

Was verstehen wir unter Test? Abstraktionsebenen, Begriffe und Definitionen

Dr.-Ing. **Gerd Baumann**, FKFS, Stuttgart

1. AutoTest; Fachkonferenz zum Thema Test und Diagnose in der Automobilentwicklung. Stuttgart, 26./26.10.2006.

Zusammenfassung

Beim Test von elektronischen Steuergeräten und Embedded Software für Kraftfahrzeuge wird eine Vielzahl von Methoden eingesetzt, die sich in den einzelnen Entwicklungsphasen der Elektronik für ein neues Kraftfahrzeug signifikant unterscheiden. Heute sind diese Verfahren sowie die eingesetzten Geräte und Software-Werkzeuge häufig nicht miteinander kompatibel, so dass ein Austausch von Testabläufen zwischen den Fachabteilungen beim Automobilhersteller sowie mit den Zulieferern schwierig oder sogar unmöglich ist.

Hinzu kommt, dass die Kraftfahrzeugelektronik ein multidisziplinäres Arbeitsgebiet ist und dass sich die Sichtweisen und Begriffswelten der beteiligten Fachrichtungen (Maschinenbau, Fahrzeugtechnik, Elektrotechnik und Informationstechnik) deutlich unterscheiden.

Das Ziel des vorliegenden Beitrags ist es, eine Übersicht zur aktuellen Situation beim Test von Automobilelektronik und -Software zu geben und wichtige Begriffe dieses Arbeitsgebiets zu definieren und beispielhaft zu erläutern.

Abstract

Within the test of electronic control units and embedded software for motor vehicles a wide variety of methods is used. These methods differ significantly in the individual development phases of the electronics for a new vehicle. Today these methods as well as the assigned testing hardware and software tools are frequently not compatible with one another. Therefore the exchange of test descriptions and test results between the divisions of a car manufacturer which are involved in the testing process is typically difficult or even impossible.

The development of vehicle electronics is a multidisciplinary working area. The problem view and the nomenclature which is used by the involved experts (mechanical, electrical and software engineers) differ substantially.

This paper intends gives an overview of the current situation in automotive electronics and software testing and defines and exemplarily describes important terms of this field of activity.

1 Einführung

1.1 Historische Entwicklung

Der Begriff „Test“ hat eine lange Entwicklungsgeschichte. Abgeleitet aus dem lateinischen Wort *testa* (irdenes Geschirr, Topf, Schale) wurde im Mittelalter mit *test* ein kleiner Schmelztiegel bezeichnet, in dem Metalle durch Schmelzen auf ihre Qualität und Reinheit geprüft wurden, also ein frühes Prüfgerät. Seit dem 16. Jahrhundert wird der Begriff *Test* in vielen europäischen Sprachen in seiner heutigen, allgemeinen Bedeutung gebraucht [1]. Der *Duden* erklärt *Test* mit *Probe, Prüfung, Experiment, Untersuchung*.

Wir haben es also mit einem Begriff zu tun, der international bekannt ist. Trotzdem soll der Begriff *Test* im Folgenden genauer definiert werden.

1.2 Ein einfaches Beispiel



Bild 1: Links: Mittelalterlicher Alchemist; rechts: Herzog Wilhelm V. von Bayern

Für dieses Beispiel ziehen wir nochmals die mittelalterliche Bedeutung des Begriffs *Test* (Schmelztiegel) heran. Überliefert ist, dass der Alchemist Marco Bragadino im 16. Jahrhundert dem bayerischen Herzog Wilhelm V. (Bild 1) erklärte, aus unedlen Werkstoffen Gold herstellen zu können. Der Herzog, unter Geldsorgen leidend, war begeistert und verhalf dem „Goldmacher“ zu einem Leben in Saus und Braus [2].

Nehmen wir nun an, dass der Herzog nach einiger Zeit die Geduld verlor und das versprochene Gold einforderte. Bragadino zeigte ihm daraufhin einen schweren, goldfarbenen glänzenden Klumpen und behauptete, es handle sich um das begehrte Edelmetall. Der Herzog, skeptisch geworden, beauftragte einen Gelehrten, dies zu überprüfen. Dieser erhitzte daraufhin die Substanz in einem Schmelztiegel. Ein zweiter Tiegel mit einer Probe echten Goldes wurde zugleich auf eben diesem Feuer

erwärmt. Man stellte fest, dass das „künstliche Gold“ bereits über einem kleinen Feuer dahin schmolz, während das echte erst bei einer starken Glut flüssig wurde (genauer: bei 1.064 Grad Celsius, wie wir heute wissen).

Der Gelehrte stellte fest: „Die Schmelztemperatur der Substanz ist kleiner als die Schmelztemperatur von Gold“. Er folgerte daraus: „Es ist kein Gold“. Und ergänzte: „Es könnte sich um Blei handeln, dem durch Beimischen weiterer Substanzen ein goldfarbenes Aussehen verliehen wurde“.¹

1.3 Definition des Testbegriffs

Das Beispiel ermöglicht eine allgemeine Definition des Begriffs Test:

Durch einen Test wird festgestellt, ob die Eigenschaften einer Sache von den Erwartungen abweichen.

Diese Definition beinhaltet keine absolute Bewertung der zu prüfenden Sache im Sinne von „gut“ oder „schlecht“, obwohl dies häufig mit dem Begriff *Test* assoziiert wird. Vielmehr fordert die Definition lediglich einen Vergleich zwischen Ist- und Sollverhalten der zu prüfenden Sache. Demzufolge lautet das Ergebnis eines Tests stets „bestanden“ oder „nicht bestanden“.

Das Nicht-Bestehen eines Tests deutet demzufolge nicht zwangsläufig darauf hin, dass die Sache „schlecht“ ist, sondern kann andere Ursachen haben:

- Die Erwartungen an die Sache wurden zu hoch angesetzt
- Die Test-Rahmenbedingungen waren ungünstig gewählt.
- Die Test-Ausrüstung ist fehlerhaft

1.4 Wozu dienen Tests ?

Es gibt zwei grundlegende Sichtweisen, was Tests leisten können und welchem Zweck sie dienen.

Sichtweise a):

Tests dienen dem Nachweis, dass die Eigenschaften eines Produkts unter allen denkbaren Betriebsbedingungen mit den spezifizierten Eigenschaften übereinstimmen.

Im Bereich der Softwaretechnik gilt es als gesichert, dass ein solcher *Nachweis der vollständigen Korrektheit* der Implementierung eines gegebenen Algorithmus aus verschiedenen Gründen nicht möglich ist, z.B. [3].

In der Realität ist dies durch die folgenden Punkte begründet:

- Die Zeit, die zum Testen zur Verfügung steht, ist endlich (bzw. immer zu kurz)
- Es existiert niemals eine Spezifikation, die den Prüfling vollständig beschreibt

¹ Das Ende der Geschichte ist wahr, aber unschön: 1591 wurde Bragadino in München enthauptet

Diese Aussage gilt auch für elektronische Steuergeräte, da diese stets Software enthalten und deren Verhalten in hohem Maße durch Software bestimmt wird. Dieser Ansatz ist also im Bereich der Kfz-Elektronik nicht sinnvoll.

Sichtweise b):

Tests dienen dem Auffinden möglichst vieler Fehler in einem Testobjekt.

