Android Testing in Car – Example Based Presentation Document

---

1 Definition of Android Testing in Car

Android testing in cars is the process of validating applications and system behavior running on **Android Automotive OS (AAOS)** inside a vehicle to ensure safety, reliability, performance, and distraction-free driving.
It focuses on testing how Android apps behave when connected to real vehicle signals like **speed, gear, ignition, and steering**.

Unlike mobile testing, car testing is **safety-critical** because any failure can distract the driver or affect driving.

---

2 Why Android Car Testing is Important

Android Automotive directly interacts with vehicle hardware, so testing is required to ensure:

- Driver safety
- Distraction-free UI
- Correct vehicle signal handling
- Legal and OEM compliance
- Smooth infotainment experience

Example (same scenario):
If a call UI blocks the reverse camera, it can cause an accident. So this must be tested carefully.

---

3 Example Scenario: Bluetooth Call While Driving

Real-life Situation

A driver is driving at **60 km/h** and receives an incoming call through Bluetooth.

Android Automotive Behavior

- ECU sends speed signal via CAN
- Vehicle HAL (VHAL) reads speed
- Android system knows the car is moving
- Call notification appears with minimal UI
- Audio is routed to car speakers
- Only **Accept / Reject** buttons are shown
- Keyboard and full screen popup are disabled
- Navigation screen is not disturbed

This behavior is controlled by Android Automotive safety rules.

---

4 What is Tested in This Scenario

In Android car testing, testers validate:

Functional Testing

- Bluetooth connects automatically
- Call can be accepted and ended
- Audio and microphone work correctly

Safety Testing

- UI is minimal while driving
- No keyboard or text input allowed
- No full-screen distractions

Integration Testing

- Speed signal is received correctly from ECU
- VHAL updates Android system
- App responds correctly to speed

Performance Testing

- Call UI opens instantly
- No lag in audio routing
- Navigation continues smoothly

---

5 Automation & Tools Used

Automation is used to validate this scenario repeatedly.

Automation Steps:

1. Simulate speed = 60 km/h (CAN tool)
2. Trigger incoming call
3. Verify minimal call UI is displayed
4. Verify audio routing to speakers
5. End call and verify screen return

Tools Used:

- ADB
- UI Automator / Espresso
- Python + Pytest
- CANoe / CANalyzer
- HIL / SIL setups

---

6 Conclusion

This example shows how Android testing in cars ensures **safe handling of phone calls while driving**, proper Android–vehicle integration, and a smooth driver experience.
Testing Android Automotive apps is essential for building reliable and safe vehicles.