With Proper Tools & Training, Anyone can be Successful with Testing

Mazen A Hamdan
Electronics Engineering Manager
Aisin Technical Center of America

Vector Software Testing Symposium
San Jose CA
June 12 2019
Aisin Technical Center of America, Inc. (ATC-A)

ATC-A is a subsidiary of Aisin Seiki, a Tier One, worldwide supplier of automotive components. As the first technical center established by Aisin Seiki outside of Japan, ATC-A leads Aisin's product development for the North, Central and South American marketplace, providing a greater presence to the important markets served by the Aisin Group of companies.

ATC-A is strong locally, providing accelerated design function and faster engineering services to customers of all AISIN products.

**Goals:**

- Expanding local SW design capabilities
- Achieving an automated SW validation testing

<table>
<thead>
<tr>
<th>Milestones</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aisin Technical Center Established</td>
<td>1995</td>
</tr>
<tr>
<td>Move for Office/Laboratory Expansion</td>
<td>2002</td>
</tr>
<tr>
<td>FTTA Inauguration of Ground Established</td>
<td>2005</td>
</tr>
<tr>
<td>ISO Certification</td>
<td>2006</td>
</tr>
<tr>
<td>Aisin Technical Center of America Established</td>
<td>2008</td>
</tr>
<tr>
<td>Aisin Technical Center of America Office and Laboratory Expansion</td>
<td>2014</td>
</tr>
</tbody>
</table>
For a Better Tomorrow

AISIN GROUP

• **System Testing**: for the complete application with its functionality, inter dependency, and communication. It tests the functional and non-functional requirements of the developed application

• **Unit Testing**: plans are developed during module design phase to eliminate bugs at code or unit level

• **Integration testing**: the modules are integrated, and the system is tested on the Architecture design phase to verify the communication of modules among themselves

• **System Testing**: for the complete application with its functionality, inter dependency, and communication. It tests the functional and non-functional requirements of the developed application

• **User Acceptance Testing (UAT)**: performed in a user environment resembling the production environment to verify that the delivered system meets user’s requirement and is ready for use in real world
Resource management:

Centered around optimization and efficiency (doing more with less).
- Optimizing people, materials, and budget efficiency for a successful project
- Efficiently plan out the optimal way to allocate & use resources

Resource allocation: take on project using available resources in the most efficient way, avoiding delays and unnecessary cost

Resource Leveling: pinpointing inefficiently used resources in the team to improve their outcome

Resource forecasting: determining the project’s scope, possible constraints, unforeseen costs, and potential risks. Becoming knowledgeable of the project lifecycle & objectives, and understanding the available resources

Importance
- Avoids unforeseen hiccups
- Prevents burnout
- Provides a safety net
- Builds transparency
- Measures efficiency
Understanding project’s needs to acquire tools and team training needs

Tools Required:
- **(UAT) HIL Tester**: Tool mimicking EOL tester that is equivalent for functional verification to conduct DV/PV testing
- **PVE Tester**: Tool to support calibration and injection of errors during vehicle verification testing

**Identifying Training Needs**: by understanding members’ development needs and pinpointing knowledge and skill gaps

**Choosing the Right Training Methods**: finding the right balance between different ways of learning that will suit everyone

**Corporate focus**: focusing on how employees will learn Versace the tool itself
Assessing Capabilities & Setting Up a Testing Goals Plan

Planning

- Discussing Local TCU development
- Identifying Local development skills

HILS Utilization Plans

- Taking on SW system spec testing and PVE testing
- HIL as EOL equivalent for functional verification for DV/PV testing & to support QA with field claim analysis.
- Start off with basic tasks like programming the HILs then expanding the test coverage

HILs Set up plan

- Maintain up to date tools, skill sets and models

Maintenance Plans

- Monitor new technology changes and ways to skill up members

Skills Improvement Plans

- Progressively expand test coverage to all areas of the software

Software Development Plans
For a Better Tomorrow
AISIN GROUP

Drivetrain Electronics HILS Capability Milestone

<table>
<thead>
<tr>
<th>Capability</th>
<th>System Level Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Validation</td>
<td></td>
</tr>
<tr>
<td>Model Integration</td>
<td></td>
</tr>
<tr>
<td>VT System Configuration</td>
<td></td>
</tr>
<tr>
<td>CANoe Configuration</td>
<td></td>
</tr>
<tr>
<td>HW Preparation</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **HILS Hardware**
  - Vector Hardware
  - Load Box
  - Load Box Harness
  - ECU Harness