Dabei gilt:

Ein Fehler ist jede Abweichung der tatsächlichen Ausprägung eines Qualitätsmerkmals von der vorgesehenen Soll-Ausprägung [4].

Mit anderen Worten: Da das „Null-Fehler-Ziel“ für komplexe Systeme stets unerreichbar ist, sollten Tests dazu eingesetzt werden, um die Anzahl der Fehler zu minimieren. Diese pragmatische Zielsetzung ist für die Kfz-Elektronik geeignet.

2 Grundlagen und Definitionen

2.1 Testphasen und Testaktivitäten

Anhand von Bild 2 werden die drei wesentlichen Phasen und die jeweils dazu gehörigen Aktivitäten eines Testprozesses für technische Systeme erläutert.

Testvorbereitung („Pre-Processing“):

Während dieser Phase, die sich in mehrere Arbeitsschritte unterteilen lässt, werden alle für die Testdurchführung notwendigen Informationen ermittelt und die erforderlichen Gerätschaften bereitgestellt. Grundlage der Testvorbereitung ist stets ein Lastenheft (= Spezifikation), in dem die gewünschten Eigenschaften des zu testenden Produkts mehr oder weniger exakt festgelegt sind. Hierbei ist es zunächst unerheblich, ob das Lastenheft schriftlich, d.h. in formaler oder natürlicher Sprache, formuliert ist oder nur mündlich innerhalb des Entwicklungsteams kommuniziert wird. Folgende Schritte werden durchlaufen:

- Festlegung der Testziele für den konkreten Prüfling (Was soll getestet werden? Nach welcher Art von Fehlern soll vorrangig gesucht werden?)
- Festlegung des Testverfahrens (z.B. Black-Box/White-Box, etc.). Dazu gehört auch die Beschaffung der Testausrüstung.
- Ableitung einer Testreferenz, welche für jeden Satz von Eingabewerten (Stimuli) die korrekte Ausgabe des Prüflings in Form von Soll- oder Grenzwerten erzeugt. Die Testreferenz wird auch als *Oracle* bezeichnet.
- Definition einzelner Testfälle, d.h. der detaillierten Test- Durchführungs- und Auswertungsvorschriften. Dabei ist jeder im Lastenheft enthaltenen Anforderung mindestens ein Testfall zuzuordnen.

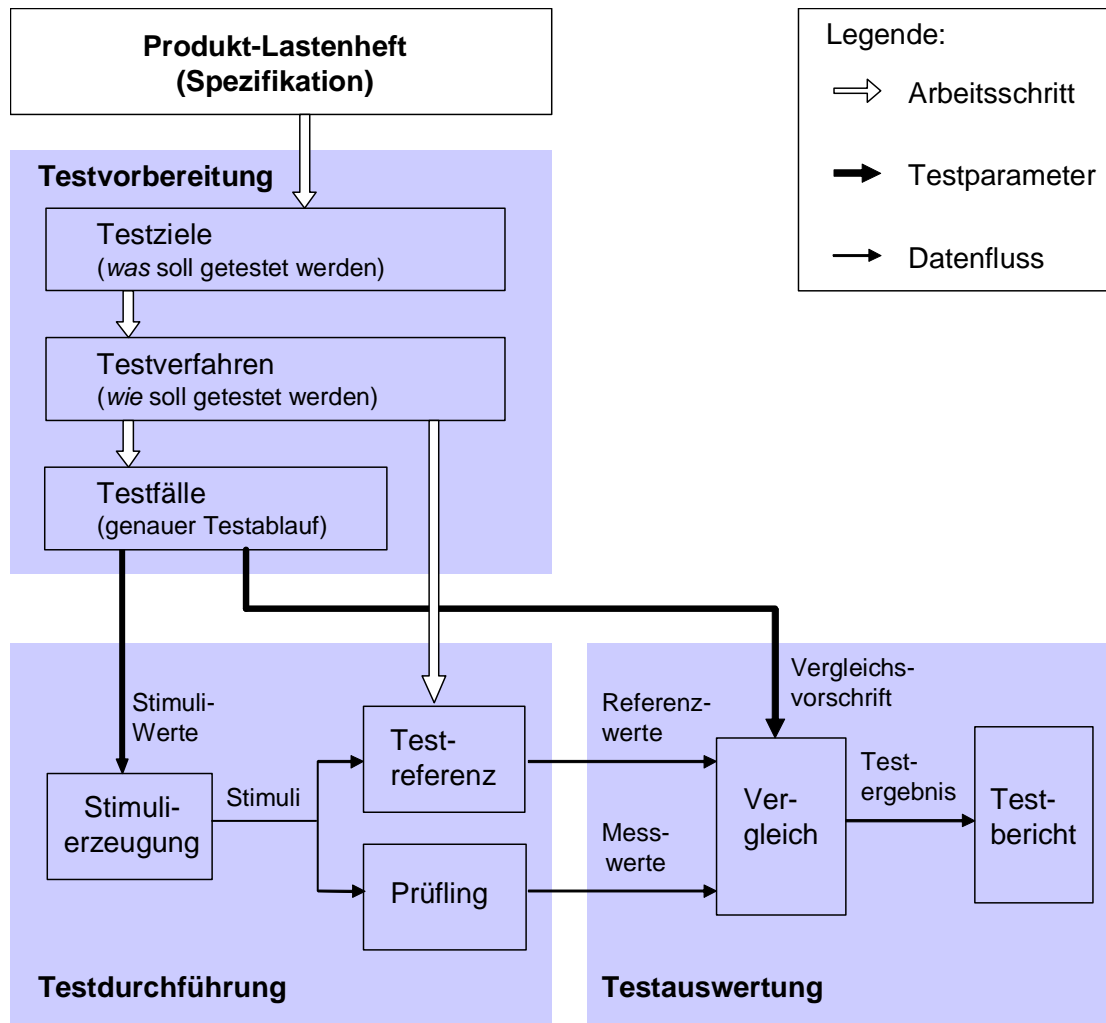


Bild 2: Testphasen und Testaktivitäten

Testdurchführung:

- Initialisierung des Prüflings: Sicherstellen, dass der Prüfling sich in einem bekannten Anfangszustand befindet. Dies ist eine wesentliche Voraussetzung für die Gültigkeit des Tests.
- Stimulation des Prüflings, d.h. Beaufschlagen der Eingänge mit physikalischen oder logischen Signalen.
- Durchführung von Messungen oder Beobachtungen am Prüfling.

Testauswertung:

- Vergleich der Ausgabewerte des Prüflings (Messwerte) mit den Ausgabewerten der Testreferenz (Referenzwerte) anhand einer Vergleichsvorschrift.
- Ableitung des Ergebnisses (Test bestanden oder nicht bestanden), sowie ergänzender Informationen zu Art und Umfang der Abweichung.
- Dokumentation dieser Ergebnisse (Eintragen in einen Testbericht).

2.2 Testebenen

Für einen effektiven Testprozess muss frühzeitig festgelegt werden, welche Arten von Tests zu welcher Zeit vorgenommen werden und wer dafür verantwortlich ist. Dabei gilt es einerseits, eine angemessene Testabdeckung zu erzielen, andererseits jedoch Mehrfach-Tests zu vermeiden. Diese Aufgabe ist alles andere als trivial, vor allem im Hinblick auf den hohen Vernetzungsgrad der Automobilelektronik und die große Zahl der Zulieferer, deren Produkte in einem modernen Kraftfahrzeug optimal zusammenwirken müssen.

