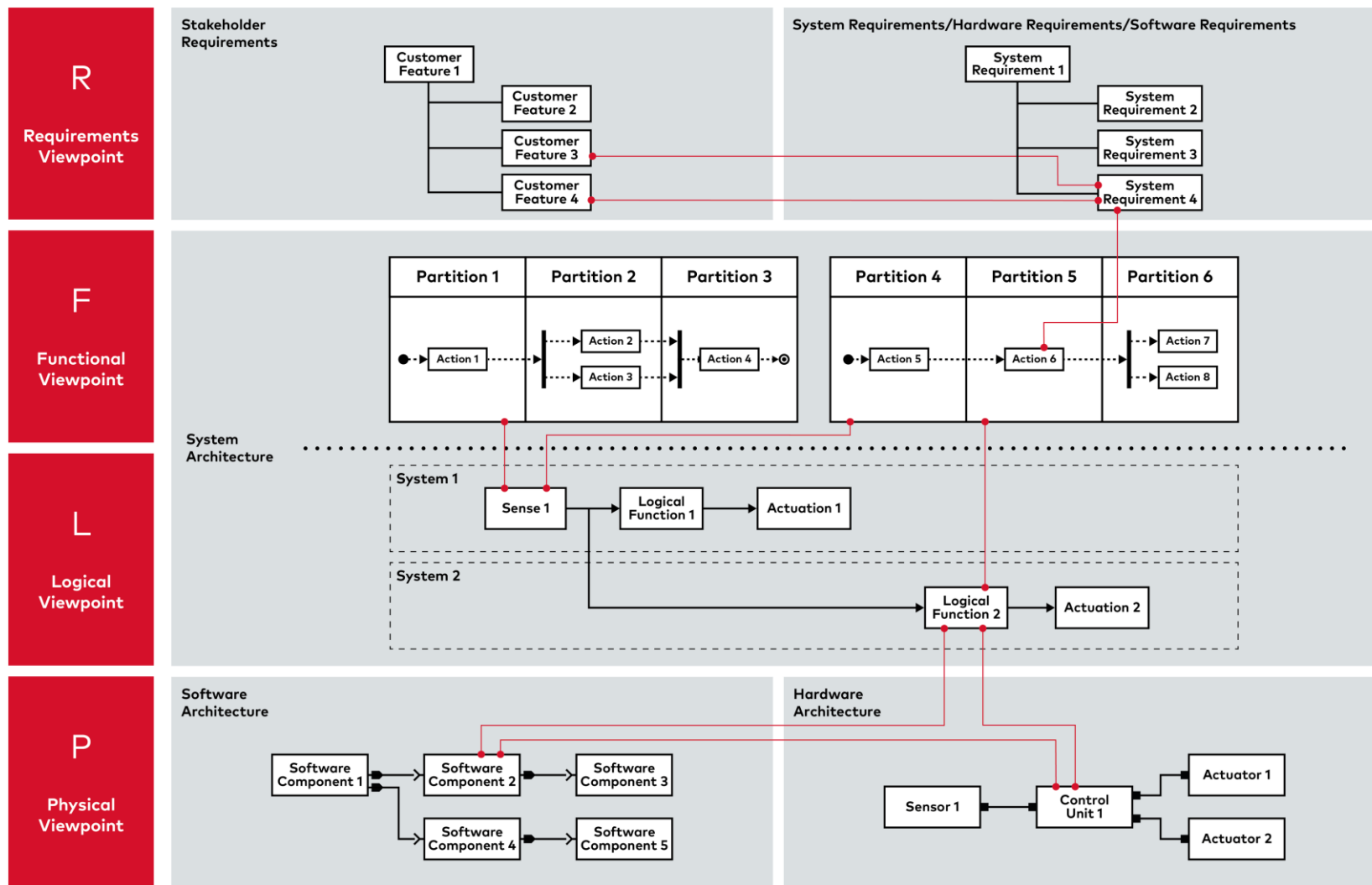


モデリング言語



モデルベースシステムズエンジニアリング(MBSE)の基礎

RFLPメソドロジー



ステークホルダー要求とシステム要求仕様

機能ごとのエンドツーエンドの処理フロー

- ▶ センサーからアクチュエーターまで
- ▶ 入力から出力まで

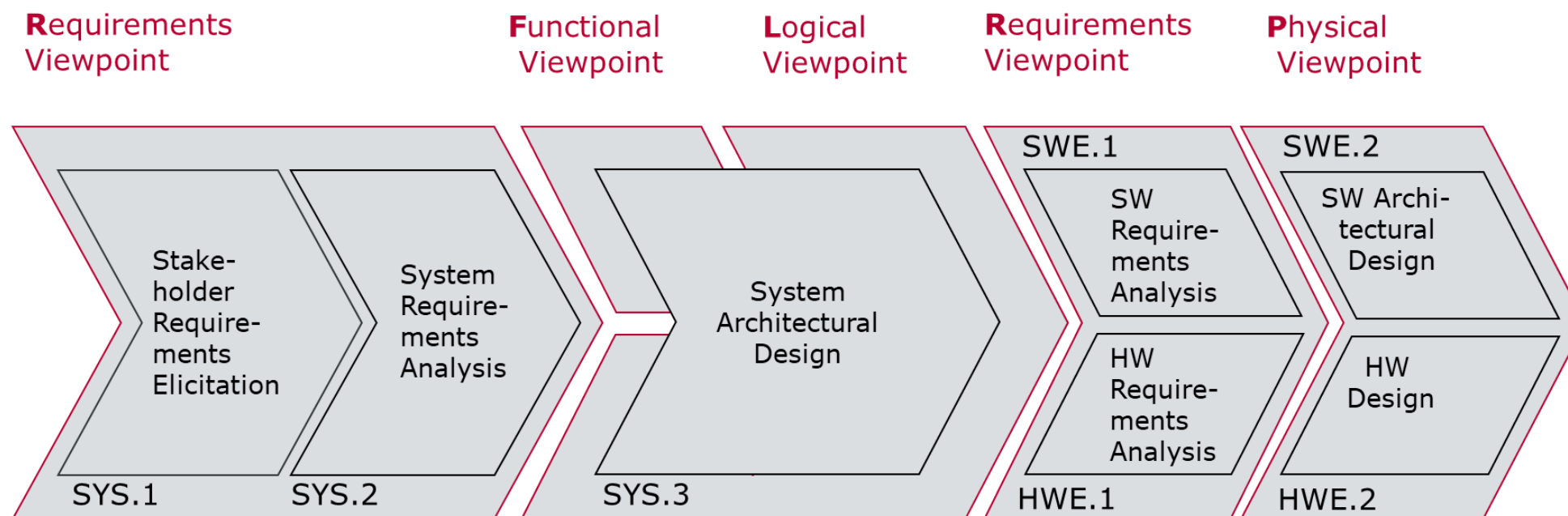
システム観点での機能処理の統合⇒
論理アーキテクチャ

- ▶ システムを論理的な論理制御ブロックで表現
 - > 複数機能が共通利用するセンサー
 - > 複数機能が共通利用するアクチュエーター
 - > サブシステム間のインターフェイス

