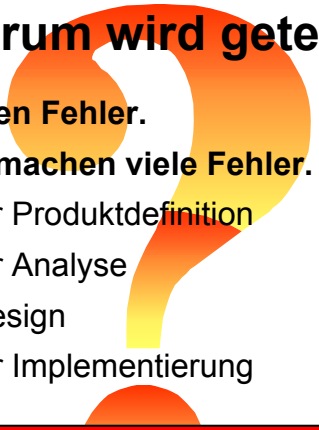
Systemtest

MICROCONSULT

Warum wird getestet?



Menschen machen Fehler.

Viele Menschen machen viele Fehler.

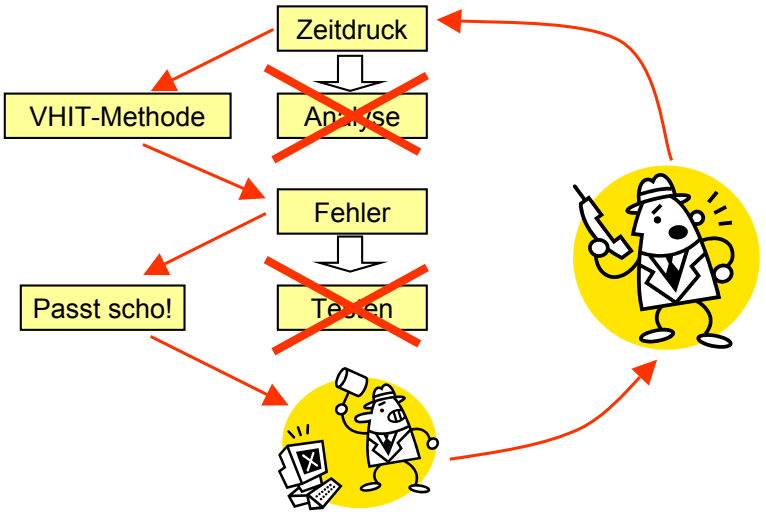
- Fehler in der Produktdefinition
- Fehler in der Analyse
- Fehler im Design
- Fehler in der Implementierung

Der Test hilft uns, die Fehler vor dem Kunden zu finden.

"Immer ist die Software Schuld, wenn wir unsere Termine nicht halten können!"

"Unser Systemtest wird immer durch blöde Fehler aufgehalten!"

"Wir hätten unser System gerne viel mehr getestet!"



VHIT: Vom Hirn ins Terminal

Systemtest

Der **Systemtest** ist der abschließende Test der Software in der realen Umgebung.

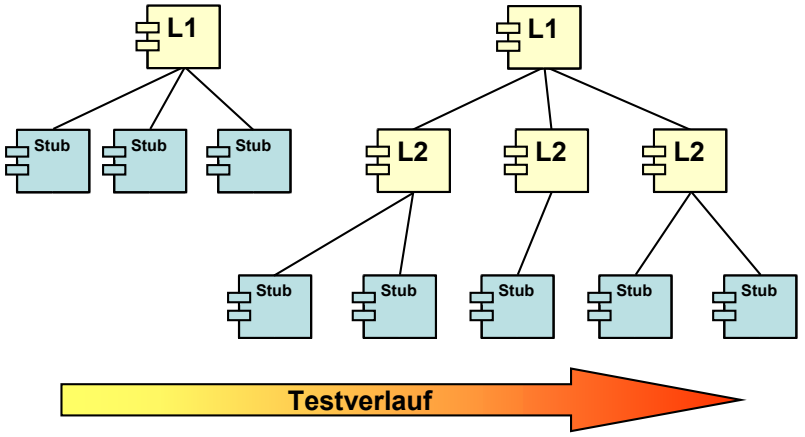
Er umfasst Systemsoftware, Hardware, Bedienungsumfeld → **System**.

Basis: Anforderungsspezifikation

Prüfung aller geforderten Qualitätsziele in ihrer jeweiligen Ausprägung → **Testziele.**