In der Automobilindustrie hat sich das in Tabelle 1 dargestellte Vorgehensmodell durchgesetzt, welches in dieser oder ähnlicher Form von den OEMs und den großen Zulieferern eingesetzt und weiterentwickelt wird, wobei im Einzelfall weitere Ebenen eingeführt oder mehrere Ebenen zusammengefasst werden.

Tabelle 1: *Testebenen in der Kraftfahrzeugelektronik*

Testebene	Testobjekte	Testverfahren (Beispiele)	Zuständigkeit
Fahrzeugtest	Fahrzeug mit vollständiger Elektronik	Fahrversuch mit Datenlogging Subjektive Bewertung „EMV-Halle“	OEM, Zentralabteilung
Gesamtverbund-Test im Labor	Alle ECUs eines Fahrzeugs incl. Gateways	„CAN-Mobil“	OEM, Zentralabteilung
Teilverbundtest	Alle ECUs eines Bussegments, z.B. Antriebsstrang	HIL-Verbundtester „Brettaufbau“	OEM, Fachabteilung
Einzel-ECU-Test	ECU mit Software, evtl. mit Peripherie	Fail-Safe-Tester, HIL	OEM Zulieferer
Software-Integrationstest	Alle Softwareteile einer ECU	Code Coverage	Zulieferer
Software-Modultest	Einzelne Funktion oder Code-Modul	Code Review Metriken	Zulieferer

2.3 Testhierarchie

Für eine strukturierte Vorgehensweise empfiehlt sich eine hierarchische Strukturierung der Testabläufe. In Anlehnung an die in der Informationstechnik üblichen Begriffe wird für die Kraftfahrzeugelektronik folgende, mehrstufige Testhierarchie vorgeschlagen, die sich sowohl für den Test von Embedded Software als auch für den Steuergerätestest bewährt hat, siehe Bild 3.

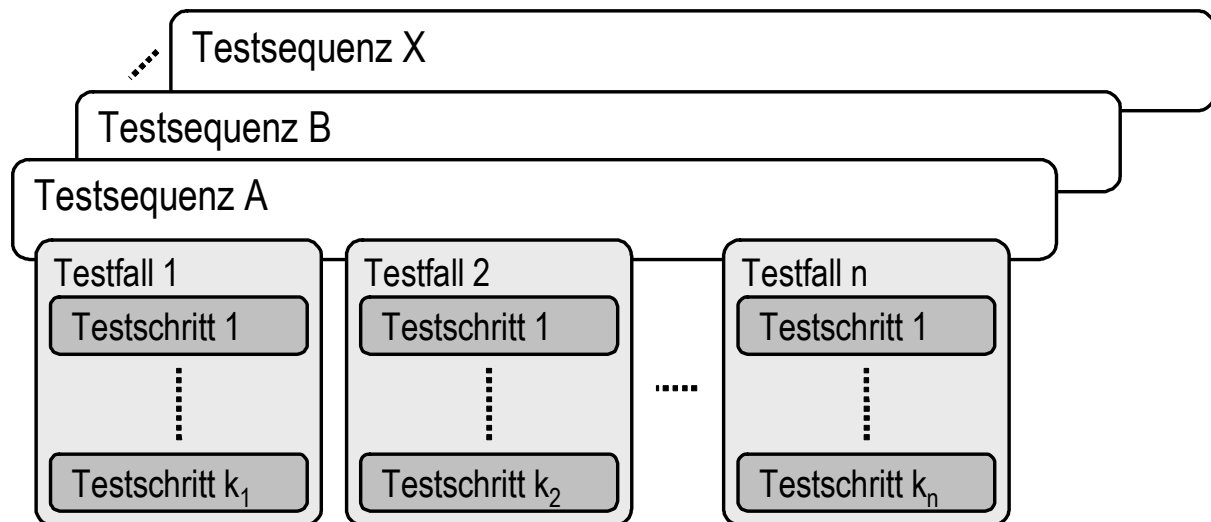


Bild 3: Hierarchischer Aufbau von Testabläufen

Im Mittelpunkt der Betrachtung steht der *Testfall*. Mit einem Testfall wird ein einzelner Pfad in der Software außerhalb oder innerhalb des Steuergerätes getestet, z.B. die Aktivierung einer einzelnen Funktion. Ein solcher Testfall besteht aus mehreren Operationen zur Erzeugung der Stimuli, Messung der Ausgangsgrößen sowie Auswertung und Dokumentation des Testergebnisses. Diese Operationen werden als *Testschritte* bezeichnet, die die „atomaren“ Bausteine eines Testablaufs bilden. Bei werkzeuggestützten Testsystemen steht eine Bibliothek solcher Testschritte zur Verfügung, z.B. in Form des Befehlssatzes einer Skriptsprache.

Der vollständige Test z.B. eines kompletten Software-Moduls oder einer ECU besteht aus einer Abfolge mehrerer, in der Regel zahlreicher, Testfälle. Diese werden zu einer *Testsequenz* zusammengefasst. Mehrere dieser Testsequenzen können wiederum zu einer Testsequenz höherer Ordnung zusammengefügt werden, z.B. für den Test mehrerer ECUs in einem Verbund.

Diese hierarchische Test-Strukturierung, die von verschiedenen marktgängigen Test-Werkzeugen unterstützt wird, sorgt für Übersichtlichkeit und korreliert mit den im vorherigen Abschnitt dargestellten Testebenen.

3 Klassifikation der Testverfahren

3.1 Qualitative und quantitative Tests, subjektive und objektive Bewertung

Ein gemeinsames Kennzeichen von Tests ist der Vergleich des Ist-Verhaltens eines realen Prüflings (auch: *Testobjekt*, *device under test*, *DUT*) mit einer Erwartung (Soll-Verhalten). Dieser Vergleich kann subjektiv erfolgen, d.h. durch Menschen anhand einer Bewertung von Sinneseindrücken, oder objektiv durch Erfassung von Messdaten und Vergleich mit Soll- und Grenzwerten.

Qualitative Tests sind für Kraftfahrzeug-Elektronik und Software von hoher Bedeutung, wenn sie eine Interaktion mit dem Benutzer gestatten. Dies gilt z.B. für Bedienelemente (Schalter, Taster, Drehknöpfe) im Kraftfahrzeug, sowie für Anzeigen und Menüführungen (Kombiinstrument, Navigationsgeräte, etc). Die Haptik und Akustik sowie das Design und die Ergonomie dieser Benutzerschnittstellen bestimmen in hohem Maße die subjektive Wahrnehmung von Fahrzeugeigenschaften wie „hochwertig“, „sportlich“, etc. Solche Bewertungen erfolgen prinzipbedingt stets durch menschliche Testpersonen, eine automatische Bewertung ist nicht möglich.

Dagegen ist eine subjektive Bewertung der „Innereien“ von Kfz-Steuergeräten, also der Elektronik und der Software, weder möglich noch sinnvoll, da diese Komponenten keinen ästhetischen Kriterien genügen müssen. Zudem sind die Ausgaben, also Daten oder elektrische Größen, einer direkten Sinneswahrnehmung nicht zugänglich.

Die folgenden Ausführungen beziehen sich deshalb ausschließlich auf quantitative Testverfahren für Elektronik und Software.