- **CANoe Configuration**
  - CANoe Base Configuration
  - CAPL Coding - CAN Communication
  - CAPL Coding - VT System control
  - CAPL Coding - Matlab/CANoe Interface Signals
  - Diagnostic CAN file
  - Vector VT System Configuration
  - HILS IO Spec

- **Model Integration**
  - Plant Model
  - HILS System Validation Plan
  - HILS System Validation Check
  - HILS System Validation Review

- **System Validation**
  - HILS System Validation Plan

- **Testing**
  - Test Spec Creation (Document)
  - Test Sequence Creation (VT testSTudio)
  - Test Sequence Review

- **Report**
  - Report

- **Functional EOL**
  - CANoe verify
  - SW requirement
  - Functional test
  - Functional EOL
  - System monitor
  - MATLAB Simulink
  - Veh. & interface model
  - ECN fault detection
  - SW requirement
  - CANoe verify

- **Fundamental check**
  - Diagnostic check
  - Functional check
  - VT System
  - VT System
  - CAN messages Coding
  - CAN Simulation
  - CAN Simulation
  - VT System

- **I/O Specification**
  - I/O Specification

- **Set up**
  - Set up
**HIL Hardware:**
- Load simulation with measurements
- Real loads with measurements
- CAN communication simulation (testing)
- Current measurements for operation & solenoids
- Vehicle configuration simulation (PRNDL change, activate ignition)
- Signal simulation (PWM) & measurement
- Frequency measurements
- Fault ignition (Battery short ground short)
• Testing:
  • Signal testing (Input & Output);
  • CAN communication testing;
  • Measurements & validation;
  • Fault injection testing;
    • Automated Functional Check;
    • Power/Ground;
    • Signal;
    • Communication;
  • Automated Functional Check;
  • Write CAPL code for testing;
  • vTestStudio (Vector SW) to set up test cases;
  • CANoe to run testing;
  • Long term fault testing;

Drivetrain Electronics PVE Capability
**Drivetrain Electronics PVE Capability**

- **Integration of current production Simulink Model with the HILS System:**
  - Simulate function of entire vehicle;
  - Quickly verify logic and calibration changes to SW function as intended before vehicle testing

- **Model includes:**
  - TCM interphase software, user inputs, and vehicle behavior simulation
  - Custom model for FCA RAM for general vehicle model

- **Create CANoe Configuration**
  - Simulates Accessory, Run, Park, and Drive (Vehicle Input Simulation)

- **Transmission Hardware Simulation**
  - Solenoids/Temperature Sensors/Revolution Sensors/Pressure Switch
Drivetrain Electronics PVE Capability

Production Vehicle Evaluation testing is done after the SW production drawings release & before the SOP to verify that it is functioning properly.

PVE testing takes place at Aisin’s own FT Techno of America, Fowlerville Proving Ground. FTTA boasts multiple test tracks, including a 4,500-foot straightaway, 48,000 square-feet of low and middle μ tiles, a 20-acre dynamic pad, a 3-mile oval track, an ADAS Test Facility and many more.
**Drivetrain Electronics PVE Capability**

- Simply opening a switch on the switch box, breaking that TCM pin’s connection with vehicle & causing error
- Shorting certain pins of the TCM to power or ground, forcing a value that registers an error for a particular circuit
- Using the breakout box to manipulate various sensor values
- Using CANoe to inject CAN communication errors

PVE tests are run to verify all of the DTCs are functioning
ATCA Vector Tools and Usage

**CANoe:**
- CAN Network Analysis and Interface for user interface with the VT System/VT System control;
- PVE testing to inject CAN comm errors

**CANoe.Diva:**
- to generate and execute (with the VT System) OEM Diagnostic test cases

**vTestStudio:**
- Automated test pattern creation and execution tool to program automated test patterns that could be loaded into CANoe and run within the CANoe simulation environment.
ATCA Vector Tools and Usage

**CANdelaStudio:**
- To create CANdela Diagnostic Description (.CDD file) that defines diagnostic requests and responses that are used to communicate with the ECU.