物理アーキテクチャ

- ▶ ハードウェア
- ▶ ソフトウェア

ASPICEとRFLP – 標準仕様とメソドロジー



議題

モデルベースシステムズエンジニアリング(MBSE)の基礎

▶ **PREEvision**によるモデルベースシステムズエンジニアリング

まとめ

PREEvision – 統合型ソリューション

ステークホルダー要求の管理

技術者以外の人に適した自然言語による要求管理を、PREEvision Requirement Text Editor (RTE) などでサポート

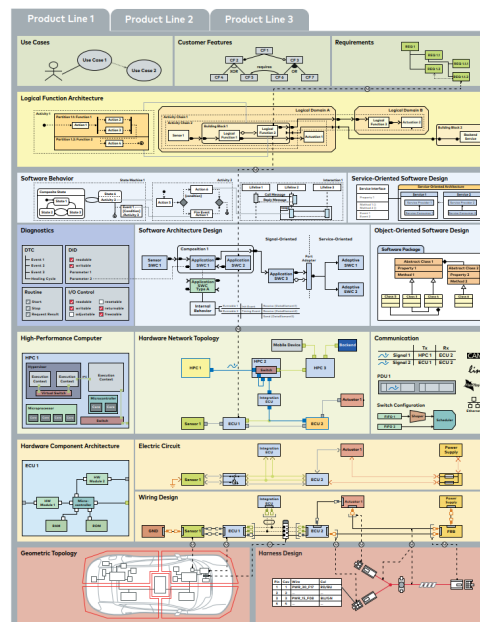
技術要求エンジニアリング

自然言語とモデルで表現

モデル表現の利点：

- システム要求、SW/HW要求に対応する強力な環境
- エンジニアのための理解しやすく普遍的な「言語」
- システムを構築する様々な役割のエンジニアにとって、理解の共通基盤であり、一元管理された情報源(Single Point of Truth)

PREEvisionデータモデル



1つの環境、
1つのユーザーインターフェイス、
1つの用語

システムズエンジニアリング

モデルを解析／変更し、次のような質問に回答可能：

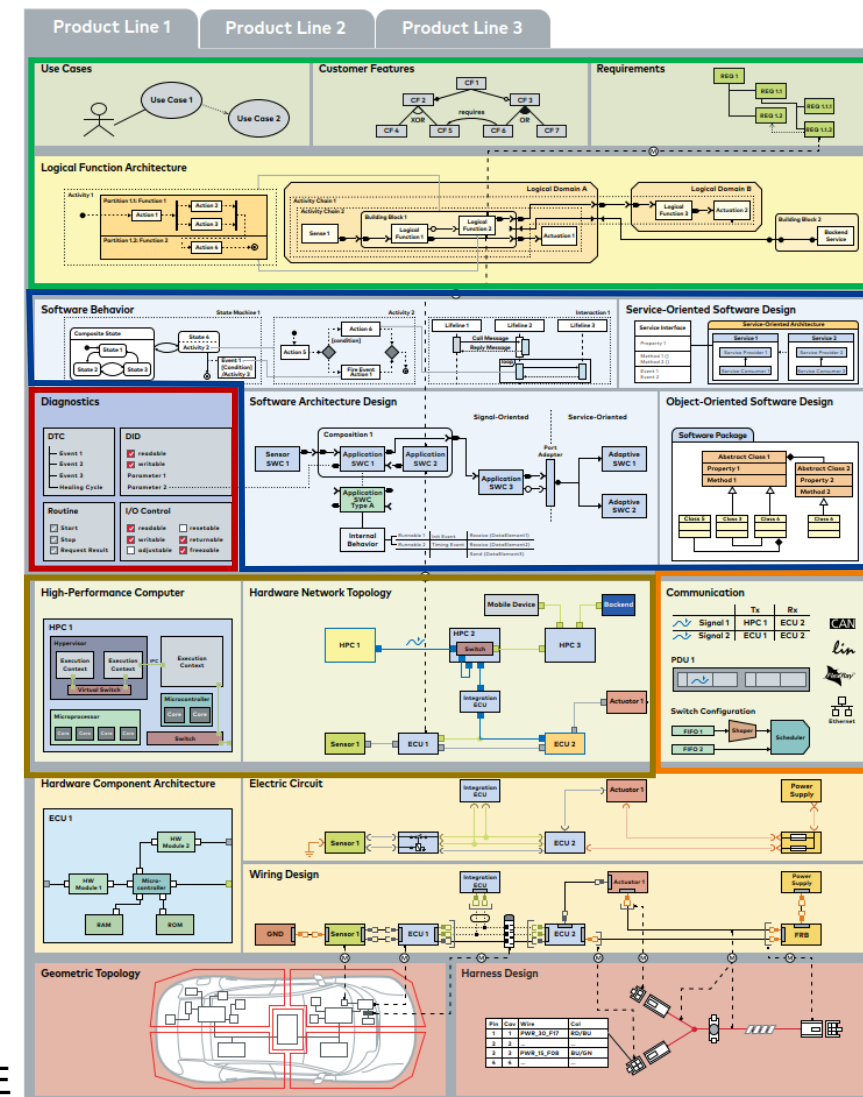
- ▶ このSWCに関連する要求はどれ？
- ▶ サブシステムは整合性をもってモデル化されている？
- ▶ インターフェイスは抜け漏れなく完全に定義されている？
- ▶ このECUを別の場所に移動するとどうなる？
- ▶ ...

システムとコンポーネントを1つのモデルで表現

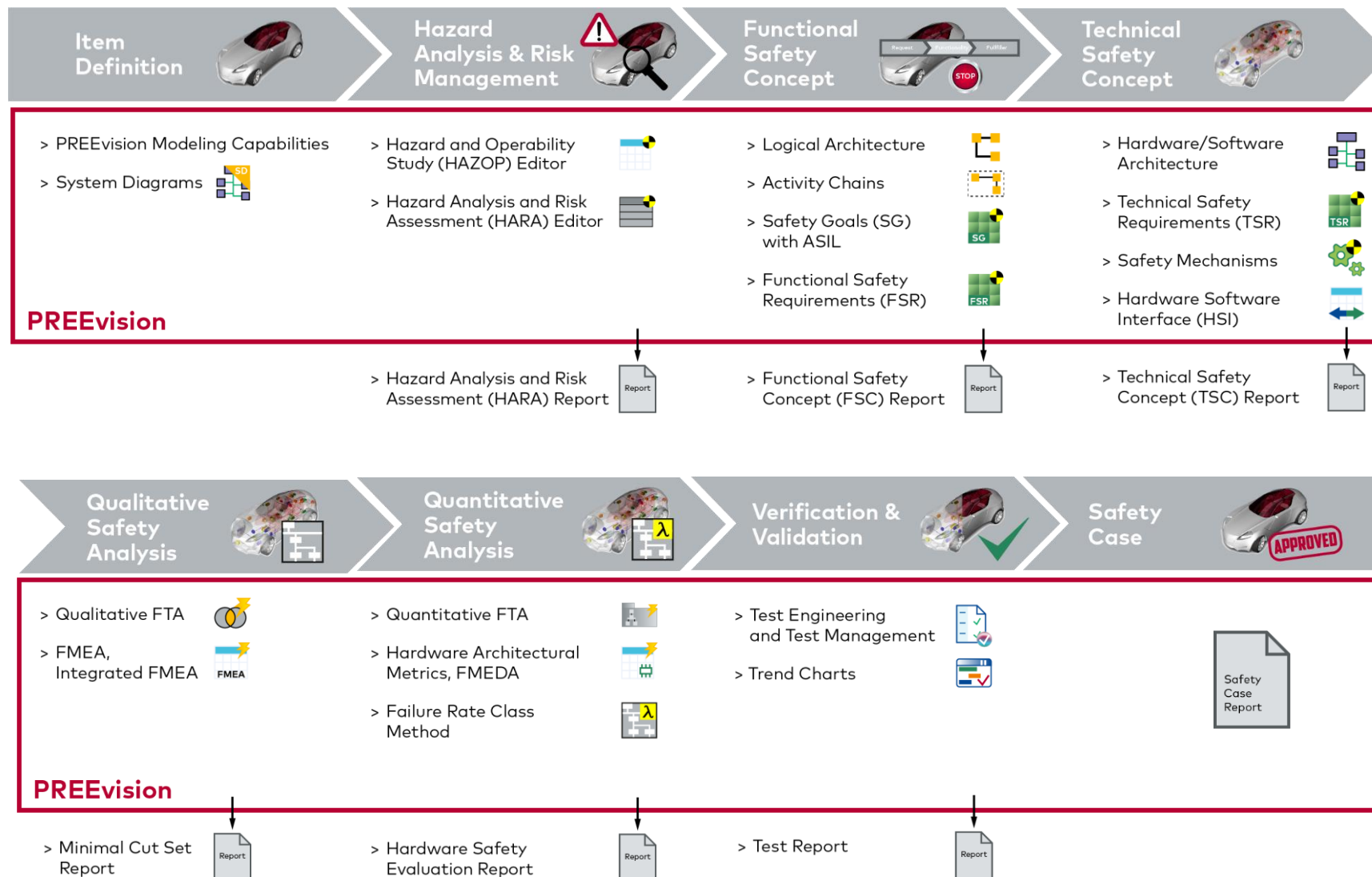
- ▶ 完全なトレーサビリティ
- ▶ 問題、矛盾、誤解、コンセプトレベルの思い違いを早期に発見