3.2 Statische und dynamische Testverfahren

3.2.1 Statische Verfahren

In der Softwaretechnik versteht man unter statischen Tests alle Verfahren, bei denen der Programmcode nicht auf einem Rechner ausgeführt wird. Statt dessen wird der Code auf bestimmte Merkmale untersucht. Verbreitete Verfahren sind z.B.:

- *Code Review (code inspection)*: Durchsicht von neuem Programmcode durch erfahrene Programmierer mit dem Ziel, fehlerhafte oder fehleranfällige Konstrukte zu finden, die Einhaltung von Programmierrichtlinien zu überwachen oder Software-Entwicklern individuelle Hinweise zur Optimierung ihres Programmierstils zu geben.
- *Software-Metriken*: Automatische Bestimmung von qualitätsrelevanten Kennzahlen von Programmcode. Z.B. bestimmt die Halstead-Metrik den zu erwartenden Test- und Wartungsaufwand eines Softwaremoduls aus der Anzahl der verwendeten Operatoren und Operanden [5].

3.2.2 Dynamische Verfahren

Bei dynamischen Verfahren wird der Programmcode entweder auf einem Emulator oder auf dem Zielrechner ausgeführt. Somit sind dynamische Verfahren grundsätzlich sowohl für Software als auch für komplette elektronische Steuergeräte und Steuergeräte-Verbünde geeignet. Da die dynamischen Verfahren von sehr hoher Bedeutung im Entwicklungsprozess sind, werden sie in weitere Klassen eingeteilt. Darauf wird in den folgenden Abschnitten eingegangen.

3.3 White-Box-, Black-Box- und Grey-Box- Testverfahren

Mit dieser Klassifikation der Testverfahren wird unterschieden, ob und in welchem Maße die interne Struktur des Prüflings bekannt ist, und auf welche Schnittstellen beim Testen zugegriffen wird.

3.3.1 White-Box-Test

White-Box-Tests (zutreffender auch *Clear-Box-Tests* genannt) erfordern die vollständige Kenntnis der internen Struktur des Prüflings. Für den Test von Software bedeutet dies, dass der Quellcode verfügbar sein muss. Eingesetzt werden White-Box-Tests vor allem für Software-Modultests. Die wichtigste Zielsetzung ist dabei die vollständige Abdeckung aller Pfade und Anweisungen (*code coverage*). Unterschiedliche Pfade ergeben sich durch Verzweigungen und Schleifen (z.B. if, while, ...) in Abhängigkeit von Variablen oder Konstanten. Jeder mögliche Pfad des Softwaremoduls und folglich auch jede einzelne Anweisung soll mindestens einmal durchlaufen werden. Dabei wird einerseits überprüft, ob die Ausgabegrößen der Software plausibel sind, andererseits sollen Implementierungs- und Laufzeitfehler wie Speichermangel, Variablen-Überläufe, fehlerhafte Typumwandlungen, nicht erreichbare oder nicht verlässbare Pfade etc. entdeckt werden.

White-Box-Tests sind in hohem Grade automatisierbar. Dies ist auch unabdingbar, da die Anzahl der möglichen Pfade mit jeder Verzweigung exponentiell ansteigt. Es existiert eine große Bandbreite von Testverfahren und Werkzeugen, die im Bereich der Informatik entwickelt wurden. Einzelheiten zu den einzelnen Methoden finden sich z.B. in [6].

Auf dem Gebiet der Kraftfahrzeugelektronik werden White-Box-Tests typischerweise beim Zulieferer durchgeführt, da der OEM in der Regel keinen Zugriff auf den Quellcode von Steuergeräte-Software hat (es sei denn, er entwickelt diesen selbst).

3.3.2 Black-Box-Test

Beim Black-Box-Verfahren werden die Tests ohne Zugriff auf interne Größen des Prüflings durchgeführt.

In der Kraftfahrzeugelektronik werden Black-Box-Tests auf allen Testebenen angewendet, d.h. für Software, für Einzel-Steuergeräte, mehrere Steuergeräte im Verbund bis zum vollständigen Fahrzeug.

In der Regel ist die interne Struktur des Prüflings (z.B. der elektrische Schaltplan einer ECU oder der Quellcode der eingebetteten Software) beim Black-Box-Test nicht oder nur in groben Zügen bekannt. Deshalb wird der Prüfling an seinen Eingangs- Schnittstellen mit einer Auswahl an Testsignal-Kombinationen (auch als *Testvektoren* bezeichnet) beaufschlagt. Anschließend wird festgestellt, ob die Ausgangssignale plausibel sind. Black-Box-Verfahren sind somit stets funktionsorientiert.

Da die Menge der möglichen Eingangssignalkombinationen bei umfangreichen Systemen sehr hoch ist, werden Black-Box-Tests typischerweise stichprobenartig durchgeführt.

Eine übliche, systematische Vorgehensweise beim Black-Box-Test ist die *Äquivalenzklassenbildung*. Hierbei werden alle Eingangssignale in Intervalle eingeteilt. Bewegt sich ein Eingangssignal innerhalb eines dieser Intervalle, so tritt keine Änderung an den Ausgängen des Prüflings auf. Die Anzahl der möglichen Eingangssignal-Kombinationen ergibt sich dadurch aus der Kombinatorik dieser Äquivalenzklassen. Eine Sonderform der Äquivalenzklassenbildung ist die *Grenzwertanalyse*. Die Intervallgrenzen liegen hierbei stets an den Grenzen zwischen gültigen und ungültigen Wertebereichen von Eingangssignalen. Dieses Verfahren ist von hoher praktischer Bedeutung, z.B. um das Verhalten eines Steuergeräts bei ungültigen Sensorsignalen zu überprüfen (*Fail-Safe-Tests*).

Eine in der Automobilelektronik bekannte und von verschiedenen Werkzeugen unterstützte Implementierung der Äquivalenzklassenbildung stellt die *Konfigurationsbaum-Methode* dar, die z.B. in [7] auf Embedded Software angewandt wird.

3.3.3 Grey-Box-Test

Der im vorherigen Abschnitt beschriebene Black-Box-Test stößt in der Praxis auf erhebliche Schwierigkeiten, weil die Spezifikation das *Ein-/Ausgangsverhalten* des Prüflings nur selten hinreichend exakt beschreibt. Die stets vorhandene Spezifikation der Art und Wertebereiche von Ein- und Ausgangssignalen ist nicht ausreichend, da das Gesamtverhalten nicht nur von den Eingangsgrößen, sondern auch von inneren *Zustandsgrößen* des Prüflings abhängt, deren Werte beim Black-Box-Test definitionsgemäß nicht bekannt sind.

Solche Zustandsgrößen treten immer dann auf, wenn das betrachtete System ein eigenes Zeitverhalten aufweist. Betrachtet man ein Steuergerät, so besitzt sowohl die Hardware als auch die Software Zustandsgrößen. Bei der Hardware sind diese z.B. Spannungen und Ströme in einem analogen Filter oder der Betriebsmodus eines CAN-Transceivers. Beispiele für Software-Zustandsgrößen ist der Wert eines Zählers oder der Integral-Anteil eines Reglers. Diese Größen beeinflussen den internen Programmablauf, also die Auswahl der bereits erwähnten Software-Pfade, und damit auch die Ausgangsgrößen.

