Name of Testers:

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

Infotainment Device with HMI and AUTOSAR Layered Architecture

Version: 1.0

Created: 05/21/2024

Last Updated: 05/21/2024

Status: Currently working on the “HMI Integration”

Table of Contents

1. Introduction

The infotainment system is a critical component of modern vehicles, providing drivers and passengers with a range of features such as media playback, navigation, connectivity, and remote diagnostics. To ensure the reliability, functionality, and compliance of this system, we have developed a comprehensive test plan. This plan leverages the AUTOSAR layered architecture and integrates advanced Human Machine Interface (HMI) capabilities, aiming to deliver a seamless user experience while adhering to industry standards. The purpose of this document is to outline the test strategy, scope, and methodologies we will employ to validate the system’s performance and integration within the vehicle. It is intended for project managers, developers, testers, and other stakeholders involved in the development and deployment of the infotainment system.

1.1. Purpose

The purpose of this document is to outline the test plan for the infotainment device integrated with a Human Machine Interface (HMI) using the AUTOSAR layered architecture. The plan aims to ensure the system's functionality, reliability, and compliance with industry standards.

1.2. Project Overview

This project involves developing and integrating an infotainment system with advanced HMI capabilities within a vehicle, utilizing the AUTOSAR layered architecture for modular and scalable design. The system will offer features such as media playback, navigation, connectivity, and remote diagnostics.

1.3. Audience

The intended audience for this document includes project managers, developers, testers, and stakeholders involved in the infotainment system's development and deployment.

2. Test Strategy

2.1. Test Objectives

The main objectives are to validate the infotainment system's functional requirements, ensure seamless integration with vehicle systems, test performance under various conditions, and confirm compliance with AUTOSAR standards.

2.2. Test Assumptions

It is assumed that hardware components are pre   tested and functional, the test environment simulates real   world conditions, and all necessary documentation is available.

2.3. Test Principles

In testing an infotainment device with HMI and AUTOSAR layered architecture, the following principles guide the process:

1. Standards Compliance: Adhere to AUTOSAR standards for consistency and interoperability.
2. Comprehensive Coverage: Test all system functionalities thoroughly.
3. Systematic Processes: Follow structured testing methods for reliability.
4. Realistic Scenarios: Use real world scenarios for accurate validation.
5. Early Defect Detection: Identify defects early to minimize impact.
6. Risk Based Approach: Prioritize testing based on potential impact.
7. Continuous Improvement: Adapt and refine testing processes over time.
8. Collaboration: Foster teamwork between stakeholders for alignment.
9. Compliance Validation: Ensure adherence to industry standards.
10. User Centric Focus: Prioritize testing from the user's perspective.

2.4. Data Approach

In ensuring the effectiveness of an infotainment device with HMI and AUTOSAR layered architecture:

1. Realistic Data Sets: Use authentic data sets reflecting diverse usage scenarios.
2. Use Case Coverage: Include various use cases, even edge cases, for comprehensive validation.
3. Data Quality Assurance: Implement checks for data accuracy, consistency, and completeness.
4. Data Management: Establish efficient processes for storage, retrieval, and manipulation.
5. Data Integration: Ensure seamless data exchange across system layers and components.
6. Data Security: Employ measures like encryption and access controls to protect sensitive information.
7. Scalability and Performance: Design data approaches that scale and optimize system performance effectively.

2.5. Scope and Levels of Testing

Unit Testing: Testing individual software components.
Integration Testing: Ensuring different system components work together.
System Testing: Verifying the entire system's functionality.
Regression Testing: Ensuring new changes do not affect existing functionalities.
Performance Testing: Assessing system performance under various conditions.
Compliance Testing: Ensuring adherence to AUTOSAR and automotive standards.

2.6. Test Effort Estimate

Estimating the time and resources required for each phase of testing, including preparation, execution, and analysis.

3. Test Acceptance Criteria

Define the criteria for test acceptance, including specific performance metrics, error thresholds, and compliance requirements that must be met for the system to be considered ready for deployment.

4. Test Deliverables

List of deliverables expected from the testing process, including test plans, test cases, test scripts, test results, defect logs, and test summary reports.

5. Risk Management

Identify potential risks related to the testing process, such as hardware failures, software bugs, integration issues, and develop mitigation strategies to address these risks.

6. BOD, UDS, and DoIP Explanation

In the context of an infotainment device with HMI and AUTOSAR layered architecture:

Basic Object Dictionary (BOD): Standardizes data definitions across software modules, ensuring seamless communication and integration.

Unified Diagnostic Services (UDS): Provides standardized protocols for diagnostic communication, enabling technicians to diagnose issues and perform maintenance tasks efficiently.

Diagnostics over IP (DoIP): Utilizes IP based networks for remote diagnostics and updates, enhancing connectivity and enabling faster data transfer rates.

Integration of these components ensures a robust, scalable, and efficient platform for the infotainment system, meeting industry standards and user expectations.

7. Diagnostics of the Defects

In diagnosing defects for an infotainment device with HMI and AUTOSAR layered architecture:

Identification: Employ testing techniques to spot defects in the system and monitor behavior for anomalies.

Recording and Analysis: Log defects with detailed descriptions and analyze root causes, collaborating with development teams.

Prioritization: Rank defects by severity and urgency, focusing on critical issues first.

Resolution: Apply corrective actions like code fixes and configuration changes, testing thoroughly for effectiveness.

Verification and Closure: Confirm fixes, close resolved defects, and document lessons learned for future improvements.

