

# INSIGHTS

Automatisierte Systemtests an einem Kommunikationsmodul mit  
Funktion zum Flashen Over-The-Air in der Automobilindustrie

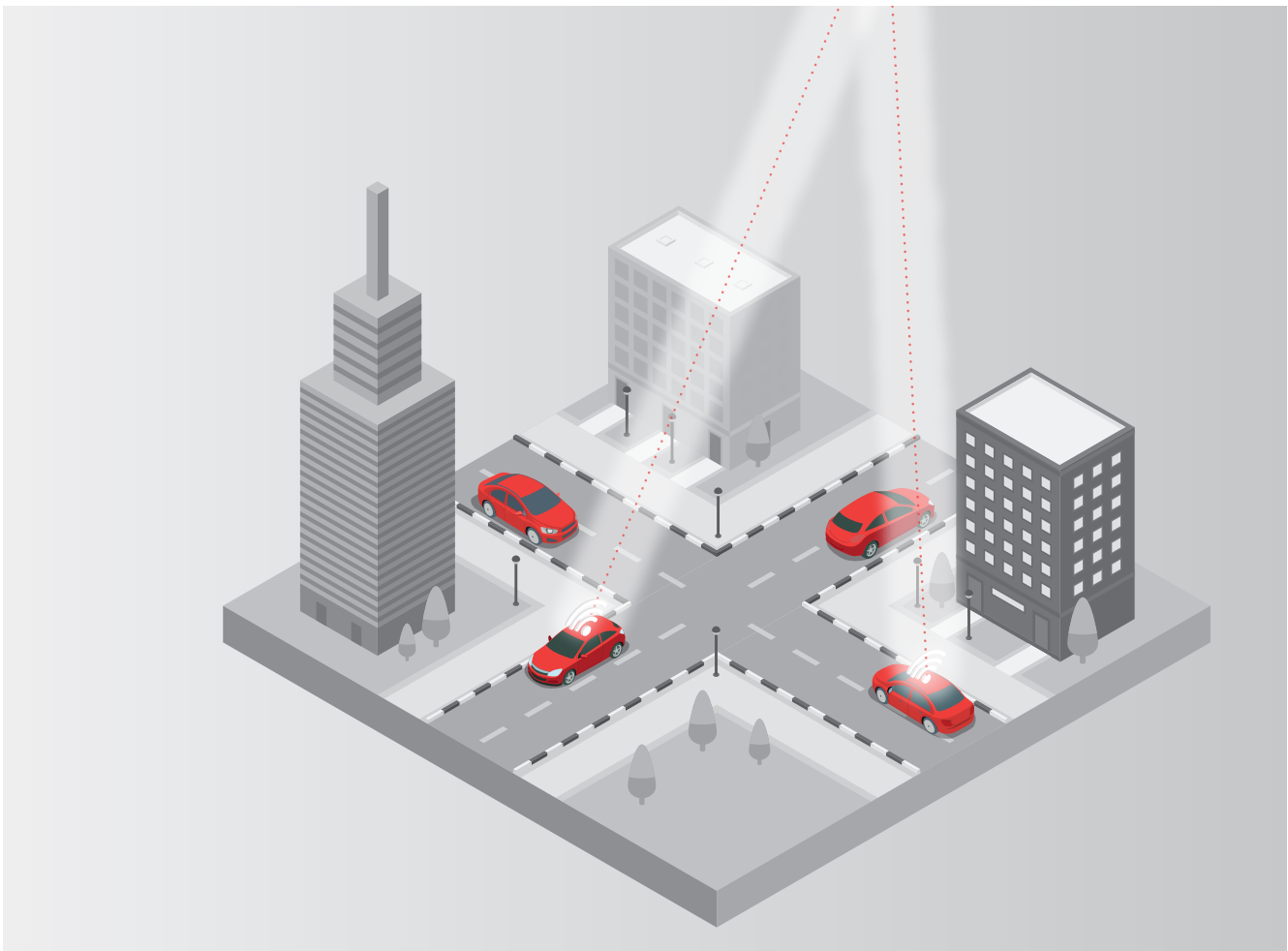
DA ENGINEERING  
VINCI

Autor: Markus Forner, Senior Software Engineer bei Da Vinci Engineering

## Automatisierte Systemtests an einem Kommunikationsmodul mit Funktion zum Flashen Over-The-Air in der Automobilindustrie