Deshalb ist es von entscheidender Bedeutung, dass sichergestellt wird, in welchem Betriebsmodus sich der Prüfling gerade befindet. Dies kann z.B. dadurch erreicht

werden, indem der Prüfling zu Beginn eines Tests durch einen Reset in einen definierten Ausgangszustand versetzt wird. Bei einer ECU kann dies zum Beispiel durch Aus- und Einschalten der Stromversorgung („Klemme 30“) erreicht werden. Anschließend muss das Steuergerät durch genau definierte Werte-Abfolgen an den Eingängen in den Betriebsmodus versetzt werden, für den Tests durchgeführt werden sollen, was in der Realität sehr aufwendig und manchmal kaum möglich ist.

Abhilfe schafft das Grey-Box-Verfahren, das eine Mischform aus Black-Box- und White-Box-Test darstellt. Zusätzlich zu den Ein-/Ausgangsbeziehungen wird eine Auswahl von internen Zustandsgrößen beschrieben, auf die im Testablauf sowohl lesend als auch schreibend zugegriffen werden kann. Beim Steuergerätest gibt es hierbei unter Anderem folgende Möglichkeiten:

- Auslesen von Fehlercodes über die Diagnose-Schnittstelle,
- Zugriff auf ECU-interne Messwerte und Applikationsparameter über Diagnose- und Applikations-Schnittstellen,
- Explizite Bereitstellung ausgewählter interner Zustandsgrößen über eine eigene Schnittstelle für Entwicklungszwecke („Entwicklungs-CAN-Bus“),
- Explizite Aktivierung ausgewählter Software-Funktionen, z.B. so genannter Stellgliedtests über Diagnose-Services.

Mit Hilfe dieser Zusatzinformationen kann in vielen Fällen der augenblickliche Betriebszustand der ECU hinreichend genau ermittelt oder ein Modus explizit angefahren werden. Geeignete Diagnose- und Applikationsschnittstellen sowie Softwarewerkzeuge sind für diesen Zweck verfügbar. Das Grey-Box-Verfahren findet daher in der industriellen Praxis breite Anwendung für Steuergeräte-Tests.

3.4 Open-Loop- und Closed-Loop- Testverfahren

3.4.1 Closed-Loop-Verfahren

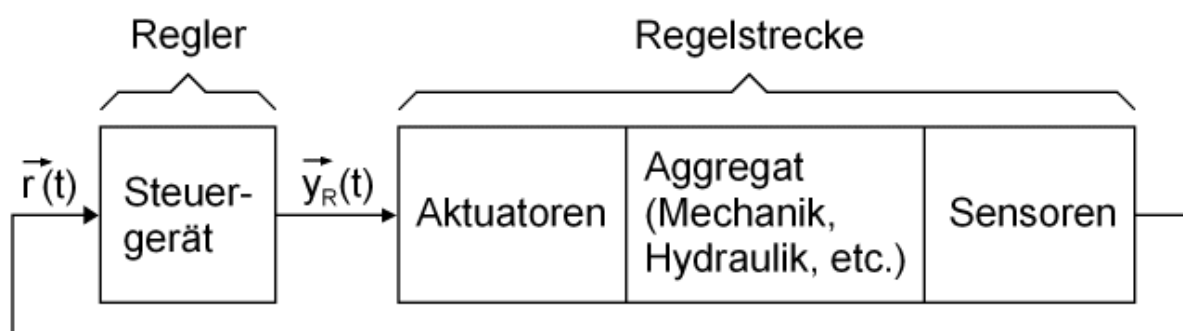


Bild 4: Regelkreis für ein elektronisch geregeltes Kfz-System

Im Laufe der vergangenen zehn Jahre hat sich eine Reihe von Testverfahren etabliert, bei denen der Prüfling in einer synthetischen Umgebung betrieben wird, die die Realität (z.B. ein Fahrzeug) durch eine Simulation möglichst exakt nachbildet.

Wenn sich hierbei ein geschlossener Regelkreis ergibt (Bild 4), so handelt es sich um ein „Closed-loop-“ bzw. „In-the-Loop“-Verfahren.

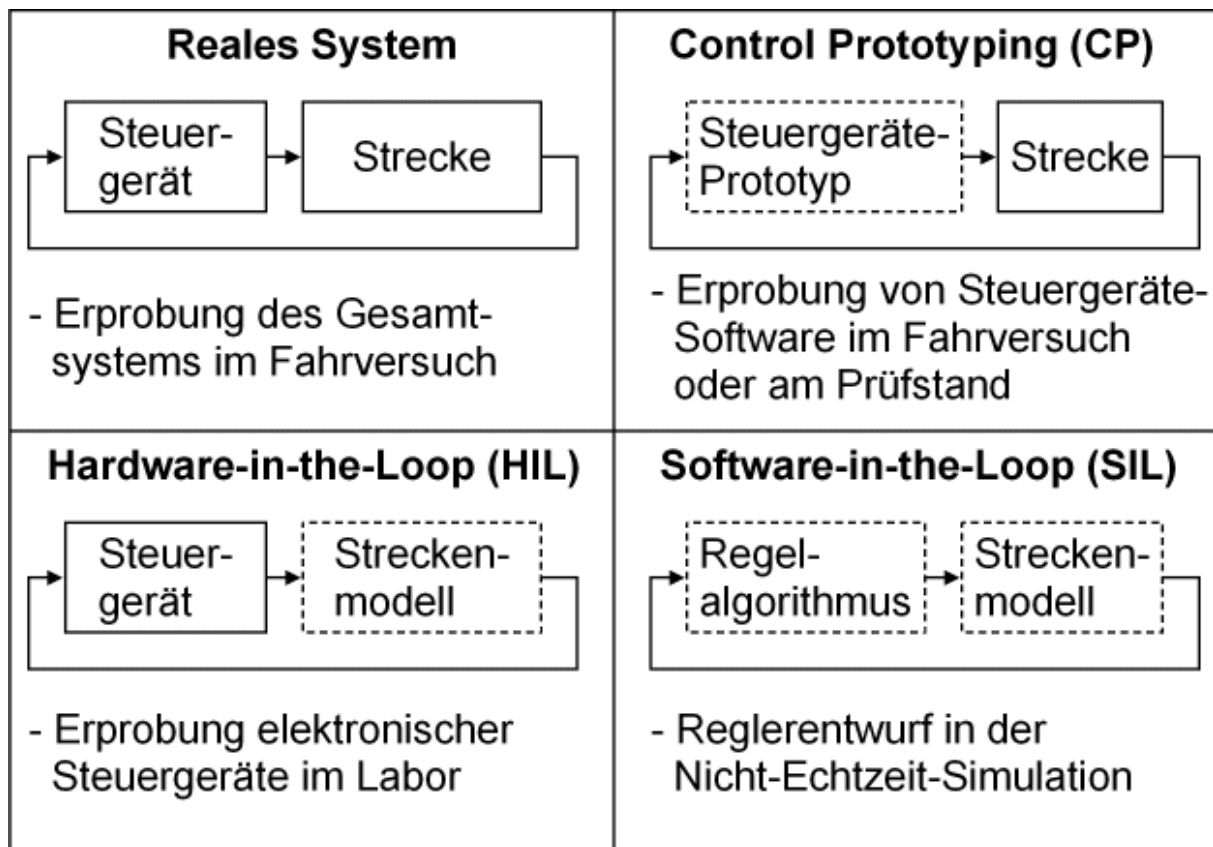


Bild 5: Einteilung der Testverfahren nach der Repräsentation von Steuergerät (Regler) und Regelstrecke.