**Vector HIL System:**
- Contains variety of I/O modules used to simulate the transmission hardware & other vehicle inputs to the TCM.
<table>
<thead>
<tr>
<th>Trainings Completed by Members</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CANoe/CANalyzer Basic</strong></td>
</tr>
<tr>
<td>- CAN network and characteristics of the CAN &amp; CAN FD protocols;</td>
</tr>
<tr>
<td>- Analysis functionality, saving &amp; analyzing measurement data, &amp; sending CAN messages;</td>
</tr>
<tr>
<td>- Create &amp; enhance rest bus simulations using the CAPL programming language.</td>
</tr>
<tr>
<td><strong>CANoe/CANalyzer Advanced</strong></td>
</tr>
<tr>
<td>- Setting Up &amp; creating Test Modules and CAPL test Modules;</td>
</tr>
<tr>
<td>- Interface to Measurement Hardware;</td>
</tr>
<tr>
<td>- Introduction to Diagnostics in CANoe;</td>
</tr>
<tr>
<td>- Integration of Diagnostics into CAPL Test Modules;</td>
</tr>
<tr>
<td><strong>LIN</strong></td>
</tr>
<tr>
<td>- Measuring, analyzing, and stimulating a LIN environment using CANoe</td>
</tr>
<tr>
<td><strong>vTestStudio</strong></td>
</tr>
<tr>
<td>- Developing test projects using: integrated test design editors; test tables and test commands project organization, exchange with CANoe, CAPL test modules, test diagrams, information exchange with requirement management systems</td>
</tr>
<tr>
<td><strong>Matlab Simulink</strong></td>
</tr>
<tr>
<td>- MATLAB technical computing environment: data analysis, visualization, modeling, and programming;</td>
</tr>
<tr>
<td>- Simulink system &amp; algorithm modeling and design validation by applying basic modeling techniques and tools to develop Simulink block diagrams.</td>
</tr>
<tr>
<td><strong>C++</strong></td>
</tr>
<tr>
<td>- Learn to Write, build, and code in C++, the popular and pervasive object-oriented programming (OOP) language.</td>
</tr>
</tbody>
</table>
For a Better Tomorrow

AISIN GROUP

DLL Compilation & Incorporating
Incorporating the vehicle model into the CANoe simulation environment
• Simulink models can be added to the CANoe environment if they compiled into a DLL file. The DLL file can then be attached to a network node.

Controlling CAN communication
Determining how to control the cyclic sending of CAN messages to the TCM
• Create and send the network messages in CAPL vs. using the CANoe Interaction Layer (IL). Ultimately the CANoe IL was decided upon due to it requiring less up front work and variety of built in CAPL functions used to control the IL behavior, inject CAN bus faults, etc.

Determining Accuracy
Verifying the accuracy of the vehicle model/simulation system
• CAN logs of actual vehicle data can be played back as inputs to the simulation model to recreate the same user inputs from the truck. When played back the TCM behavior on the HIL system can be compared to the TCM behavior on a vehicle.

Automated testing
Developing automated test cases & incorporating them into the simulation environment
• vTestStudio/CANoe .DiVa/ and CAPL code have all been utilized for automatic test case creation and execution.
• VT test for repetitiveness
• VTest to generate a test report

Obstacles Faced, Tools Used & Lessons Learned

- Know capabilities of engineering tools in order to better understand how to implement them
- Approach issues/road blocks from various angles
- Understand the limits of tool and the information/data it can provide
Concluding Message

- Organizations need to accept that Problems & Technology live in the same world;
- Even when investing in the latest technologies problems will always arise for various factors;
- Before considering what technology to invest in, first ensure the outcomes are clearly identified and identify how to measure the effectiveness of a learning programme;
- Organizations should shift focus to not only owning latest tools but also understanding members’ skill needs;
- After acquiring the right tools learning should be the challenge, not technology

Finally As the Japanese proverb goes

“Vision without action is a daydream.
Action without vision is a nightmare.”

Thank you for your time!