20XX-01-XXXX

Optimal test strategy for Clusters and CDCs in multi power train vehicle ecosystem

Author, co-author (Do NOT enter this information. It will be pulled from participant tab in MyTechZone)

Affiliation (Do NOT enter this information. It will be pulled from participant tab in MyTechZone)

Abstract

Innovation in energy storage and generation system will lead to multiple power train solutions across the vehicle categories in the Automative segment. With various options to the end consumer across different vehicle segments, the complexity associated with E/E Architecture and software engineering will be multi-fold both for the OEMs and Suppliers. Over the air updates shall become mandatory features to manage this complexity and to calibrate the vehicle features inline with changing trends and efficiency plus feature enhancements in post-market release scenarios. These upgrades are more common in digital clusters, in-vehicle entrainment and central digital cockpits. OEMs are introducing the vehicle platforms in multiple power train variants keeping comfort, instrument clusters and in-vehicle entertainment as core features across different power trains.

A well-defined and managed comprehensive optimal test strategy and infrastructure will be critical to ensure seamless release of the software solutions to different power trains during the product development phase and post-launch software upgrades.

In this novel work, we extensively explore the current practices of segregating features through common versus specific powertrains, managing the overall test strategy across varied test types, test data and test infrastructure; then address the advantages and challenges in our current practices. The paper would summarize the optimal test strategy in approaching the software development pipeline for a multi-powertrain architecture and focus specifically on early test-driven interventions for robust software deployment on production for both parallel and staggered release pipeline.

Introduction

Aim to create the test strategy document which satisfies requirements of cluster and CDC ECUs developed based on different types of powertrain vehicles, in this paper a well-defined test concepts and methodologies are mapped into traditional test strategy development stages and proposal of common test framework to achieve testability of multi powertrain architecture-based software implementations.

**Proposed Test strategy:**

To establish a **common test strategy** across multiple powertrain platforms by aligning with:

* OEM’s vehicle architecture
* System architecture
* Software architecture (Owned by Software Suppliers)

**Architectural Stages for Test Strategy Alignment**

1. **OEM’s Vehicle Architecture**
   * Defines the overall vehicle-level integration and communication.
   * Influences feature behavior across ECUs and subsystems.
2. **System Architecture**
   * Represents the functional and logical design of the vehicle systems.
   * Determines how features interact across domains (e.g., powertrain, infotainment).
3. **Software Architecture (Supplier-Owned)**
   * Includes AUTOSAR layers (Base Software, Middleware, Application Software).
   * Managed by Tier 1 suppliers responsible for ECU software delivery.

**Test Strategy Deployment**

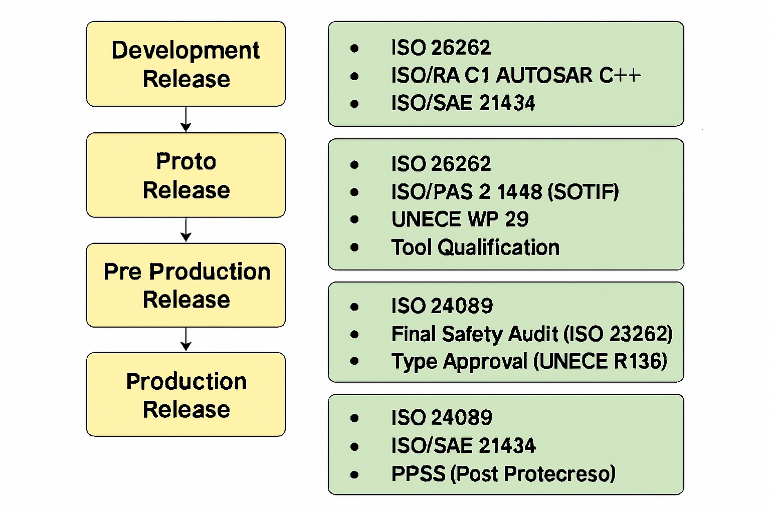
* **Feature-Based Testing**: Each stage contributes to identifying and validating feature sets.
* **Strategy Tailoring**: Test methods and environments are adapted based on the architecture stage and feature complexity.

**Release Lifecycle Integration**

Test strategy is applied across the following software release stages:

1. **Development Release** – Unit and module-level testing.
2. **Proto Release** – Integration and early validation.
3. **Pre-Production Release** – System-level and performance testing.
4. **Production Release** – Final validation and compliance.
5. **Post-Production Release** – Maintenance, diagnostics, and updates.