Unsere Expertinnen und Experten überprüfen im Auftrag verschiedener OEM die Software-Funktionen künftiger Fahrzeugreihen auf ihre Funktionalität. So führt Da Vinci Engineering auch Automatisierte Systemtests an einem Kommunikations-Modul mit Funktion zum Flashen Over-The-Air in der Automobilindustrie durch. Durch eine gewährleistete ordnungsgemäße Funktion der Telematics Control Unit kann sichergestellt werden, dass die Fahrzeuge auf dem neusten Stand gehalten werden und sich fortlaufend mit der umgebenden Infrastruktur vernetzen.

Inzwischen ist die Baureihe, in der die Telematics Control Unit eingesetzt wurde, in Serie gegangen. Deshalb möchten wir für alle Interessierten Hintergründe und Details unseres erfolgreich abgeschlossenen Projekts beleuchten.



## UMFELD

**Device under Test war eine „Telematics Control Unit“ (TCU) mit folgenden Funktionen:**

- Automatisch oder manuell auslösbarer Pannenhilfe- und Notruf
- Sammlung und Übermittlung von Telemetriedaten
- Remote Diagnose
- Remote Flashing von Steuergeräten inklusive des Self-Updates der TCU
- Assistenzfunktionen (Remote Fensteröffnung, Remote Klimakonditionierung, ...)
- Internet im Fahrzeug
- GPS im Fahrzeug
- Zusatzdienste (Wetter, Parkplätze, Tank- und Ladesäulen-Finder, Live-Traffic, ...)

**Technisch basiert das System auf:**

- TCU mit Ethernet-Schnittstelle ins Fahrzeug
- Zwei-Prozessor-System
- LTE-Modul
- Sichere Daten- und Diagnose-Kommunikation im Fahrzeug
- Update-Funktionalität auf Basis der OMA (Open Mobile Alliance) DM (Device-Management) Spezifikation

## AUFGABENSTELLUNG

Da Vinci Engineering wurde von einem OEM beauftragt, eine Testspezifikation und ein Testkonzept

---

aus den insgesamt über  
**3.000**  
einzelnen Anforderungen zu erarbeiten.

---

Infolgedessen waren die entwickelten Automatisierte Systemtests für die laut Release-Schedule gelieferten Softwarestände durchzuführen. Dabei waren die Ergebnisse inklusive der Fehler zu dokumentieren, deren Abstellung zu tracken und dem Kunden zu berichten.

**Getestet wurden die Features für**

- die Remote Diagnose, welche den Ablauf von Scripten für die Diagnose der im Fahrzeug befindlichen Steuergeräte ermöglicht;
- das Flashen Over-The-Air, das die Möglichkeit zum Flashen und Codieren von Steuergeräte-Software im Fahrzeug im Kundenbetrieb bietet.

Zwei Generationen des Moduls sollten parallel getestet werden; teilweise unterschieden sich einzelne Anforderungen, fielen weg oder kamen neu dazu. Darüber hinaus sollten jeweils zwei Ausbaustufen (Basisversion, erweiterte Version) der Module parallel getestet werden.