In Bild 5 sind die verschiedenen Varianten und deren Anwendungen dargestellt. Die Verfahren unterscheiden sich dadurch, dass der Regler und die Regelstrecke jeweils real (im Bild durchgezogen dargestellt) oder simuliert (strichliert) sein können [8].

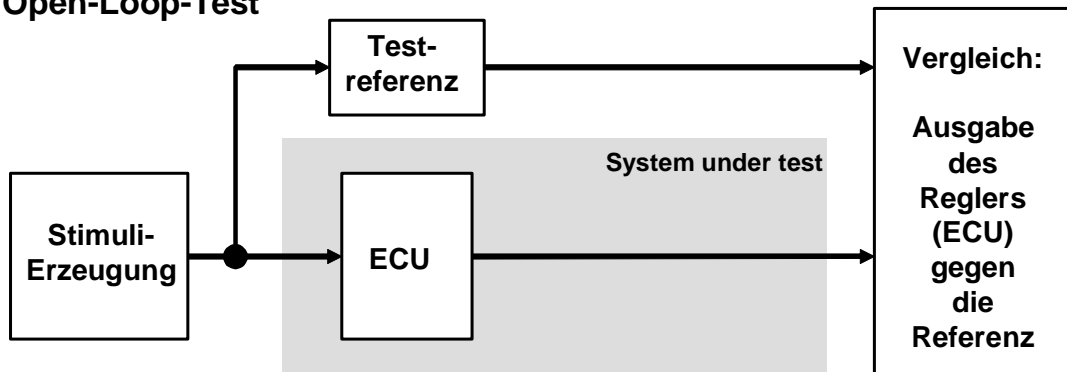
Daneben hat sich noch der Begriff *Model-in-the-Loop* etabliert. Hierbei wird der Regelalgorithmus in einem Modellierungswerkzeugs mit graphischem Editor (z.B. SIMULINK oder ASCET) abgebildet und im Regelkreis mit einem Streckenmodell betrieben. Es handelt sich dabei um eine Variante des Software-in-the-Loop-Verfahrens, da die erwähnten Softwarewerkzeuge streng genommen eine Softwareerstellung auf höherer Abstraktionsebene ermöglichen, wobei durch „Autocoding“ eine Generierung von Quellcode ermöglicht wird, meistens in der Programmiersprache C.

Eine detaillierte Untersuchung zum HIL-Verfahren findet sich z.B. in [9].

3.4.2 Abgrenzung zum Open-Loop-Test

Die im Folgenden beschriebene Unterscheidung der Testverfahren in Open-Loop- und Closed-Loop-Verfahren wird in der Fachliteratur und in der Test-Praxis häufig vernachlässigt. Ein Grund dafür ist, dass sich im Automobil-Umfeld der Begriff *Hardware-in-the-Loop* für eine Vielzahl von Testverfahren etabliert hat, bei denen elektronische Steuergeräte an einer Testmaschine im Labor geprüft werden. Da diese Unterscheidung jedoch nicht nur theoretischer Natur ist, soll darauf näher eingegangen werden.

Open-Loop-Test



Closed-Loop-Test

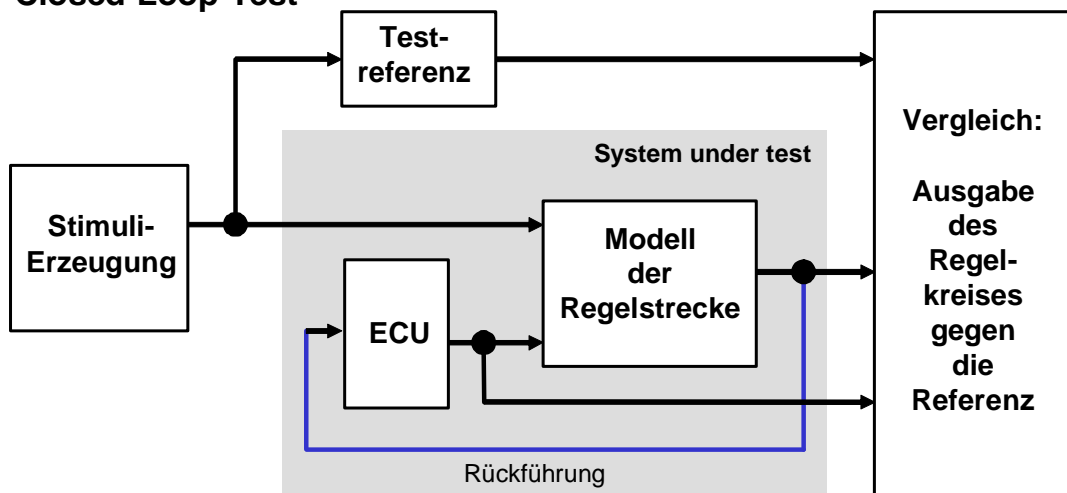


Bild 6: Blockschaubild für Open-Loop- und Closed-Loop-Tests

Die Begriffe *Open-Loop-Test* und *Closed-Loop-Test* werden in Analogie zur Regelungstechnik definiert. In Bild 6 sind Blockschaubilder für beide Verfahren jeweils anhand des Tests eines einzelnen Steuergeräts dargestellt. Die Überlegungen gelten sinngemäß auch für den Softwaretest.

Beim *Open-Loop-Verfahren* werden die Eingänge des Steuergeräts unmittelbar mit Testvektoren (Stimuli) beaufschlagt. Die Ausgangssignale werden erfasst und mit den von der Testreferenz erzeugten Sollwerten verglichen.

Durch Vergleich mit dem in Bild 2 dargestellten, allgemeinen Testmodell sieht man, dass das Steuergerät den Prüfling bildet.

Im Gegensatz dazu werden beim Closed-Loop-Verfahren die ECU-Eingangssignale nicht unmittelbar vom Stimuli-Generator des Testsystems erzeugt, sondern von einem Simulationsmodell der Regelstrecke. Diesem Modell werden sowohl die vom Testsystem generierten Stimuli als auch die Ausgangsgrößen der ECU zugeführt. Durch die Rückführung ergibt sich ein geschlossener Regelkreis. Das Testsystem erfasst sowohl die ECU-Ausgangssignale als auch die Modell-Ausgangsgrößen und vergleicht diese mit der Testreferenz.

Vergleicht man das Blockschaltbild wieder mit dem allgemeinen Test-Modell, so zeigt sich, dass bei dieser Testanordnung streng genommen der geschlossene Regelkreis aus ECU und Streckenmodell den Prüfling bildet. Dies bedeutet, dass die Aussagekraft des Testergebnisses nicht zuletzt von der Qualität des Simulationsmodells abhängt.

Die Entscheidung, ob ein Closed-Loop- oder Open-Loop- Testverfahren eingesetzt wird, hängt von den Eigenschaften der betrachteten ECU ab und sollte von Fall zu Fall sorgfältig abgewogen werden.