自動車および分散型組み込みシステム向けメタモデルと機能

- ▶ 約5,000のクラス、約2,600の属性、約3,200 の関係からなる詳細な自動車および分散型組み込みシステム向けメタモデル
 - ▶ 分散型組み込みシステム向け：
 - > HW、SWCとシグナル／サービスI/F、DBCベースのCAN通信
 - ▶ 自動車向け：AUTOSAR Classic/Adaptiveベースの
 - > HW、サービス/SWC、通信(CAN/LIN/Ethernet/FlexRay)
- ▶ モデルベースの要求エンジニアリング – システムとコンポーネントをモデルで記述。一元管理されたモデル(Single Point of Truth) から仕様書をエクスポート(ReqIF、PDF、MS-Wordなど)。常に最新の状態
- ▶ 「通常の」要求と安全要求を1つのツールで定義・管理。「通常の」要求が安全上重要になった場合、およびその逆の場合、取り扱いが容易
- ▶ モデルベースの診断仕様 – DTCとそのコンテキストなどをシステムスコープでモデル化。ECUスコープのCANdelaStudioとのシームレスな統合
- ▶ 機能安全対応 ➡ ISO26262の完全なモデルベースのサポート(HARA、FSC、TFC、FTA、FMEA、FMEDAなど)。次ページを参照
- ▶ メタモデルをゼロから作る必要なし ➡ 工数削減、組織内・組織間整合性



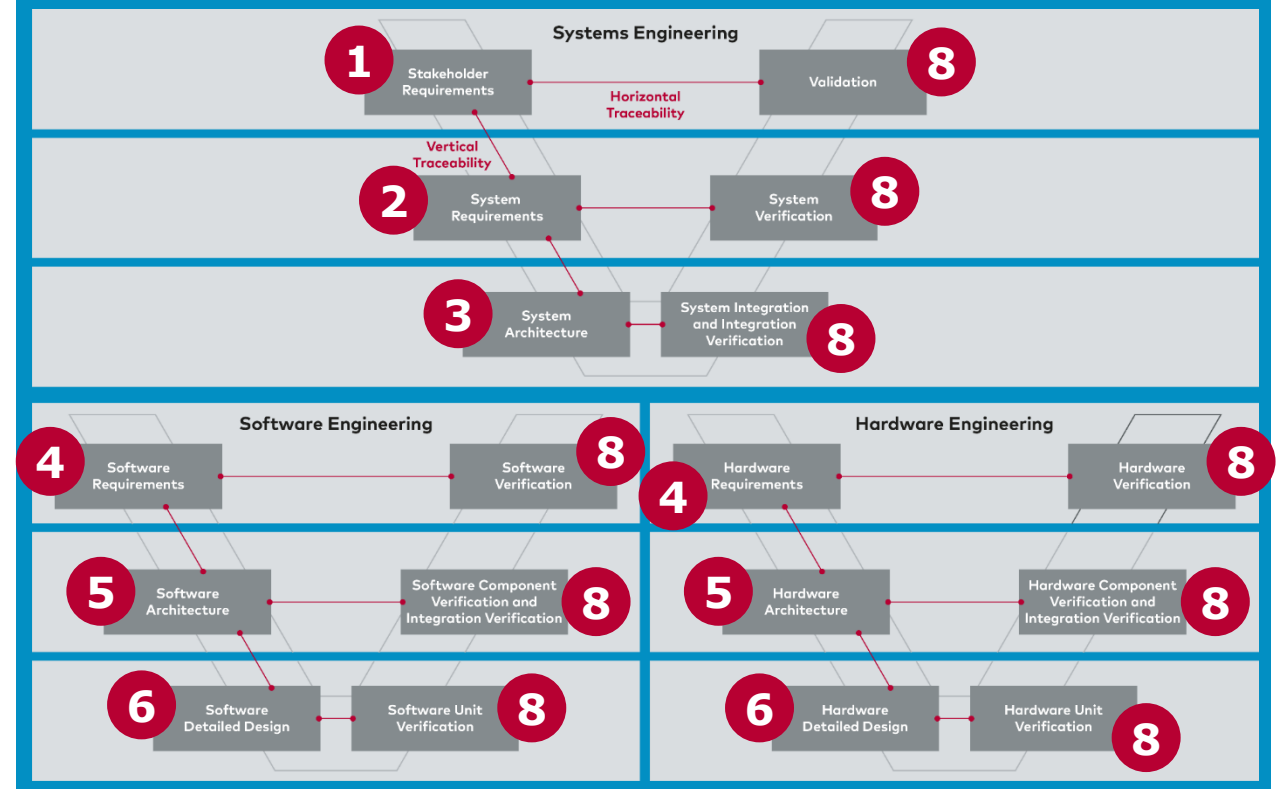
PREEvisionに統合された機能安全ワークフロー



PREEvisionのトレーサビリティ

1. システムに対するステークホルダー要求の引き出し
2. 「問題領域」の分析とシステム要求仕様の導出
3. システムアーキテクチャ設計：システム全体から、充足すべき機能を導出。論理アーキテクチャでそれらを配置、文書化、モデル化
4. SW要求仕様、HW要求仕様の導出
5. ハードウェアコンポーネントとソフトウェアコンポーネントで構成されるサブシステムへのシステムの分解により、HWアーキテクチャとSWアーキテクチャを作成/モデル化
6. SW詳細設計、HW詳細設計
7. SW実装、HW実装
8. ベリフィケーション、統合、バリデーション、デリバリー

PREEvisionは「7. 実装」以外の全ステップに対応



PREEvisionのトレーサビリティ

- ▶ **ASPICE**準拠のプロセスでは、システム要素が統合され、システム/SW/HW要求のベリフィケーション、ステークホルダー要求のバリデーションがなされる