Im Rahmen des Feature-Rollout- und Projekt-Plans erhielten wir in der Regel

---

**alle fünf Wochen, zu Hoch-Zeiten sogar jede Woche,**  
eine neue Version der Software.

---

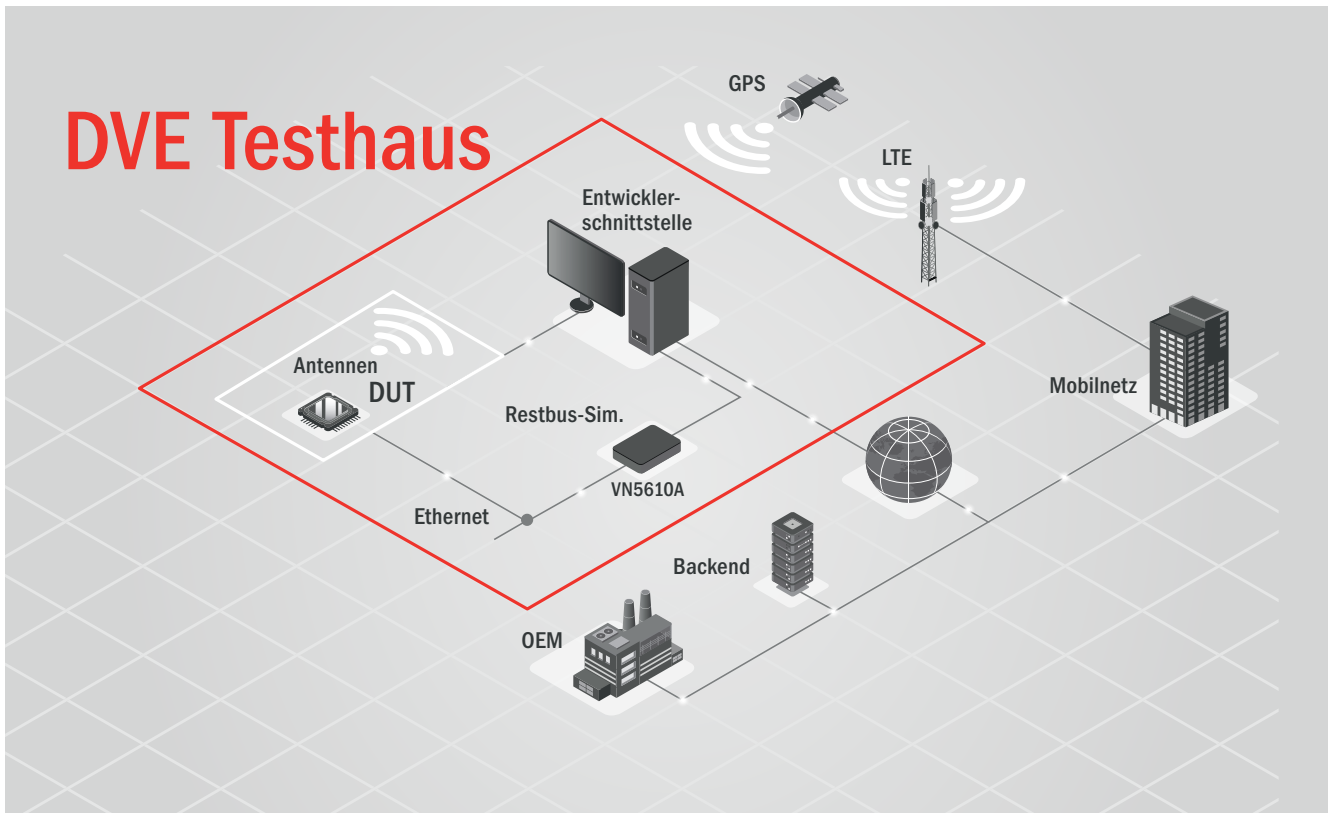
Die Auffälligkeiten wurden im Product Lifecycle Management-System des OEM dokumentiert und dem Kunden nach jedem Testlauf statistisch reportet. Zur Validierung der Fehlerabstimmung hat Da Vinci Engineering Progressionstests durchgeführt.

