CASE STUDY - ADAS

- ❖ Introduction to iJbridge
- ❖ Client Requirement
- ❖ Project Phases
- ❖ Key Achievements
- ❖ Conclusion

Contact us: services@ijbridge.com

ADAS Development by iJbridge Incorporation

This case study details the development of Lane Keep Assist (LKA) and Traffic Sign Recognition (TSR) systems, showcasing our comprehensive approach from requirement analysis to system deployment. A leading automotive manufacturer from Japan, required the development of LKA and TSR systems integrated into their vehicle's ECU. The project encompassed the following stages:

- Requirement Analysis.
- System Design (High- and Low-Level Design)
- Software Development (Model, Auto Code, System Integration)
- System and Unit Testing (MIL, SIL, and PIL)
- Flashing to ECU Controller.

Project Phases 1/3

Requirement Analysis

Objective: Understand client requirements and translate them into technical specifications.

Activities:

- Conducted stakeholder meetings to gather detailed requirements.
- Developed functional and non-functional requirements documentation.
- Created use case scenarios for LKA and TSR.

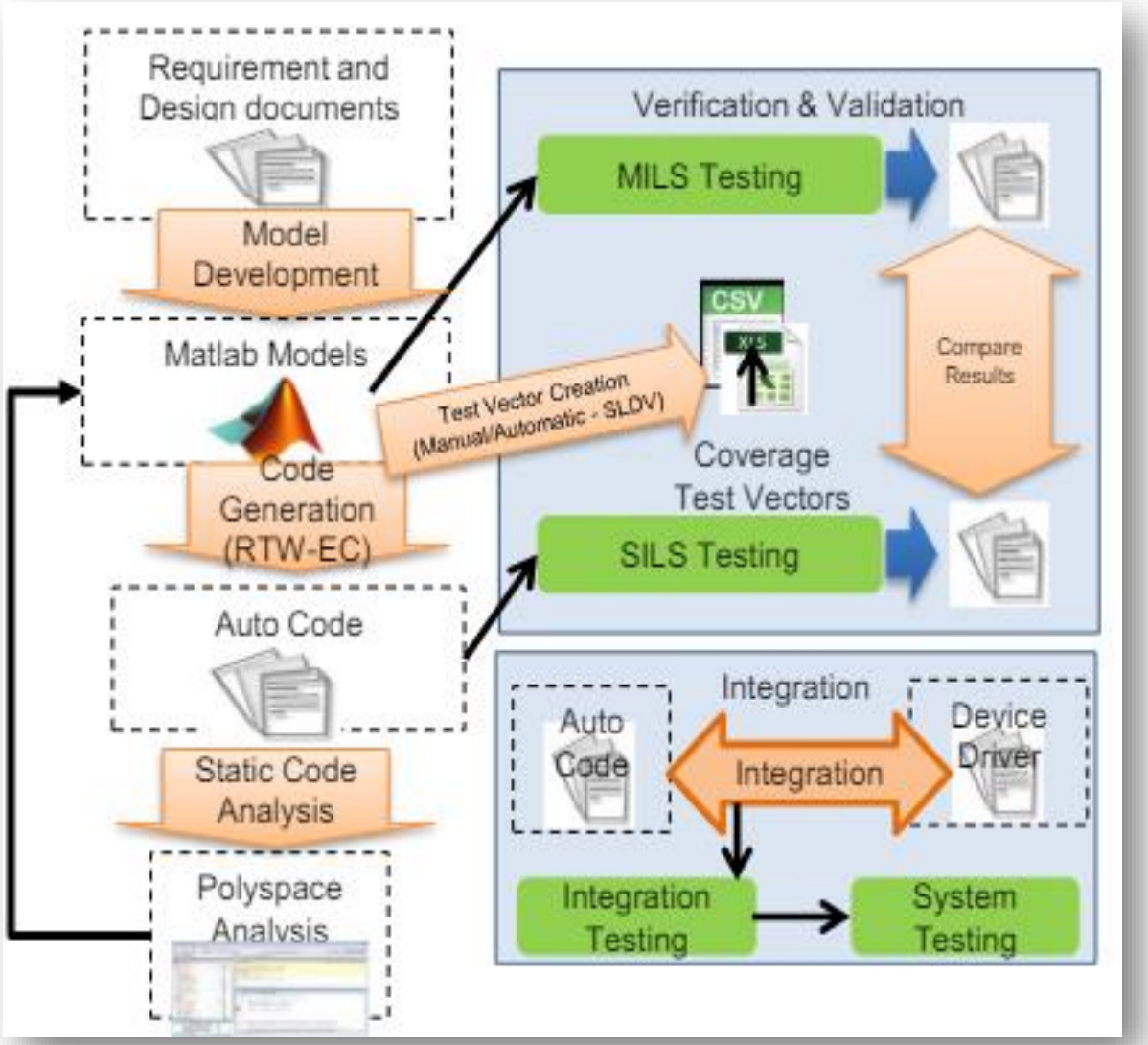
System Design

Objective: Develop a robust design framework for the LKA and TSR systems.

Activities:

- High-Level Design: Defined system architecture and data flow.
- Low-Level Design: Detailed design specifications for software components.
- Tools Used: MATLAB Simulink, State flow.

Project Phase 2/3



Software Development

Objective: Develop and integrate software components for LKA and TSR.

Activities:

- Model Development: Created simulation models using MATLAB Simulink and State flow.
- Auto Code Generation: Used Real-Time Workshop Embedded Coder (RTW-EC) for optimized code generation.
- System Integration: Integrated software components with the ECU hardware.
- Tools Used: MATLAB Simulink/State flow, RTW-EC, Fixed-Point Toolbox, SLDV, V&V tools.

Project Phase 3/3

System and Unit Testing:

Objective: Ensure the developed software meets all functional and safety requirements.

Activities:

- Model-in-the-Loop (MIL) Testing: Verified models against requirements.
- Software-in-the-Loop (SIL) Testing: Tested generated code within a simulation environment.
- Tools Used: Green Hills Compiler, Vector Tools (CANape, CANoe), iSystem Debugger (winIDEA).

Flashing to ECU Controller

Objective: Deploy the developed software to the vehicle's ECU.

Activities:

- Flashing Process: Uploaded the final software onto the ECU.
- Validation: Conducted final validation tests to ensure system functionality in real-world conditions.

Key Achievement

Cost-Effective System Development

- **Methodology:** Employed Model-Based Development (MBD) techniques.
- **Outcome:** Reduced development time and costs by automating code generation and testing processes.

Optimized Code Generation

- **Tool:** Real-Time Workshop Embedded Coder (RTW-EC).
- **Outcome:** Generated efficient, optimized code that enhanced system performance.

Simultaneous MIL/SIL Testing

- **Approach:** Conducted MIL and SIL tests concurrently.
- **Outcome:** Accelerated the testing phase, ensuring rapid identification and resolution of issues.