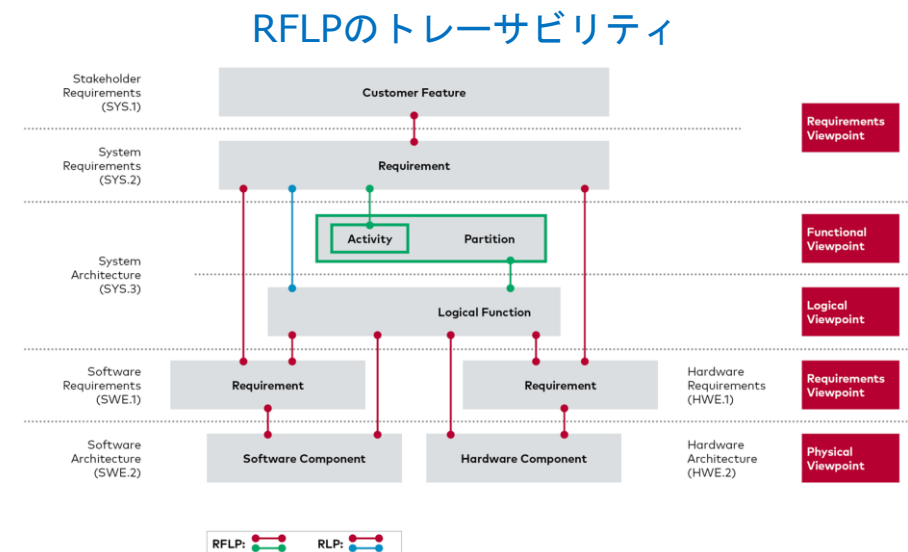
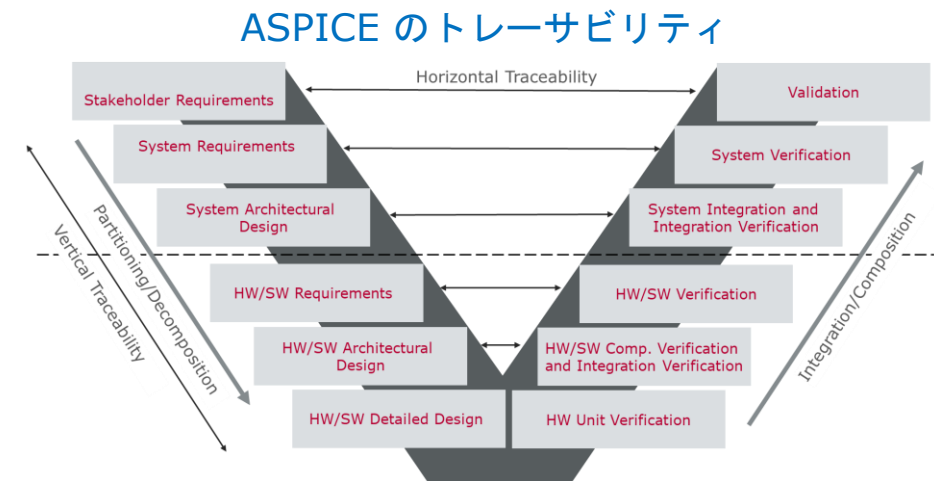
- ▶ PREEvisionでは、テスト仕様と個々のテスト項目をVプロセスの左側にあるシステム/SW/HW要求にリンク

例 “システムテスト仕様” ➡ “システム要求”:

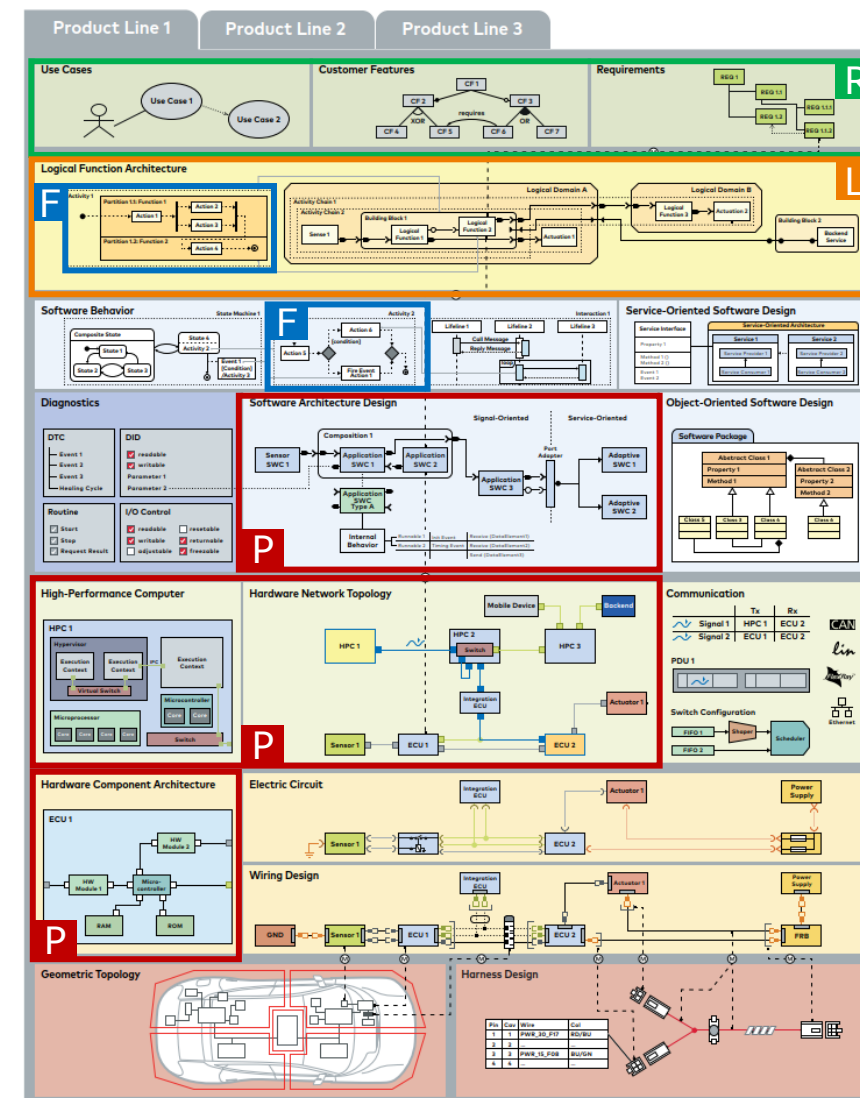
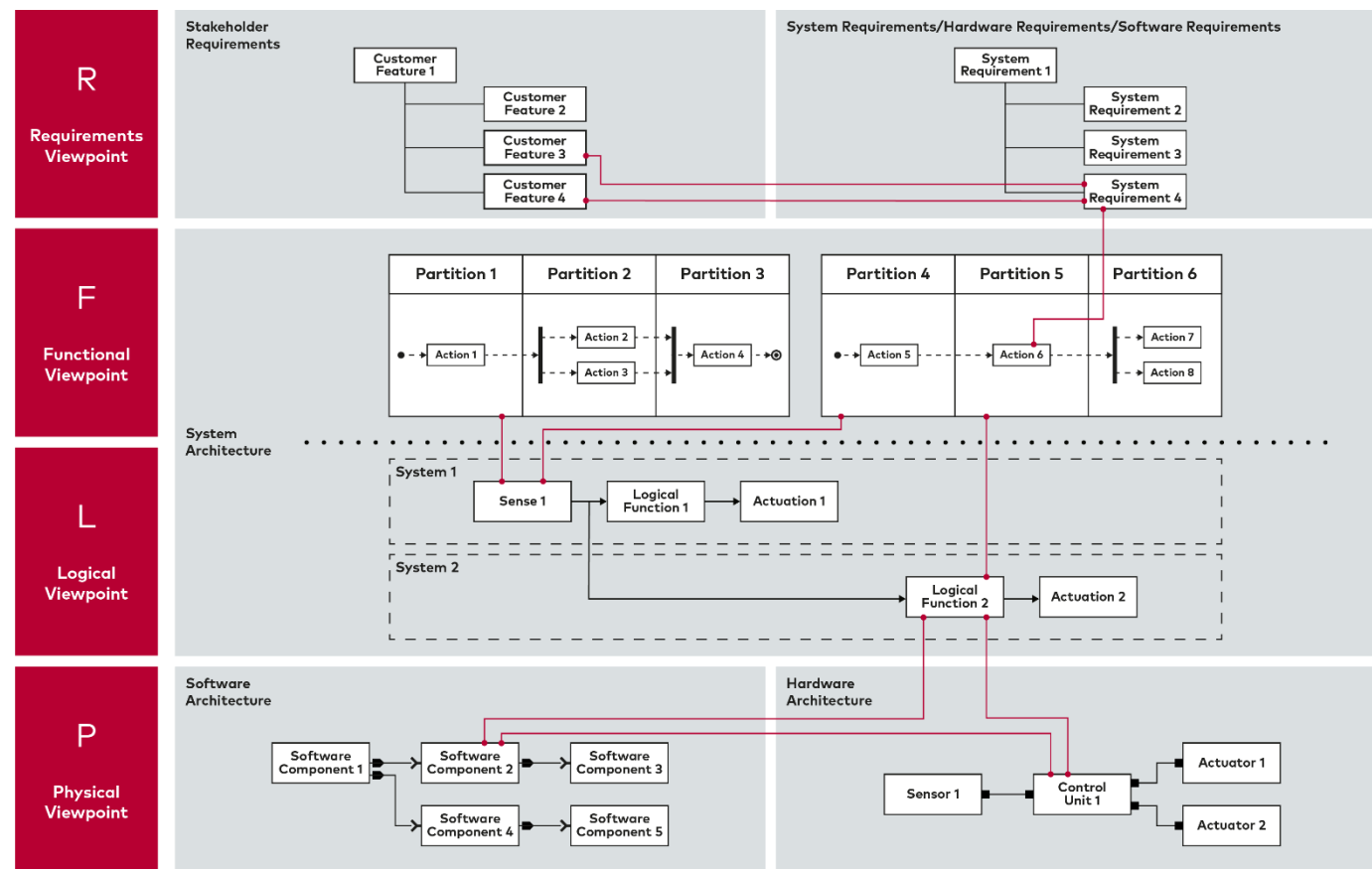
LEV...	Custom ID	Hierar...	Text	Type	#	Prio...	Comment	Req/CF Links	Test Demand Links	Test Condition Links	Goal	Precondition	Action
Δ 1-1			Test Data Package	Test Data Package									
Δ 2-2			1 Test Specification	Test Specification				0					
Δ 3-3			1.1 E-Call Test Item	Test Item				1	0	0	E-Call is artificially triggered in a vehicle driving over 20kmph		
Δ 4-4			Start vehicle	Test Item Step (Init)				0	0	0			Press brake ped
Δ 5-4			Vehicle driving forwards over 20kmph	Test Item Step (Init)				0	0	0			Change driving i accelerate
Δ 6-4			Trigger crash 'severe' state	Test Item Step (Run)				0	0	0			Using applicatio variable CrashIS
Δ 7-4			Shutdown vehicle	Test Item Step (Shutdown)				0	0	0			Stop the vehicle, and press start

