Name of Testers:

1. Hrushikesh
2. Harshvardhan
3. Asha
4. Shilpa
5. Vinith
6. Jayavardhini

Test Plan for Cruise control System with diagnostic report with Diagnostic error code analysis

Version: 1.0

Created: 05/21/2024

Last Updated: 05/21/2024

Status: Currently working on the "Diagnostic Error Code Analysis"

Table of Contents

1.Introduction

1.1. Purpose

The purpose of this test plan is to outline the testing strategy for the Cruise Control System integrated with a car, ensuring it meets the functional, integration, and performance requirements within the AUTOSAR (AUTomotive Open System ARchitecture) layered architecture.

1.2. Project Overview

The project involves integrating a Cruise Control System into a car's existing architecture. This includes lane departure warning, lane keeping assist, and related diagnostics, all implemented using the AUTOSAR layered architecture.

1.3. Audience

This document is intended for project managers, developers, testers, and stakeholders involved in the development and deployment of the Cruise Control System.

2. Test Strategy

2.1. Test Objectives

- Validate the functional requirements of the Cruise Control System..
- Ensure seamless integration with other vehicle systems.
- Test the system's performance under various conditions.
- Confirm the system's compliance with AUTOSAR standards.

2.2. Test Assumptions

- The hardware components are pre-tested and functioning correctly.
- The test environment mimics the real-world operating conditions of the car.
- Access to all necessary documentation and specifications.

2.3. Test Principles

- Adherence to AUTOSAR standards.
- Comprehensive coverage of all functional and non-functional requirements.
- Systematic and repeatable testing processes.

2.4. Data Approach

- Use of realistic data sets for cruise control scenarios and diagnostics.
- Test data should cover a range of use cases, including edge cases.

2.5. Scope and Levels of Testing

2.5.1. Unit Testing

- Application Layer:
  - Verify individual functions of the cruise control application.
  - Validate user interface responses to inputs.
- Runtime Environment (RTE):
  - Check RTE configuration and data exchange between application and BSW.
  - Validate task scheduling and timing requirements.
- Basic Software (BSW) Modules:
  - Test communication services (COM, PDU Router).
  - Validate memory services (NvM, MemStack).
  - Check diagnostic services (DEM, DCM).
- Microcontroller Abstraction Layer (MCAL):
  - Verify peripheral drivers (CAN, Ethernet).
  - Test microcontroller-specific configurations.

2.5.2. Integration Testing

- Communication Interfaces:
  - Validate CAN and Ethernet communication between cruise control components.
  - Test data exchange between head unit and external devices (e.g., diagnostic tools).
  - Sensor Integration:
  - Verify functionality of speed sensors and radar/lidar for adaptive cruise control.
  - Check integration with vehicle's braking and steering systems.
- User Interface Integration:
  - Verify display and controls for cruise control settings and alerts.
  - Check responsiveness and accuracy of user controls.

2.5.3. System Testing

- System Boot and Shutdown:
  - Verify system start-up and shutdown sequences.
  - Check for error-free initialization and graceful shutdown.
- Functional Scenarios:
  - Test common use cases (e.g., maintaining set speed,adaptive speed regulation).
  - Validate multi-functionality (simultaneous operation of multiple systems).

- Stress and Load Testing:
  - Assess system performance under high load conditions.
  - Validate system behavior under extended operation.
- Failover and Recovery:
  - Test system response to component failures.
  - Validate recovery mechanisms and data integrity.

2.5.4. User Acceptance Testing (UAT)

- Conduct UAT with real users to ensure the system meets their needs and expectations.
- Collect feedback and make necessary adjustments.

2.5.5. Performance Testing

- Measure response times for user inputs and system commands.
- Assess data transfer rates and communication latency.
- Monitor CPU, memory, and network usage.
- Validate system performance under varying resource availability.

2.5.6. Security Testing

- Ensure the lane management system is secure from external threats.
- Test for vulnerabilities and potential breaches.

2.5.7. Regression Testing

- Re-run previous test cases to ensure new updates do not break existing functionalities.
- Validate bug fixes and new features.

2.6. Test Effort Estimate

- Exploratory Testing: 5 days
- Functional Testing: 15 days
- Integration Testing: 10 days
- System Testing: 12 days
- UAT: 8 days
- Performance Testing: 5 days
- Security Testing: 5 days
- Regression Testing: 8 days

3. Test Acceptance Criteria

- All high and medium severity defects are resolved.
- Functional test cases achieve a pass rate of 95%.
- Integration and system test cases achieve a pass rate of 98%.
- Performance benchmarks for response time, speed regulation accuracy, and user interface responsiveness are met.
- Security tests show no vulnerabilities.

4. Test Deliverables

- Test Plan Document
- Test Cases and Test Scripts
- Test Execution Reports
- Defect Logs and Resolution Reports
- UAT Feedback Reports
- Final Test Summary Report

5. Risk Management

- Identify Potential Risks: Hardware failures, software bugs, communication

breakdowns. - Mitigation Strategies:

  - Implement robust error-handling and recovery mechanisms.

  - Conduct thorough pre-testing of individual components.