Open-Loop-Funktionstests sind vor allem für reaktive Systeme geeignet, also für ECUs, deren Verhalten sich im Wesentlichen durch Zustandsautomaten beschreiben lässt und die nicht Bestandteil von Regelkreisen mit kurzen Zeitkonstanten sind. Dies trifft auf viele Systeme im Bereich der Innenraum- und Karosserie-Elektronik zu. Hierbei lässt sich mit einfachen Mitteln eine relativ hohe Testabdeckung erzielen

Dagegen ist ein Closed-Loop-Test immer dann zu empfehlen, wenn schnelle Regelungen geprüft werden sollen, z.B. bei Motor- und Getriebesteuerungen sowie bei Schlupf- und Fahrdynamikregelungen (ABS, ESP). Aufgrund der erwähnten Abhängigkeit des Testergebnisses vom Streckenmodell sollte jedoch stets abgewogen werden, ob der Aufwand für ein Closed-loop-Testsystem mit dem notwendigen, hochwertigen Echtzeit-Simulationsmodell gerechtfertigt ist, oder ob alternativ ein frühzeitiger Test der Elektronik im Fahrversuch in Frage kommt.

4 Testautomatisierung

Die Begriffe *Testautomatisierung* und *automatische Testfallgenerierung* werden in Fachartikeln und Produktbeschreibungen häufig verwendet, ohne dass der Gegenstand und der Grad der Automatisierung exakt benannt wird. In diesem Abschnitt werden daher die Voraussetzungen, Möglichkeiten und Grenzen der Testautomatisierung beleuchtet.

Wenn man von *Testautomatisierung* spricht, so ist zunächst zu klären, in welchen *Testphasen* eine Automatisierung vorgenommen wird und welche Testaktivitäten hiervon erfasst werden.

4.1 Voraussetzungen für die Testautomatisierung

Für den gesamten Testprozess (wie auch für jede andere Automatisierungsaufgabe) gilt folgende einfache Grundregel:

Die automatische Durchführung einer Testaktivität setzt zwingend eine formale, maschinenlesbare Darstellung der Ergebnisse der vorausgehenden Testaktivität voraus.

Wendet man diese Regel beispielhaft auf die Aktivitäten *Testfallerzeugung* und *Testausführung* an, so ergeben sich folgende Konsequenzen:

- Aus einem informellen (in natürlicher Sprache verfassten) Lastenheft können Testfälle nicht automatisiert abgeleitet werden. Kurz: Automatische Testfallgenerierung erfordert formale Spezifikation.
- Die automatische Ausführung von Testabläufen auf einer Testmaschine erfordert eine formale Beschreibung der Testfälle, z.B. in Form einer Skriptsprache, die von der Maschine interpretiert werden kann.

Eine vollständige Automatisierung des Testprozesses würde demnach die Existenz einer formalen Spezifikation (Lastenheft) des Produkts voraussetzen, die die gesamte Funktionalität in Form einzelner Anforderung (Lasten) vollständig beschreibt. Dies muss in einer Art und Weise erfolgen, dass sich daraus sowohl die Testziele (also die Prioritäten der Fehlerfindung) als auch die Testfälle, die Testreferenz und die Vergleichsvorschriften automatisch mittels geeigneter Software ableiten lassen. Unter der Voraussetzung, dass eine geeignete Testmaschine zur Verfügung steht, wäre es somit möglich, den Prüfling ohne weitere menschliche Eingriffe zu testen.

4.2 Heutiger Stand der Testautomatisierung

Beim heutigen Entwicklungsprozess der Kraftfahrzeugelektronik ist man von dieser Idealvorstellung noch sehr weit entfernt.

Betrachtet man die Testaktivitäten aus Bild 2 im Hinblick auf den erreichten Formalisierungs- und Automatisierungsgrad, so zeigt sich folgender Zustand:

Testvorbereitung:

Eine Automatisierung in der Phase der Testvorbereitung ist aufwendig und bei frühen Aktivitäten nahezu unmöglich. Die Produkt-Spezifikation (Lastenheft) bildet das erste Glied in der Kette der Testaktivitäten. Das Lastenheft wird naturgemäß von Menschen verfasst, die bei der Formulierung des gewünschten Produktverhaltens eine Vielzahl von Zielgrößen berücksichtigen müssen. Neben den technischen Anforderungen, (d.h. der Funktionalität) gehen auch die Kundenakzeptanz, die Kosten oder gesetzliche Rahmenbedingungen in die Spezifikation ein.

Ähnliche Betrachtungen gelten für die Aktivitäten *Festlegung der Testziele* und *Auswahl des Testverfahrens*, die eine intelligente Priorisierung aufgrund von Erfahrungswissen der Testingenieure erfordern.

Das erstellte Lastenheft selbst kann entweder informell (in natürlicher Sprache) formuliert werden oder formal (mit einer exakt definierten Syntax und Semantik) so dass das Soll-Verhalten des Produkts unmissverständlich beschrieben wird.

Einige bekannte Vertreter solcher Beschreibungsformen im Bereich der Elektronik und Informationstechnik sind:

- UML, eine Notation für das Verhalten von Software und die Interaktion mit dem Benutzer und anderen Software-Modulen
- VHDL, eine Beschreibungssprache für programmierbare Logik-Bausteine
- Endliche Zustandsautomaten zur Beschreibung von Ereignis-getriggerten Abläufen
- Signalflussdiagramme zur Beschreibung von linearen oder nichtlinearen Signalverarbeitungsketten

Eine detaillierte Beschreibung dieser Notationen und die verfügbaren Software-Tools würde den Rahmen dieses Beitrags sprengen. Auch ist keine einzige dieser Beschreibungsformen geeignet, komplexe mechatronische Kfz-Systeme einschließlich Mechanik, Hydraulik, Elektronik und Software vollständig zu beschreiben. Hierfür müssen in der Regel mehrere Notationen kombiniert werden.

Zahlreiche Forschungsaktivitäten befassen sich mit der Frage, welche Notationen für mechatronische Kraftfahrzeug-Systeme geeignet sind, und auf welchem Weg sich daraus automatisch Testfälle und Testreferenzen ableiten lassen.

In der Praxis gestaltet sich die Einführung dieser Beschreibungsformen in den industriellen Entwicklungsprozess außerordentlich schwierig, da der Abstraktionsgrad im Vergleich zur natürlichen Sprache sehr hoch ist (Einstiegshürde). Auch ist deren Nutzen nicht unmittelbar evident, sondern zeigt sich häufig erst nach Monaten oder sogar Jahren.

Als Fazit kann festgestellt werden, dass praxistaugliche Verfahren zur automatischen Ableitung von Testfällen für Kfz-Elektronik heute nur in solchen Teilbereichen verfügbar sind, die eine Spezifikation mit einfachen Mitteln zulassen.

Stattdessen wird der überwiegende Teil der Testabläufe und Testreferenzen heute von Menschen erstellt. Die Testfälle selbst werden jedoch häufig maschinenlesbar notiert, z.B. in der Skriptsprache *Python* oder mit Hilfe graphischer Test-Editoren.

Testdurchführung und Testauswertung:

In der Phase der Testdurchführung und -Auswertung wurde bereits ein hoher Automatisierungsgrad erreicht, da auf diesem Feld Erfahrungen, Rechnerwerkzeuge und Test-Hardware aus anderen Bereichen der industriellen Automatisierungs- und Messtechnik übernommen werden konnten.

Beispielhaft ist in Bild 7 das Blockschaltbild einer Open-Loop-Testanlage für ein elektronisches Steuergerät dargestellt [10].