- ▶ **RFLPメソドロジー**は、ステークホルダー要求からシステム要求、論理機能、HW/SWコンポーネントまで、双方向の垂直トレーサビリティを確立するのに役立つ

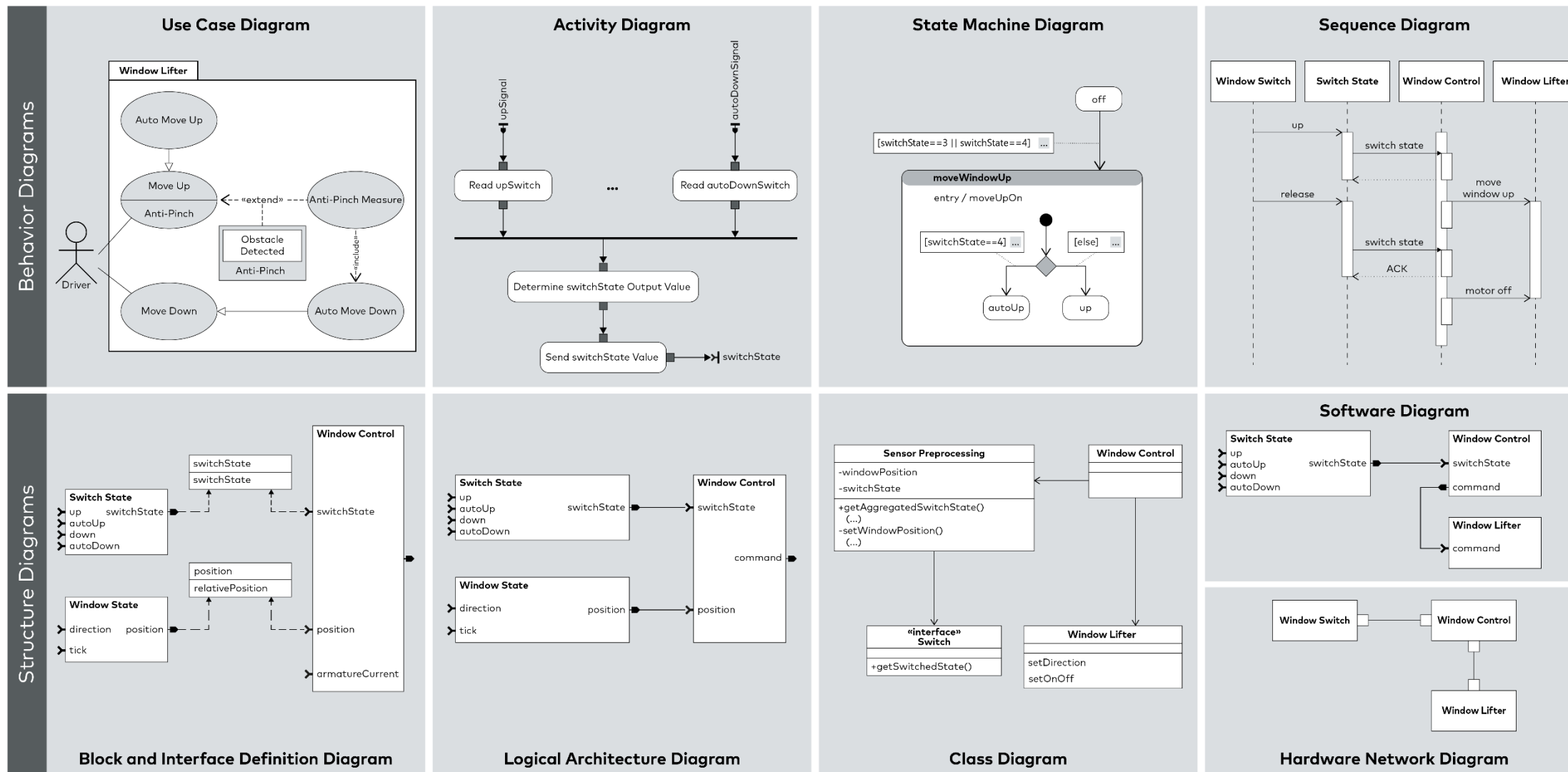
- ▶ RFLPメソドロジーとPREEvisionデータモデルのマッピング：次ページ参照



PREEvisionにおけるRFLPメソドロジー



PREEvisionのSysML対応



PREEvision Systems Engineering Explorer

エクスプローラーのカテゴリ

「システムアーキテクチャ」全体の「ステークホルダー要求」から「SW/HW詳細設計」までシームレスに対応

トレーサビリティエディター ...