  - Regularly update and review test plans based on feedback and findings.

  - Allocate additional time for unforeseen issues during test phases.

6. AUTOSAR Layered Architecture Explanation

AUTOSAR (AUTomotive Open System ARchitecture) is a standardized automotive software architecture that provides a framework for developing and integrating complex automotive software systems. It divides the software into several layers to simplify development, enhance modularity, and ensure interoperability.

Application Layer:
- Contains the functional software components that implement specific application functionalities, such as cruise control.

Runtime Environment (RTE):
- Acts as a middleware layer, providing communication services between software components and between software components and the Basic Software (BSW) layer.

Basic Software (BSW):
- Includes standardized software modules that provide services for communication, memory management, diagnostic services, and operating system functionality.

Microcontroller Abstraction Layer (MCAL):
- Provides a hardware abstraction layer that allows the higher layers to be hardware-independent. It includes drivers for peripherals like CAN, Ethernet, and other communication interfaces.

7. BOD, UDS, and DoIP Explanation

BOD (Base Operating Diagnostic):
- A foundational diagnostic service that provides basic diagnostic functionalities, such as error code reading and clearing, and basic system health checks.

UDS (Unified Diagnostic Services):
- A set of standard diagnostic services defined in ISO 14229. UDS provides a comprehensive set of diagnostic and communication services, including fault code management, data streaming, and system reset functions.

DoIP (Diagnostics over Internet Protocol):
- A diagnostic protocol defined in ISO 13400 that allows diagnostic communication over IP networks. DoIP facilitates remote diagnostics and software updates by leveraging standard IP networking technologies.

8. Diagnostics of the Defects

User Interface Diagnostics:
- Error Display:
  - Check Diagnostic Information: Ensure the diagnostic information displayed on the user interface is clear and accurate. Users should be informed of any system errors related to the cruise control, along with their severity.
  - Severity Indication: The interface should differentiate between minor and critical errors, providing appropriate notifications.

- Troubleshooting Guidance:
  - Step-by-Step Solutions:Validate that the user interface offers troubleshooting steps or suggestions for resolving cruise control errors.
  - User-Friendly Design: Ensure the diagnostic interface is intuitive, easy to navigate, and accessible to all users.

Diagnostic Logging and Reporting:
- Log Generation:
  - Detailed Logs:Ensure the cruise control system generates detailed diagnostic logs for all

detected errors and anomalies.
- Log Accuracy and Completeness: Validate that these logs are accurate and comprehensive, covering all relevant data points and events.

- Reporting:
  - External Reporting: Test the system's ability to compile and report diagnostic information to external systems or maintenance personnel.
  - Industry Standards: Verify that the reporting format adheres to industry standards and can be easily interpreted by external diagnostic tools.

Periodic Self-Checks:
- Automated Diagnostics:
  - Self-Check Capabilities:Validate the system's ability to perform periodic self-checks and diagnostics specific to cruise control functions.
  - Comprehensive Coverage: Ensure these self-checks encompass all critical components and functionalities of the cruise control system.

- Alert Generation:
  - Issue Detection Alerts:Test the system's ability to generate alerts for any issues detected during self-checks.
  - Timely Notifications:Ensure timely notifications of potential problems to the user or maintenance system to prevent any lapse in cruise control functionality.

Making Diagnostics Easier with AUTOSAR:
- Standardized Interfaces:
  - Diagnostic Services: AUTOSAR provides standardized diagnostic interfaces and services, which simplifies the integration and interoperability of diagnostic tools within the cruise control system.

- Modularity:
  - Layered Architecture: The layered architecture of AUTOSAR supports modular diagnostics, making it easier to isolate and diagnose specific components or functions within the cruise control system.

- Scalability:
  - Flexible Framework: AUTOSAR's scalable framework supports a wide range of diagnostic capabilities, from basic error reporting to advanced remote diagnostics, ensuring the cruise control system can be effectively monitored and maintained regardless of complexity

9. Diagnostic Error Code Analysis

Error Code Documentation:
- Ensure that all diagnostic error codes are documented comprehensively.
-Provide detailed descriptions for each error code,including possible causes and troubleshooting steps.

Code Validation:
-validate the correctness of error codes generated by the system.
-Ensure that each error code corresponds to a specific fault or issue within the cruise control system.

System Response:
-Test the systems response to various diagnostic error codes.
-Verify that the system takes appropriate actions based on the severity and type of error detected.

Code Mapping:
-Map error codes to specific system components and functions.
-Ensure that the mapping is accurate and helps in pinpointing the root cause of issues.

Historical Analysis:
-Perform historical analysis of error codes to identify recurring issues and trends.
-Use this analysis to improve system reliability and performance over time.

Automated Error Resolution:
-Test the systems ability to automatically resolve certain types of errors.
-Ensure that automated resolution processes are safe and do not compromise system integrity.

10.Diagnostic Error Codes for Cruise Control System

A. Unit Test Cases

1. Application Layer

- Set Speed Function:
- Test Case: Verify that the set speed function correctly maintains the desired speed. - Expected Result: The vehicle maintains the set speed accurately without fluctuations. - Potential Defect: The vehicle speed fluctuates instead of maintaining a constant speed. - Error Code: P0500 (Vehicle Speed Sensor Malfunction).