Der Systemtest ist der abschließende Test: Der Weg dahin

Top-Down Integration



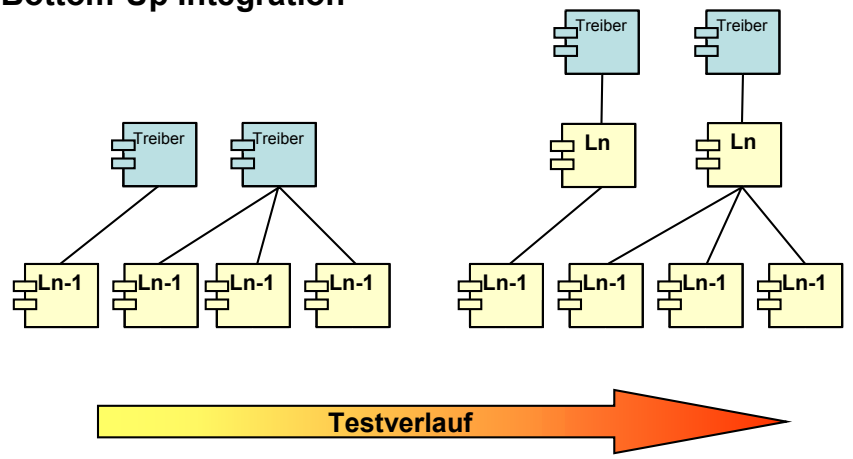
Top-Down Integration

+ Frühe Verfügbarkeit des Systems
+ Es sind keine Testtreiber nötig

- Hohe Anzahl an Stubs
- Später Test der Hardware-Anbindung

Ideal für einen frühen Systemtest

Bottom-Up Integration



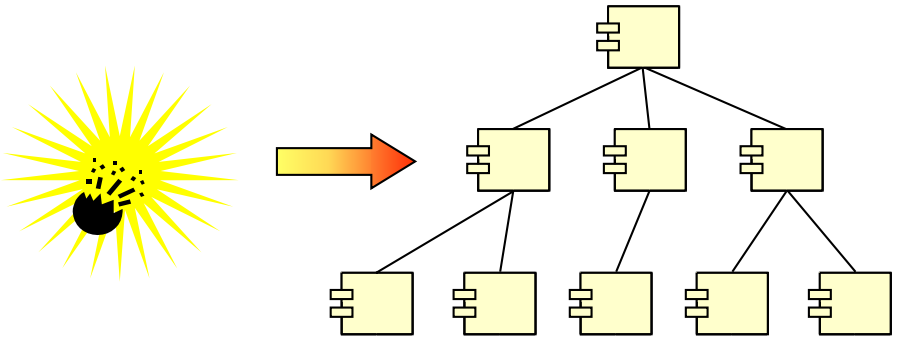
Bottom-Up Integration

+ Es sind keine Stubs nötig
+ Einfache Interpretation der Testergebnisse

- Späte Verfügbarkeit eines lauffähigen Gesamtsystems
- Ständige Änderung der Testtreiber und damit auch der Testfälle

Früher Systemtest ist sehr aufwendig

Big-Bang Integration



Big-Bang Integration

+ Es sind keine Testtreiber und Stubs nötig

- Schwierige Fehlersuche

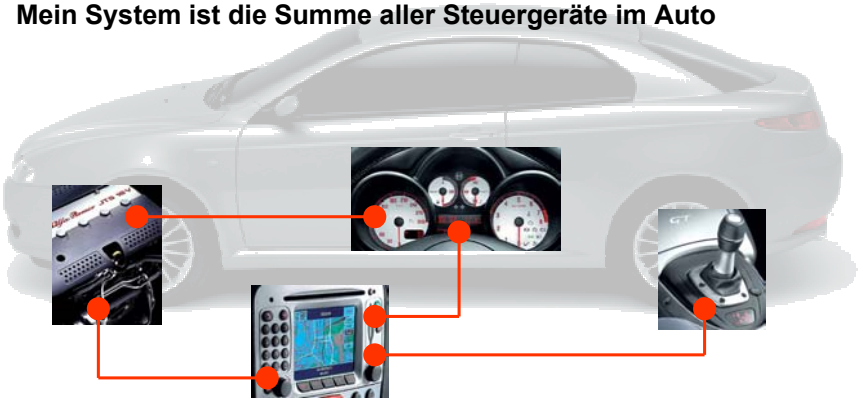
Früher Systemtest ist nicht möglich

Das System:
Systemsoftware, Hardware, Bedienungsumfeld

Was heißt System?

Automobilhersteller:

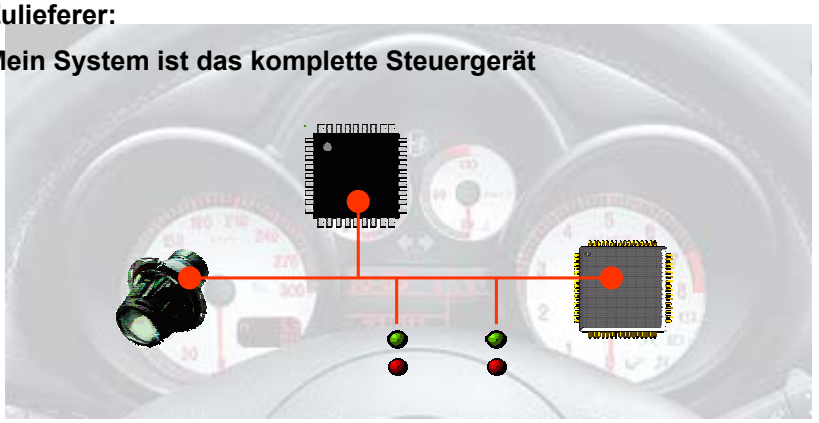
Mein System ist die Summe aller Steuergeräte im Auto