「Link to System Requirements」

...を選択すると垂直トレーサビリティを表示

「トレーサビリティ エディター」のハイライトで、トレーサビリティの欠如などをわかりやすく表示

The screenshot displays the PREEvision Systems Engineering Explorer interface. The left sidebar shows a tree view of the project structure, including Stakeholder Requirements, System Requirements, System Architecture, Software Requirements, Software Architecture, Software Detailed Design, Hardware Requirements, Hardware Architecture, and Hardware Detailed Design. The main area is divided into two panes: the Stakeholder Requirements Editor and the System Requirements view.

Stakeholder Requirements Editor: This pane shows a table with two columns: Stakeholder Requirement and Linked System Requirement. The table lists various requirements, such as "1 Robot vacuum cleaner" and "1.1.1 Stakeholder Requirement (Standard)". A red arrow points to the "Link to System Requirements" button in the toolbar.

System Requirements: This pane shows a table with two columns: System Requirement and Linked Stakeholder Requirement. The table lists various system requirements, such as "1.1.1.2 Cleaning function" and "1.1.1.3 Operating". A red arrow points to the "Link to Test Specification" button in the toolbar.

Color Meaning: A tooltip is visible in the bottom right corner, explaining the color coding used in the tables:

- Yellow: Link missing
- Blue: Ambiguous link
- Pink: Unexpected link

PREEvision Systems Engineering Explorer

エクスプローラーのカテゴリ

「システムアーキテクチャ」全体の「ステークホルダー要求」から「SW/HW詳細設計」までシームレスに対応

トレーサビリティエディター ...

「Link to Test Specification」

...を選択すると水平トレーサビリティを表示

「トレーサビリティ エディター」のハイライトで、トレーサビリティの欠如などをわかりやすく表示

The screenshot displays the PREEvision Systems Engineering Explorer interface. On the left, a tree view shows the project structure under 'SE Example: Robot Vacuum Cleaner'. The main area is divided into two panes. The left pane, titled 'Stakeholder Requirements', shows a table with columns for 'Stakeholder Requirement' and 'Linked Validation Specification'. The right pane, titled 'Validation Specification', shows a table with columns for 'Validation Specification' and 'Linked Stakeholder Requirement'. A 'Color Meaning' legend on the right indicates that yellow highlights represent 'Link missing' and pink highlights represent 'Unexpected link'. Red arrows point from the text boxes above to specific elements in the interface: one to the 'Stakeholder Requirements' tree, one to the 'Link to Test Specification' button, and one to the 'Color Meaning' legend.

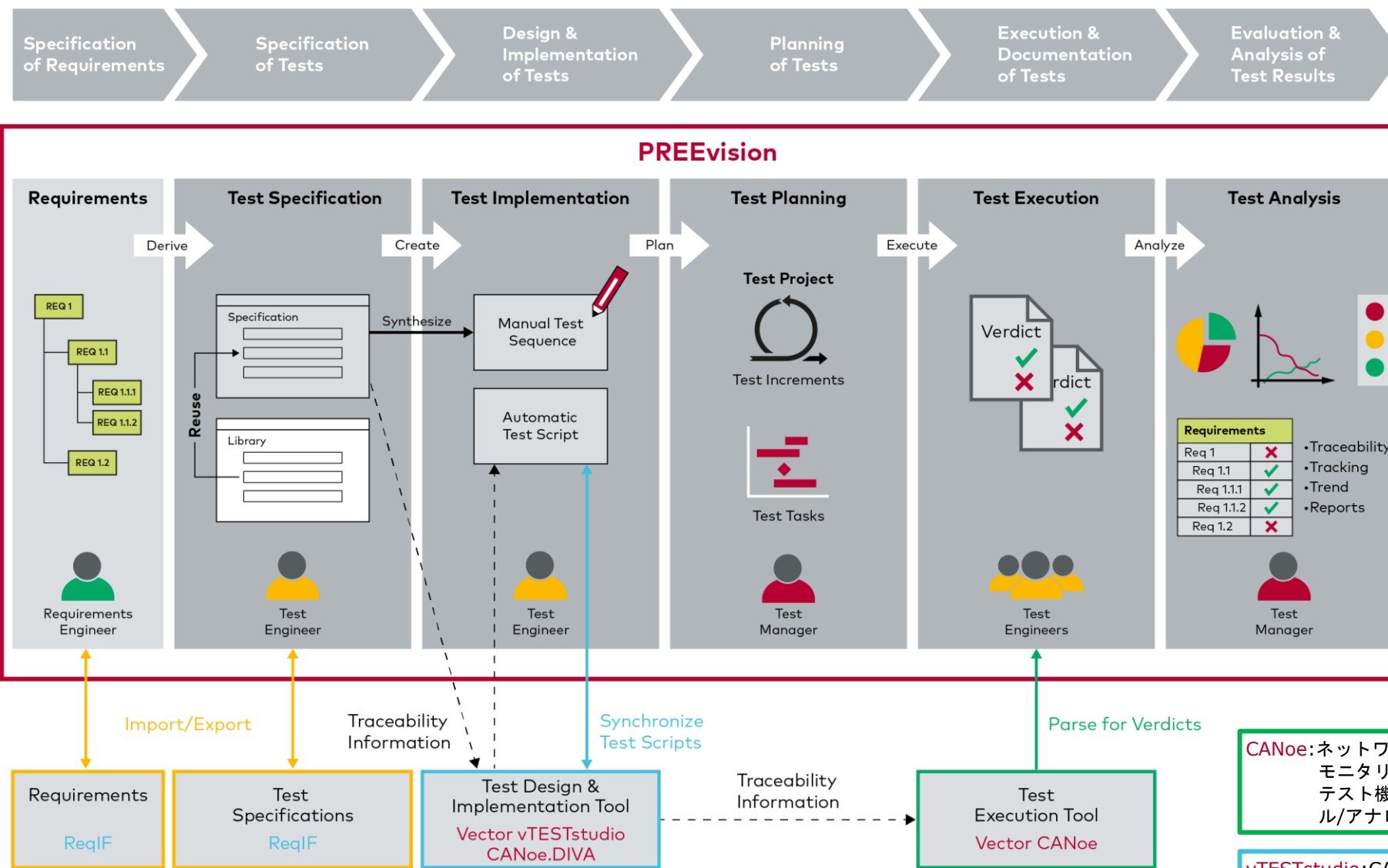
Stakeholder Requirement	Linked Validation Specification
1 Robot Vacuum Cleaner / -:- (Customer Feature Package)	1 Test Robot Vacuum Cleaner
1.1 Robot vacuum cleaner / -:- (Customer Feature)	
1.1.1 Stakeholder Requirement (Standard) / -:- (Cust...)	
1.1.1.1 Cleaning area / -:- (Customer Feature)	
[SYS.1] 1 The cleaning area shall be captured.	
1.1.1.2 Thorough cleaning on different floor coveri...	
[SYS.1] 2 The vacuum robot shall be able to be ...	
[SYS.1] 3 The vacuum robot shall be able to be...	
1.1.1.3 Long operating time / -:- (Customer Feature)	
[SYS.1] 4 The vacuum robot shall clean a 4-ro...	
1.1.1.4 Detection Landing / -:- (Customer Feature)	
[SYS.1] 5 A landing shall be detected	
1.1.1.5 Collision Avoidance (Standard) / -:- (Custo...	
[SYS.1] 6 An intelligent obstacle avoidance ma...	
1.1.1.5.1 No collision with a stair landing	
[SYS.1] 7 An intelligent obstacle avoidance ma...	
1.1.1.5.1 No collision with objects in the room	
1.1.2 Stakeholder Requirement (Advanced) / -:- (Cust...	
1.1.2.1 Collision Avoidance (Advanced) / -:- (Custo...	
[SYS.1] 7 An intelligent obstacle avoidance ma...	
1.1.2.1.1 No collision with humans and animals in the room	
1.1.2.2 Smart Communication Technology / -:- (Cu...	