**Compliance each stage**



**1. Vehicle Architecture Level**

**Focus**: High-level vehicle functions and integration

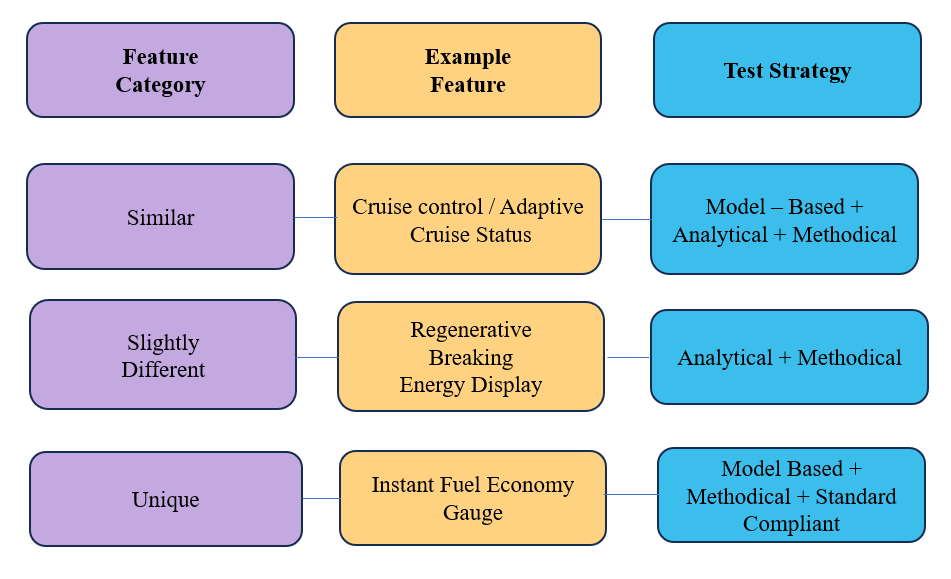
At the vehicle architecture level, Tier 1 suppliers derive features by analyzing the OEM’s vehicle architecture perspective. These features are identified and grouped into the following categories:

1. Similar to Powertrain
2. Slightly Different Across Powertrain
3. Unique Features

Once features are categorized, the corresponding functionalities are mapped to the appropriate test strategy, test type, test infrastructure, and deployment method. This feature mapping enables architects to:

* Identify common test methodologies that can be reused across multiple features.
* Allocate more focused effort toward testing and validating unique or OEM-specific features.

This structured approach ensures efficient test planning, reduces redundancy, and enhances the overall quality and integration of vehicle-level functionalities such as the Instrument Cluster and Central Display Controller (CDC).



**2.System Architecture Level**

**Focus**: Subsystems such as powertrain, braking, and battery management

Analyzing the system architecture is essential for defining a common test strategy. This level of architecture helps link use cases to system functionalities, enabling clear communication between subsystems. It also supports:

* Prioritization of messages based on the criticality of features
* Response mechanisms to malfunctions and errors
* Impact analysis of software failures across systems

At this stage, we derive the common test strategy for powertrain systems and map each feature to:

* Test Strategy (e.g., Model-Based, Analytical, Methodical)
* Test Type (e.g., SIL, HIL, Integration, Regression)
* Test Infrastructure (e.g., SIL Environment, HIL Rig, Emulators)
* Deployment Method (e.g., CI/CD Pipeline, Manual Validation)

This structured mapping ensures consistency in testing across shared functionalities while allowing targeted validation for unique or OEM-specific features.

**3.Software Architecture Level**

**Focus Area**

* **Domain**: ECU Software, Algorithms, Diagnostics
* **Architecture Framework**: AUTOSAR (Automotive Open System Architecture)

**AUTOSAR Software Architecture Stages**

AUTOSAR is divided into two main paradigms:

* **Classical AUTOSAR**
* **Adaptive AUTOSAR**

Each ECU software stack typically includes:

1. **Base Software (BSW)**
2. **Middleware**
3. **Application Software**

**Special Considerations for Instrument Cluster & Infotainment ECUs**

These ECUs integrate:

* **AUTOSAR-based logic** (vehicle network, diagnostics, algorithms)
* **Graphics Software** (animations, warnings, trip computer info)