## KONZEPT

Testcases wurden zunächst in einer Testspezifikation erarbeitet und dokumentiert. Dann wurden sie manuell durchgeführt und ggf. angepasst. Danach wurden sie programmiert und validiert, und, falls nötig, nochmals angepasst. Sämtliche Fahrzeugfunktionen wurden mittels Restbussimulation, welche durch CANoe Model Generation Wizard generiert wurde, simuliert. Dabei waren eigene Anpassungen am generierten Code notwendig, um beispielsweise einzelne Knoten komplett abschalten zu können.

Dann erfolgte die Kontaktierung der seriellen Schnittstelle am Prozessor für das Prozessor-Logging und den Zugriff auf die Engineering-Schnittstelle. Für die Übermittlung von Steuerinformationen und den Paket-Download wird eine Backend-Verbindung über das GSM-Netz hergestellt. Im Anschluss wurden die Diagnose-Responses von „flashbaren“ ECUs im Fahrzeug-Netzwerk simuliert. Die SQL-basierte Datenbank nimmt die Ergebnisse der Testcases auf und dient als Basis für alle Auswertungen und Statistiken. Die selbst entwickelte Automatisierung auf Basis von Vector CANoe, vTestStudio und C# ermöglicht den Betrieb und die Steuerung der gesamten Testumgebung inklusive

- der CANoe Restbussimulation
- des Verhaltens des Backends über REST-requests
- der Diagnose am DUT
- der Engineering-Schnittstelle am DUT
- des Testablaufs
- der Dokumentation der Testergebnisse inkl. der zugehörigen Logs



## TESTDURCHFÜHRUNG

Zur Testdurchführung wurde das Systems für einen Test Case vorbereitet (Preconditions setzen). Es folgte das Setzen eines Reizes an das Gerät (Action) und das erwartete Verhalten (Expected Result) wurde ermittelt. Das tatsächlich beobachtete Verhalten wurde dokumentiert und anschließend ausgewertet. Die Ergebnisse von fehlerfrei durchlaufenen Test Cases (Passed) wurden automatisiert in die Datenbank eingetragen. Test Cases mit Fehlern (Failed) wurden ausgewiesen und manuell nochmals getestet, um sicherzustellen, dass das Ergebnis valide ist.

## HÜRDEN

Die verwendete CANoe-Version in Verbindung mit der RBS auf Basis adaptive AUTOSAR führte zu einer intensiven Nutzung der Ressourcen des Test-PCs bis hin zum Abbruch der Simulation.

**Diese Hürde konnte durch folgende zwei Maßnahmen erfolgreich überwunden werden:**

- a. Deaktivierung der Nutzung von SOME-IP-Services simulierter Knoten durch andere simulierte Knoten dank eines von Vector bereitgestellten Patches im Model Generation Wizard
- b. Entfernen der nicht benötigten Panels nach der Generierung der RBS

## BENEFITS & POTENTIALE

Die Automatisierung von Tests hat viele Vorteile, stößt aber manchmal auch an ihre Grenzen. Die Dynamik in der SW-Entwicklung durch den Lieferanten ist hier ein nicht zu unterschätzender Faktor. Wenn die Inhalte von Zeilen aus dem Prozessor-Log zwischen zwei Software-Ständen geändert werden, erkennt ein menschlicher Tester die Log-line unter Umständen noch korrekt, das automatisierte System aber schon nicht mehr und läuft auf einen Fehler. In manchen Fällen führt eine Verbesserung der HW-Performance zwischen zwei Musterständen zu einem veränderten Zeitverhalten, sodass mühsam erarbeitete Timeout-Kriterien nicht mehr zutreffen und angepasst werden müssen. Auch die Bedienung der Testkonfiguration, bestehend aus CANoe, der Automatisierung und der unterstützenden Tools, ist nicht komplett trivial.

Die Vorteile bei Automatisierungen von Tests wie Wiederholbarkeit, Zeitersparnis und Verbesserung der Test-Tiefe und Test-Qualität überwiegen jedoch klar. Hierbei bietet unsere Lösung noch Potential für eine Skalierung auf ein größeres Umfeld in naher Zukunft.