Validation Specification	Linked Stakeholder Requirement
1 Test Robot Vacuum Cleaner	1 Robot Vacuum Cleaner
1.1 Robot vacuum cleaner	
1.1.1 Stakeholder Requirement (Standard)	
1.1.1.1 Cleaning area	
1.1.1.2 Thorough cleaning on different floor coverin...	
1.1.1.3 Long operating time	
1.1.1.4 Detection Landing	
1.1.1.4.1 No collision with a stair landing	[SYS.1] 5 A landing shall be detected
1.1.1.5 Collision Avoidance (Standard)	
1.1.1.5.1 No collision with objects in the room	[SYS.1] 6 An intelligent obstacle avoidance maneuver strategy
1.1.2 Stakeholder Requirement (Advanced)	
1.1.2.1 Collision Avoidance (Advanced)	
1.1.2.1.1 No collision with humans and animals ...	[SYS.1] 7 An intelligent obstacle avoidance maneuver strategy
1.1.2.2 Smart Communication Technology	

Color Meaning:

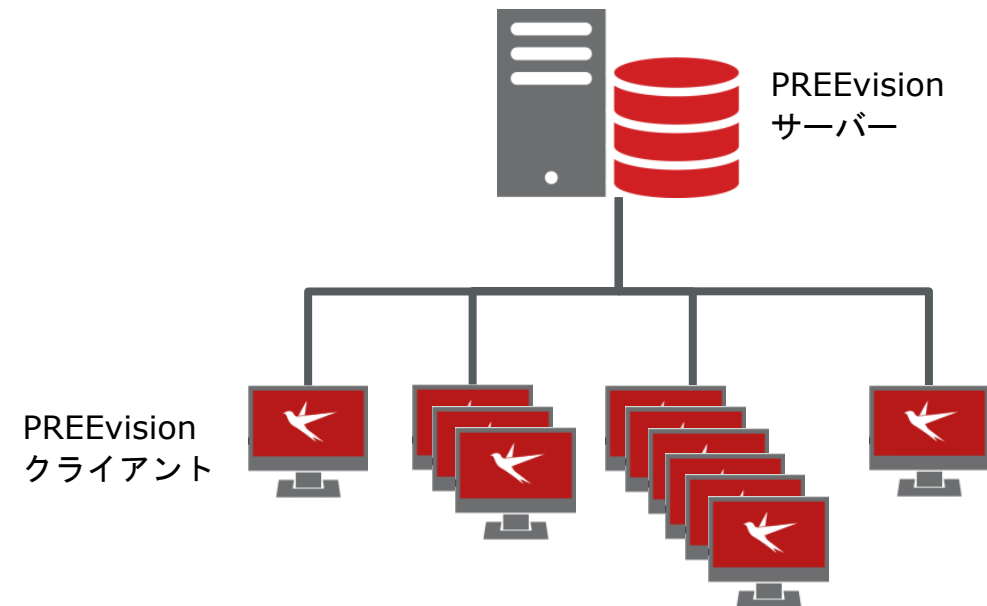
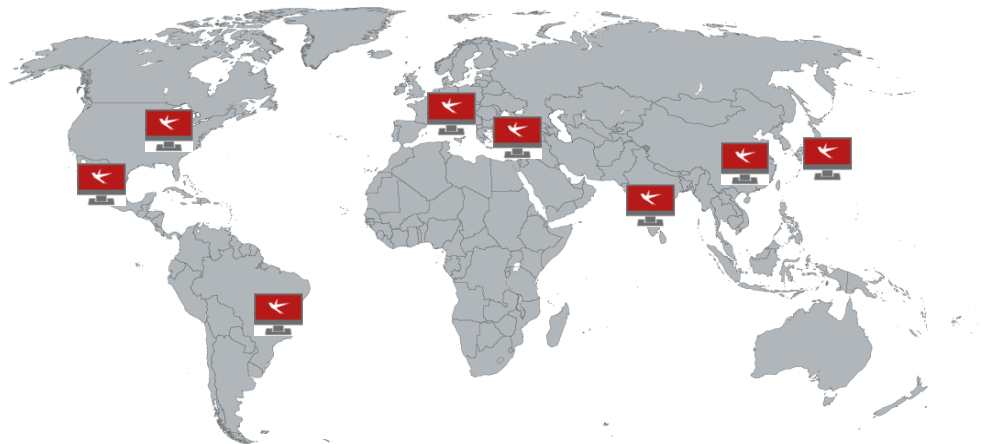
- Yellow: Link missing
- Pink: Unexpected link