**Testing Scope Includes**:

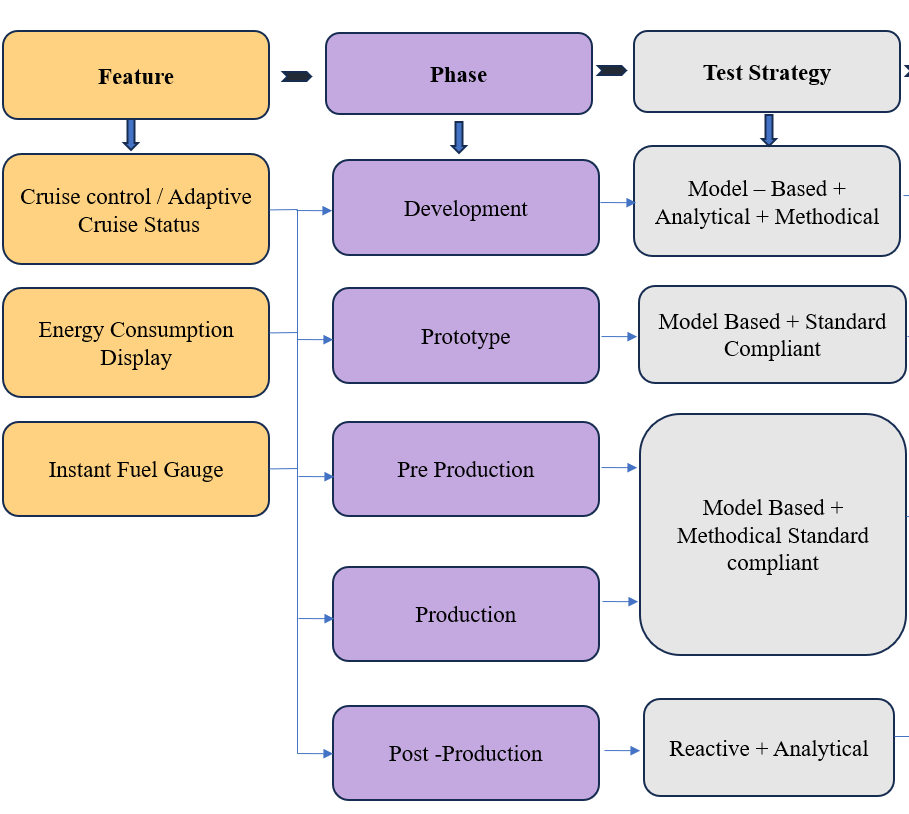
* **Vehicle Network & Algorithm Logic** (AUTOSAR level)
* **Graphical Views & UI Behavior** (Graphics software level)

**Software Release Stages in Multi Powertrain ECUs**

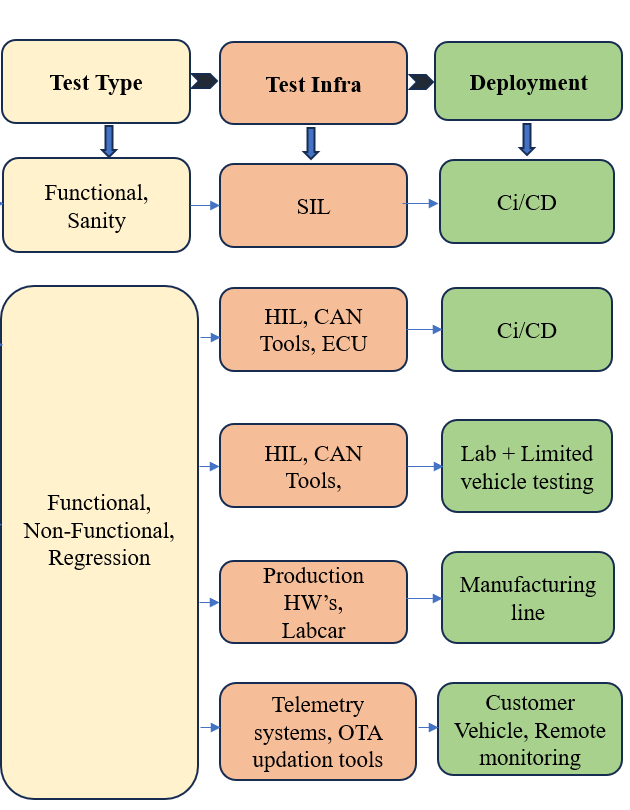
A common test strategy is applied across software levels, with features identified and validated through the following release stages:

1. **Development Release** – Initial feature implementation and unit testing, SIL with Functional Validation
2. **Proto Release** – Early integration and functional testing.
3. **Pre-Production Release** – System-level validation and performance testing.
4. **Production Release** – Final release for manufacturing and deployment.
5. **Post-Production Release** – Maintenance, diagnostics, and updates.

**Defining Test Strategy**:



**Test Methods Followed by Strategy**



Summary/Conclusions

Deriving common test strategy is the challenging task with the consideration of evolving technologies in the automotive domain with respect to Architecture, Protocols, Specification’s, Vehicle Variants etc ,Its achievable with the right test approach , design and mitigation of risks can be key elements to deliver a quality product on time.