Die Testfälle und Testsequenzen werden in diesem Fall manuell in einen Testeditor programmiert, jedoch in einer formalen Form abgespeichert.

Die Testmaschine besteht im Wesentlichen aus einem Stimuli-Generator zur Erzeugung von elektrischen Signalen an den Steuergeräte-Eingängen und einem Messmodul zur Erfassung der ECU- Ausgänge. Eine Fehlereinheit ermöglicht die Erzeugung von Kurzschlüssen, Unterbrechungen etc auf den ECU-Leitungen. Des Weiteren ist ein Kommunikationsmodul zur Erzeugung und Erfassung von CAN- und ggf. LIN-Daten sowie zur Abfrage interner ECU-Daten über die Diagnose-Schnittstelle vorhanden.

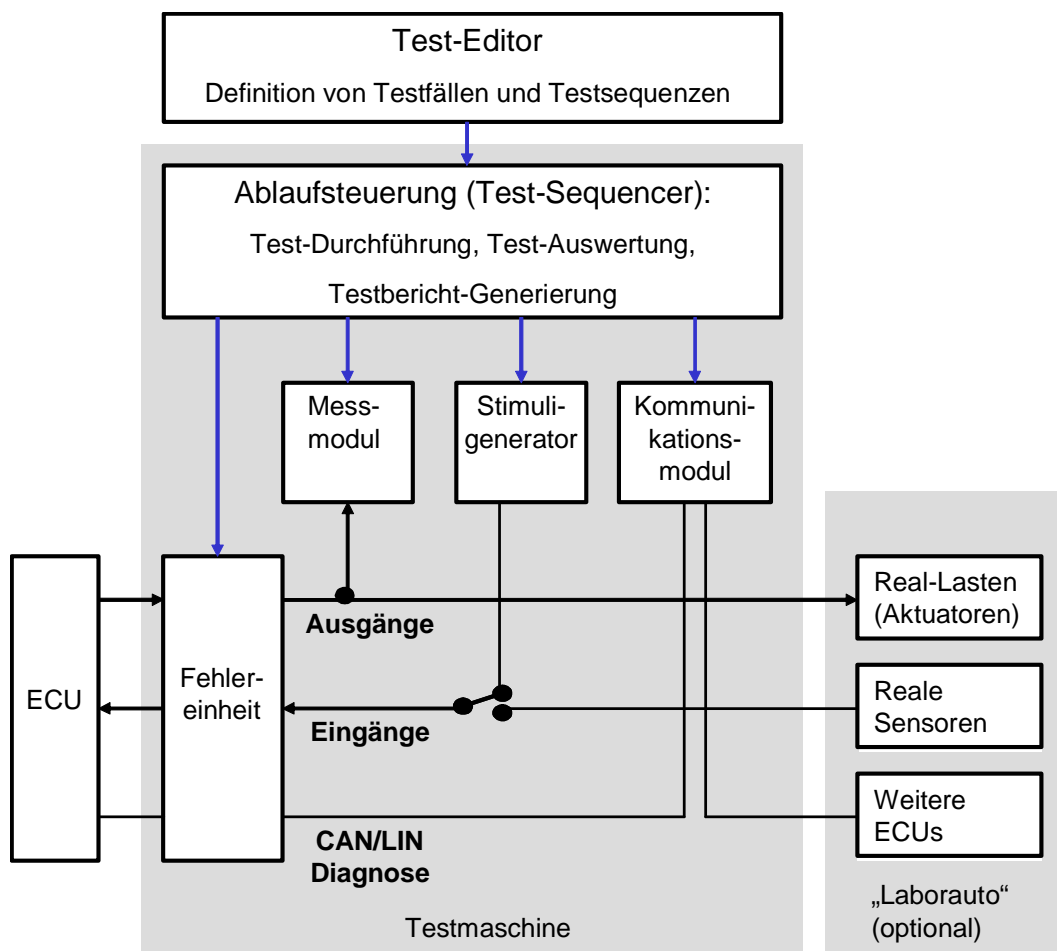


Bild 7: Automatische Testmaschine

Gesteuert wird die Anlage von einem Test-Sequencer, also einer Ablaufsteuerung, die auf einem Rechner ausgeführt wird. Der Sequencer liest die formal definierten Testsequenzen ein, führt die notwendige Stimulation der ECU durch, erfasst die ECU-Reaktionen, wertet diese aus und erzeugt automatisch einen Testbericht.

Bei der beschriebenen Anordnung wurden somit die Phasen „Testdurchführung“ und „Testauswertung“ vollständig automatisiert, während die Testvorbereitung manuell erfolgt.

5 Fazit

Aus der Informationstechnik ist bekannt, dass fast die Hälfte der Entwicklungskosten von Software auf Tests entfällt. Da die Embedded Software in der Kfz-Elektronik eine dominierende Rolle spielt, liegt der Testanteil bei den Entwicklungskosten für ECU's in einer ähnlichen Größenordnung. Das Testen von Kfz-Elektronik und -Software ist deshalb keine Nebenaktivität, sondern ein fester Bestandteil des Entwicklungsprozesses, der einen wesentlichen Beitrag zur Sicherung und Verbesserung der Qualität von mechatronischen Kfz-Systemen darstellt.

Um diese anspruchsvolle Aufgabe auch zukünftig bewältigen zu können, werden klar definierte Methoden, gut ausgebildete Ingenieure, leistungsfähige Software-Werkzeuge und standardisierte Test-Beschreibungsformate benötigt, die zwischen den auf dem Markt befindlichen Softwarewerkzeugen ausgetauscht werden können.

Auf diesen Gebieten liegen die Herausforderungen für die Zukunft.

Literatur

- [1] Harper, D.: *Online Etymology Dictionary*. <http://www.etymonline.com/>
- [2] Kreißl, F.; Krätz, O.: *Feuer und Flamme, Schall und Rauch*. Schauexperimente und Chemiehistorisches. Wiley-VCH Verlag, 1999.
- [3] Howden, W. E.: *Functional Program Testing & Analysis*. McGraw-Hill Verlag, 1987.
- [4] Liggesmeyer, P.: *Modultest und Modulverifikation: State of the art*. BI Wissenschaftsverlag, 1990.
- [5] Ebert, C.; Dumke, R.: *Software-Metriken in der Praxis*. Berlin: Springer-Verlag, 1996.
- [6] Dustin, E.; Rashka, J; Paul, J.: *Software automatisch testen*. Verfahren, Handhabung und Leistung. Berlin: Springer, 2000.
- [7] Conrad, M.: *Modell-basierter Test eingebetteter Software im Automobil*. Dissertation, TU Berlin, 2004.
- [8] Baumann, G.: *Werkzeuggestützte Echtzeit-Fahrsimulation mit Einbindung vernetzter Elektronik*. Dissertation, Universität Stuttgart, 2003.
- [9] Hartmann, N.: *Automation des Tests eingebetteter Systeme am Beispiel der Kraftfahrzeugelektronik*. Dissertation, Universität Karlsruhe, 2001.
- [10] Baumann, G.; Brost, M.; Reuss, H.-C.: *Durchgängige Open-Loop-Testverfahren für Kfz-Elektronik im Labor und Fahrversuch*. Vortrag, ASIM-Fachgruppe „Simulation technischer Systeme“ 2006.