PREEvisionによるテストエンジニアリングとテスト管理：ワークフロー

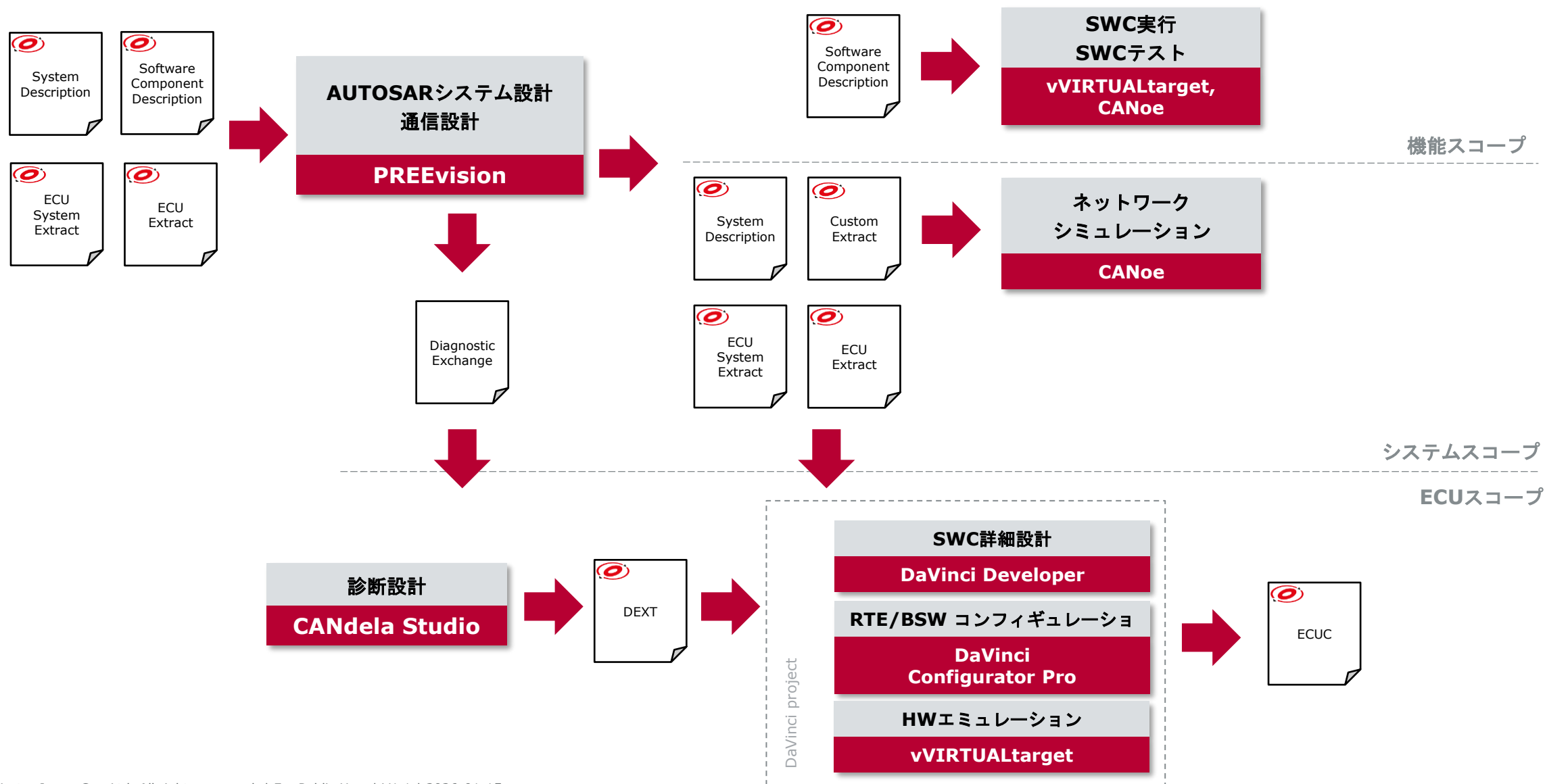


複数ユーザーによる協調設計の支援

- ▶ 協調設計支援のためのPREEvision製品Edition ➡ Team Edition
 - ▶ クライアント・サーバー ベース: 設計データを格納するサーバーに、複数ユーザーが複数の拠点からアクセス可能
 - ▶ バージョン管理、変更履歴管理
 - ▶ 変更管理
 - ▶ ライフサイクル管理(応用例: 設計変更の承認フロー定義)
 - ▶ アクセス権管理(Roles & Rights)
- ▶ 複数ユーザーによる協調設計に必須
- ▶ サーバーの所在
 - ▶ オンプレミス=ユーザー企業が用意
 - ▶ ベクターによるホスティング



Backup: ベクターAUTOSARツールチェーン



議題

モデルベースシステムズエンジニアリング(MBSE)の基礎
PREEvisionによるモデルベースシステムズエンジニアリング

▶ まとめ

PREEvision – システムズエンジニアリングのための総合的ソリューション

統合型 ソリューション

- ▶ 市場の他のソリューションと比較し、PREEvisionは統合された「ワンツールソリューション」を提供
- ▶ システム／SW／HW各領域の要求、設計、テスト仕様の定義
- ▶ バージョン管理、バリエーション管理、変更管理、ライフサイクル管理を統合

MBSE エコシステム 対応

- ▶ モデリング言語 SysMLに対応。必要なすべてのエディター（例：アクティビティ図、シーケンス図）を装備
- ▶ RFLPのようなメソドロジーに対応

分散型組み込み システム対応

- ▶ システム／SW／HW用メタモデル
- ▶ DBCベースのCAN通信を含む
- ▶ メタモデルをゼロから作る必要なし→工数削減、組織内・組織間整合性

自動車 システム対応

- ▶ ASPICEやISO26262などの標準仕様に対応
- ▶ シミュレーション、コード生成に、AUTOSARツールチェーンとして対応

まとめ

無償トライアル

デモ動画、無償トライアル環境申込み（トライアル用のクラウド上仮想PCへのアクセス）

- ▶ <https://www.vector.com/jp/ja/products/products-a-z/software/preevision/systems-engineering/rflp/>



VECTOR 

お問い合わせ 日本 | 日本語 買

製品 ノウハウ イベント サポート/ダウンロード 採用情報 会社情報

ホーム ▶ 製品 ▶ 製品一覧 ▶ ソフトウェア ▶ PREEvision ▶ システムズエンジニアリング ▶ RFLP

R
Requirements Viewpoint

F
Functional Viewpoint

L
Logical Viewpoint

P
Physical Viewpoint

PREEvision: 一つのMBSEツールでRFLPを完全サポート

ギャップのないMBSE

PREEvisionはRFLPのアプローチ、すなわち要求（Requirements）、機能（Functional）、論理（Logical）、物理（Physical）のビューポイントを一つの統合環境でサポートする、高機能なMBSE（モデルベースシステムズエンジニアリング）ツールです。複雑なシステムズエンジニアリングへの対応に軸足を置いた設計により、シームレスなモデリング、トレーサビリティ、コラボレーションをあらゆる抽象化レベルで可能にします。

- RFLPを利用する理由
- PREEvisionを利用する理由
- SysMLを超えたMBSE
- RFLPのはじめの一歩

デモを見る →

お問い合わせ

ホワイトペーパー

- ▶ <https://www.vector.com/jp/ja/products/products-a-z/software/preevision/knowledge-hub/white-paper-systems-engineering/>



VECTOR 

お問い合わせ 日本 | 日本語 買

製品 ノウハウ イベント サポート/ダウンロード 採用情報 会社情報

ホーム ▶ 製品 ▶ 製品一覧 ▶ ソフトウェア ▶ PREEvision ▶ ナレッジハブ ▶ 自動車システムエンジニアリング

シームレスでアジャイル、しかも全体的な自動車システムエンジニアリングをPREEvisionで

将来性を備えたE/Eシステムエンジニアリングの実践とは

→ 要約

→ ダウンロード

お問い合わせ

要約

自動車の電気/電子アーキテクチャ（E/Eアーキテクチャ）は、デジタル化への転換を迎えています。高性能コンピューターを搭載したSDV（Software-Defined Vehicle）では、外部システムとの接続、エッジ/クラウドコンピューティング、OTAでのソフトウェア更新が可能となっています。

オープンで動的な接続が可能ないわゆるサイバーフィジカルシステムでは、ソフトウェアアプリケーションを定期的に更新し、更新することができ、そしてそれをホストのゲートウェイ指向アーキテクチャ

← PREEvision

WHITE PAPER

ベクター・ジャパン株式会社

www.vector.com/jp/ja/

- ▶ 営業部および各種お問い合わせ（Eメール）
- ▶ サポートリクエスト

※記載内容については予告なく変更されることがありますので、あらかじめご了承ください。