Abstract:

Innovation in energy storage and generation system will lead to multiple power train solutions across the vehicle categories in the Automotive 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.

Test Strategy:

1. Master Test plan
2. Test strategy with Design
3. Software Architecture Differences
4. Test Levels
5. Traceability
6. Risks
7. Conclusion

Test strategy:

1. Developing Master Test Plan:
   1. Identify the Scope of the testing of Clusters and CDC’s with respect Multi Power terrain architecture
   2. Differentiate the common and Domain Specific Features to be tested
   3. Identify the common test environment across domains (ICE, BEV, Hybrid, Hydrogen, FCEV)
   4. Identify competent resources and Responsibilities
2. Common Test strategy design for Multi Power Terrain Architecture

Power Trains

Common & Domain Specific KPI's Measurement

Cross Functional Features Cluster & IVI

Domain Specific Features

Common features of Cluster & IVI

Test Section 4

Test Section 3

Test Section 2

Test Section 1

Centralized Service

Oriented architecture

Distributed ECU Arch

Classic AUTOSAR

Zonal Arch

Adaptive AUTOSAR

Distributed ECU Arch

Classic AUTOSAR

Fuel Cell Electric Vehicle (FCEV)

Hydrogen ICE

Battery Electric

Vehicle (BEV)

Combustion Engine (ICE)

Common Test Strategy

(Applicable for SWE4, SWE5, SWE6, SYS5, SYS6)

**Software Architectures Changes:**

| **Feature** |  | **ICE** | **BEV** | **H2 ICE** | **FCEV** |
| --- | --- | --- | --- | --- | --- |
| OTA Updates |  | Limited | Extensive | Emerging | Extensive |
| Middleware |  | Classic AUTOSAR | Adaptive AUTOSAR / POSIX | Classic AUTOSAR | Adaptive AUTOSAR / Linux |
| Cluster Type |  | Analog-Digital | Fully Digital | Hybrid | Fully Digital |
| Data Bus |  | CAN | CAN + Ethernet | CAN | Ethernet |
| Cloud Integration |  | Minimal | High | Low | High |

Test Levels:

Maintain Different Platforms of test sections and execute the common and Domain Specific testcases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Level** | **Focus Area** | **Tools Used** | **Example** | **Multi Terrain arch proposal** |
| Unit | Individual functions | gTest, CppUTest, pytest | Speed logic | Grouping of common logic function testcases and Domain specific logics function testcases |
| Integration | Module interactions | CANoe, Debuggers | GPIO Register level verification, Fault injection tests for UDS, | Maintain Test suits for Different architectures and reuse the |
| Software Qualification | Simulated Software input and output to cluster | CANoe, SIL benches | Diagnostics, Software update  Gear info, State of charge, HMI | 1.Segregate AutoSAR and HMI Features  2. Customize the test scripts  With General functions based on the power trains |
| System | Full cluster behavior | HIL, test benches, Labcar | HVAC , Engine Status check, Communication between ECU’s | High level Generic Test specifications to be deployed based on type of Vehicle level and labcar |
| Acceptance | User & business requirements | OEM tools, manual scripts | UX validation | 1.FOTA, Software update over UDS and USB common use cases to be defined and tested  2. Feature and Variant specific checklists can be maintained to identify the use cases |
| Regression | Stability after changes | CI/CD, automated suites | Re-run tests | Common automated framework with multi test sections can be proposed to reuse test scripts across multi power terrain architecture |

Product Risks:

1. Derive the common test approach for Individual Customer, Software and system specifications with respect to power Trian
2. Maintaining Requirements traceability
3. Revisiting the testcases based on the change requests
4. Upgradation of generic framework based on the test approach for individual projects or complex requirements

Project Risks:

1. Common approach efforts should not impact individual project timelines
2. Strong Competent groups to be identified inside the project team members
3. Additional Efforts to have alignment with internal stakeholders and regular discussions

Conclusion:

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.