2. Runtime Environment (RTE):

- Service Call Execution:
  - Test Case: Validate that service calls from the Cruise Control application to other software components execute correctly.
- Expected Result: All service calls execute without errors and within the required time frame. - Potential Defect: Service call delays leading to delayed throttle response. - Error Code: N/A (Performance issue without specific DTC).

3. Basic Software (BSW) Modules:

- Communication Service Validation:
  - Test Case: Test the data exchange between the Cruise Control system and other ECUs via the CAN network.
- Expected Result: Data packets are transmitted and received correctly without loss. - Potential Defect: Data packets are lost or corrupted during transmission.
- Error Code: U0100 (Lost Communication with ECM/PCM).

B. Integration Test Cases

1. User Interface Integration:
   - Touchscreen Functionality:
   - Test Case: Verify the touchscreen interface responds correctly to inputs for setting and adjusting the cruise control speed.

- Expected Result: The touchscreen responds accurately and swiftly to all inputs. - Potential Defect: Touchscreen unresponsive or inputs are not correctly processed. - Error Code: B1503 (Touch Screen Circuit Malfunction).

2. Sensor Integration:

- Brake Switch Input:
  - Test Case: Test the system's response to brake switch input by pressing the brake pedal while cruise control is active.
- Expected Result: Cruise Control disengages immediately upon brake application. - Potential Defect: Cruise Control does not disengage or delays in disengaging. - Error Code: P0571 (Brake Switch "A" Circuit Malfunction).

C. System Test Cases

1. System Boot and Shutdown:
   - Startup Sequence:
   - Test Case: Verify that the Cruise Control system initializes correctly during vehicle startup. - Expected Result: The system initializes without errors and is ready for use.
   - Potential Defect: System fails to initialize or shows error messages.
   - Error Code: P0650 (Malfunction Indicator Lamp (MIL) Control Circuit).

2. Functional Scenarios:
   - Maintaining Speed Uphill:
   - Test Case: Test the vehicle's ability to maintain the set speed while driving uphill. - Expected Result: The vehicle maintains the set speed with appropriate throttle adjustments. - Potential Defect: Vehicle slows down or speeds up unexpectedly.
   - *Error Code:* P0500 (Vehicle Speed Sensor Malfunction).

D. Regression Test Cases

1. Re-run Previous Test Cases:
   - Existing Feature Validation:
   - Test Case: Re-run all previous test cases to ensure no existing functionalities are broken by new updates.
   - Expected Result: All previously passed test cases should pass again.
   - Potential Defect: Previously fixed issues reappear.

- Error Code: Varies based on specific defect reoccurrence.

E. Performance Test Cases

1. Latency and Throughput:
 - Response Time:
 - Test Case: Measure the response time for the system to engage or disengage Cruise Control. - Expected Result: The system should engage or disengage within the specified time limits. - Potential Defect: Delayed engagement or disengagement.
 - Error Code: N/A (Performance issue without specific DTC).

F. Compliance Test Cases

1. AUTOSAR Standards:
 - Module Configuration:
 - Test Case: Validate that all software components conform to AUTOSAR standards. - Expected Result: All components are compliant and properly configured.
 - Potential Defect: Non-compliance with AUTOSAR standards.
 - Error Code: N/A (Compliance issue without specific DTC).

10.1.Potential Defects and Error Codes

1. P0500: Vehicle Speed Sensor Malfunction
 - Description: This code indicates a malfunction in the vehicle speed sensor, which can affect the ability to maintain the set speed.

2. P0564: Cruise Control Multi-Function Input "A" Circuit
 Description: This code indicates an issue with the multi-function input circuit for cruise control, affecting input recognition.

3. P0571: Brake Switch "A" Circuit Malfunction
 Description: This code points to a malfunction in the brake switch circuit, which is crucial for disengaging the cruise control.

4. P0581: Cruise Control Multi-Function Input "A" Circuit High
 Description: This code indicates a high voltage issue in the multi-function input circuit for cruise control, affecting control inputs.

5. U0100: Lost Communication with ECM/PCM
 Description: This code signifies a loss of communication with the engine control module or powertrain control module, which can disrupt cruise control operation.

6. B1503 Touch Screen Circuit Malfunction
 Description: This code indicates a malfunction in the touch screen circuit, affecting user interface

functionality.

7. P0650: Malfunction Indicator Lamp (MIL) Control Circuit
   Description: This code points to an issue with the MIL control circuit, which can indicate broader system failures.

11. QA Perspective on Handling DTCs

1. Detection and Logging:
- Ensure accurate fault detection and logging.

2. Validation:
- Validate DTC accuracy and consistency through testing.

3. Reporting:
- Ensure clear, standardized diagnostic reports.

4. User Interface Testing:
- Test UI for accurate error display and user-friendly troubleshooting.

5. Resolution Process:
- Confirm easy access and interpretation of DTCs by maintenance staff.

6. Continuous Improvement:
- Use feedback to refine diagnostic systems and enhance reliability.

THANK YOU