8. Diagnostic Error Code Analysis

In the context of an infotainment device with HMI and AUTOSAR layered architecture, diagnostic error code analysis plays a crucial role in troubleshooting and maintaining system reliability. Here's an overview of the process:

1. Error Code Identification:
   Identify diagnostic error codes generated by the system, including those related to HMI functionality, communication protocols, and system integration.

2. Code Categorization:
   Categorize error codes based on their source and impact, such as HMI related errors, communication errors (e.g., CAN, LIN), or AUTOSAR module failures.

3. Interpretation:
   Interpret the meaning of each error code, considering its description, severity level, and potential implications for system operation.

4. Root Cause Analysis:
   Analyze the root cause of each error code, investigating underlying issues such as software bugs, hardware malfunctions, or configuration errors.

5. Troubleshooting Steps:
   Define troubleshooting steps for each error code, providing guidance on how to diagnose and resolve the underlying issue effectively.

6. Resolution and Validation:
   Implement corrective actions to address identified root causes, such as software updates, hardware

replacements, or configuration adjustments.

Validate the effectiveness of the resolution measures through testing and verification, ensuring that the error codes are no longer triggered under normal operating conditions.

7. Documentation and Reporting:

   Document the analysis process, including findings, actions taken, and outcomes, in a centralized repository or defect tracking system.

   Generate reports summarizing diagnostic error code analysis results, highlighting trends, recurring issues, and areas for improvement.

9. Diagnostic Error codes for HMI

When developing and testing the Human Machine Interface (HMI) for the infotainment system, it's crucial to identify and handle diagnostic error codes effectively. These codes help in troubleshooting and ensuring the smooth operation of the HMI. Below is a list of common diagnostic error codes for HMI along with their descriptions, possible causes, and recommended actions:

1. HMI_ERR_001: Touchscreen Calibration Error

   Description: The touchscreen is not responding accurately to user inputs.

   Possible Causes: Calibration data corrupted, hardware malfunction, software glitch.

   Recommended Actions: Recalibrate the touchscreen, check and replace hardware if necessary, update or reinstall the HMI software.

2. HMI_ERR_002: Display Rendering Error

   Description: The display is not rendering UI elements correctly.

   Possible Causes: Software bug, insufficient memory, faulty display hardware.

   Recommended Actions: Update HMI software, free up memory resources, check and replace display hardware if needed.

3. HMI_ERR_003: Voice Recognition Failure

   Description: The voice recognition feature is not accurately processing commands.

   Possible Causes: Microphone issue, background noise, software error.

   Recommended Actions: Check and replace the microphone, improve noise filtering, update the voice recognition software.

4. HMI_ERR_004: Button Input Error

Description: Physical or virtual buttons are not registering inputs correctly.
Possible Causes: Hardware defect, software malfunction, improper configuration.
Recommended Actions: Inspect and replace faulty buttons, update HMI software, verify button configuration settings.

5. HMI_ERR_005: Connectivity Issue
   Description: The HMI is unable to connect to external devices (e.g., smartphones).
   Possible Causes: Bluetooth/Wi Fi module malfunction, software bug, interference.
   Recommended Actions: Restart the connectivity module, update connectivity drivers, reduce interference sources.

6. HMI_ERR_006: Navigation Display Error
   Description: Navigation maps or routes are not displayed correctly.
   Possible Causes: Corrupted map data, GPS signal issues, software error.
   Recommended Actions: Update map data, check GPS hardware, reinstall navigation software.

7. HMI_ERR_007: Audio Output Error
   Description: Audio playback is not functioning or is distorted.
   Possible Causes: Speaker or amplifier issues, software bug, incorrect settings.
   Recommended Actions: Inspect and replace audio hardware, update audio drivers, verify audio settings.

8. HMI_ERR_008: System Lag or Freeze
   Description: The HMI system is experiencing significant lag or is freezing.
   Possible Causes: Overloaded CPU, insufficient RAM, software inefficiency.
   Recommended Actions: Optimize software performance, upgrade hardware resources, reboot the system.

9. HMI_ERR_009: Firmware Update Failure
   Description: Firmware updates for the HMI are failing to install.
   Possible Causes: Corrupted update file, interrupted update process, hardware compatibility issues.
   Recommended Actions: Verify the integrity of the update file, ensure stable power supply during updates, check hardware compatibility.

10. HMI_ERR_010: Diagnostic Communication Error
    Description: The HMI is unable to communicate with diagnostic tools.
    Possible Causes: Faulty communication ports, software incompatibility, protocol mismatch.
    Recommended Actions: Check and replace communication ports, update diagnostic software, ensure

correct diagnostic protocols are used.

By identifying and addressing these diagnostic error codes, we can ensure the HMI of the infotainment system operates smoothly, providing a reliable and user friendly experience for the driver and passengers.

10. QA Perspective on Handling DTCs

In managing Diagnostic Trouble Codes (DTCs) for an infotainment device with HMI and AUTOSAR layered architecture, QA focuses on:

1. Identification and Monitoring: Actively monitoring the system to detect and capture DTCs during testing and realworld usage.

2. Severity Assessment: Prioritizing DTCs based on severity to focus resolution efforts effectively.

3. Root Cause Analysis: Investigating the underlying issues triggering DTCs and collaborating with teams to address them.

4. Validation of Resolution: Testing and validating corrective actions to ensure successful DTC resolution without introducing new issues.

5. Documentation and Reporting: Maintaining detailed records of DTC occurrences, resolutions, and validation activities for stakeholders' insight.

6. Continuous Improvement: Participating in post mortem meetings to analyze processes, identify areas for improvement, and implement preventive measures.

THANK YOU