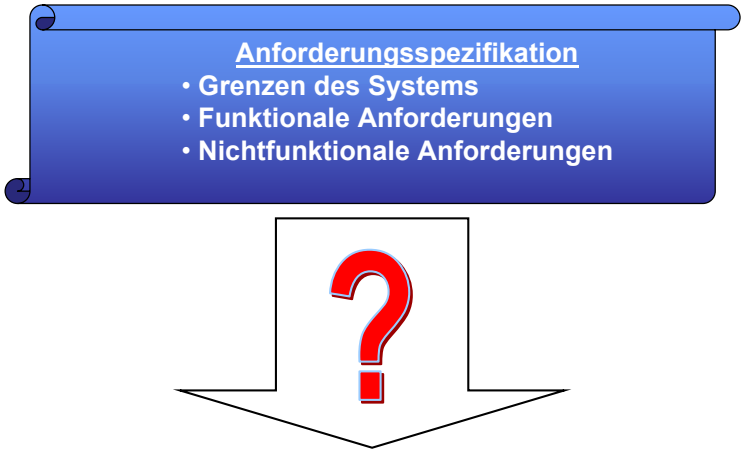
Was heißt System?

Zulieferer:

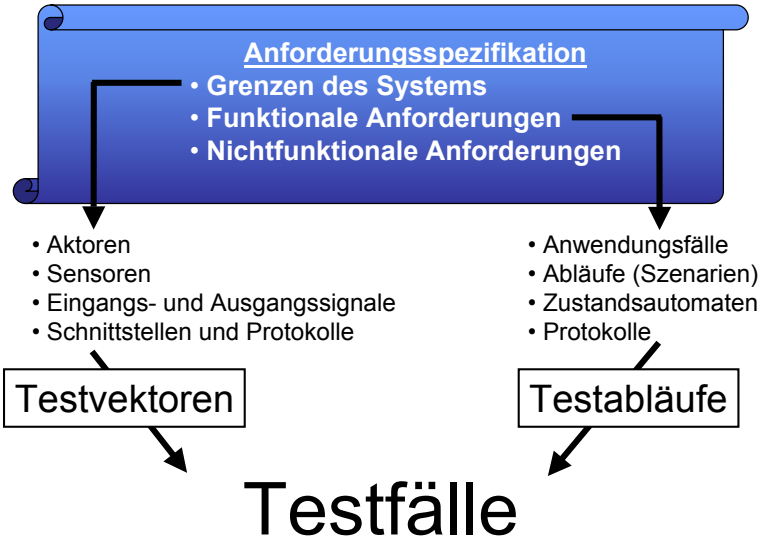
Mein System ist das komplette Steuergerät

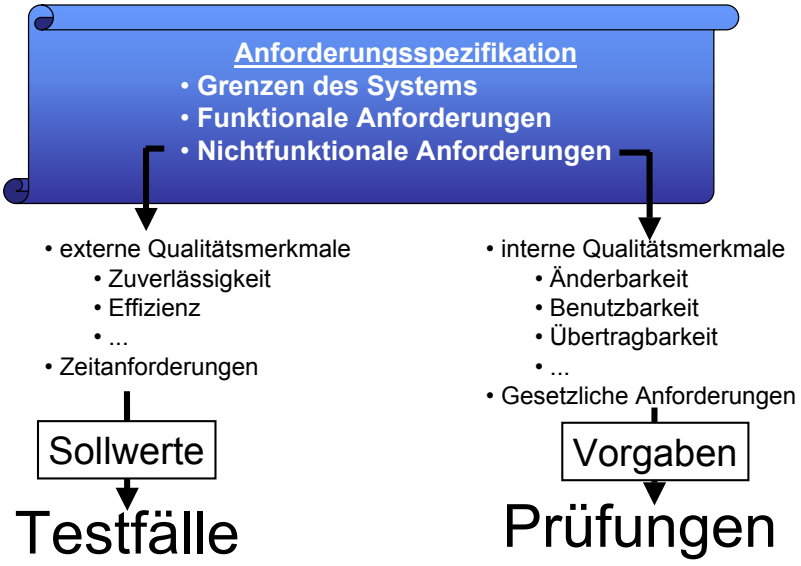


Die Basis: Anforderungsspezifikation



Testfälle, Prüfungen





Testziele

Die Haupttestziele bestehen in der Regel im Nachweis der aufgezählten Qualitätsmerkmale:

Qualitätsmerkmal	Qualitäts-Teilmerkmal
Funktionalität	Angemessenheit, Richtigkeit, Interoperabilität, Ordnungsmäßigkeit, *Sicherheit*
Zuverlässigkeit	Reife, Fehlertoleranz, Wiederherstellbarkeit
Benutzbarkeit	Verständlichkeit, Erlernbarkeit, Bedienbarkeit
Effizienz	Zeitverhalten, Verbrauchsverhalten
Änderbarkeit	Analysierbarkeit, Modifizierbarkeit, Stabilität, Prüfbarkeit
Übertragbarkeit	Anpassbarkeit, Installierbarkeit, Konformität, Austauschbarkeit

Nach DIN 66272, Oktober 1994 (ISO/IEC 9126 : 1991)

Mit freundlicher Genehmigung von CATS, Dr. Glöe

MICROCONSULT

Softwarequalität kann anhand verschiedener Merkmalen bewertet werden:

- **Funktionalität**
  ... Vorhandensein einer Menge von Funktionen, die festgelegte oder vorausgesetzte Erfordernisse erfüllen
- **Zuverlässigkeit**
  ... Fähigkeit der Software, ihr Leistungsniveau unter festgelegten Bedingungen über einen festgelegten Zeitraum zu bewahren
- **Benutzbarkeit**
  ... Aufwand, der zur Benutzung der Software erforderlich ist, sowie die individuelle Bewertung einer solchen Benutzung durch eine festgelegte oder vorausgesetzte Gruppe von Benutzern
- **Effizienz**
  ... Verhältnis zwischen dem Leistungsniveau der Software und dem Umfang der eingesetzten Betriebsmittel unter festgelegten Bedingungen
- **Änderbarkeit**
  ... Aufwand, der zur Durchführung vorgegebener Änderungen notwendig ist
- **Übertragbarkeit**
  ... Eignung der Software, von einer Umgebung in eine andere übertragen zu werden

Beispiel Kontaktanzeige:

Suche treuen Ehemann für gelegentliches Abenteuer,

keine finanziellen Interessen, Tel: 0190...

Die Haupttestziele bestehen in der Regel im Nachweis der a… …lten Qualitätsmerkmale:

Qualitätsmerkmal	Qua…
Funktionalität	Angemess… …erabilität, Ordnun… …eit
Zuverlässigkeit	R… …derherstellbarkeit
Benutzbarkeit	…rnbarkeit, Bedienbarkeit
Effizienz	…, Verbrauchsverhalten
Änderba…	…erbarkeit, Modifizierbarkeit, Stabilität, …arbarkeit
Übertragbar…	Anpassbarkeit, Installierbarkeit, Konformität, Austauschbarkeit

Maximal vier

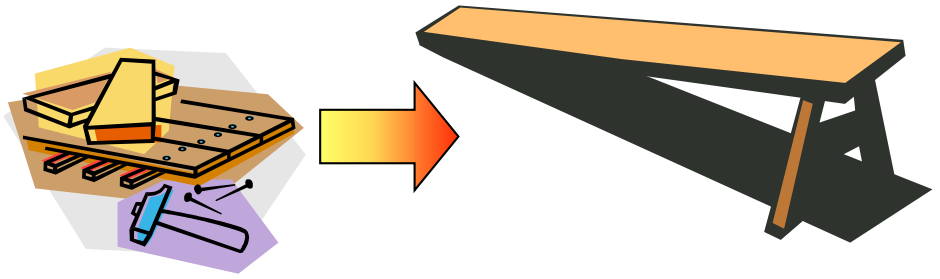
Nach DIN 66272, Oktober 1994 (ISO/IEC 9126 : 1991)

Mit freundlicher Genehmigung von CATS, Dr. Glöe

"Unser Systemtest wird immer
durch blöde Fehler aufgehalten"

MICRO CONSULT

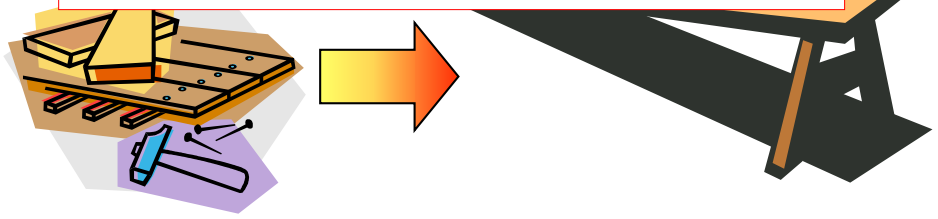
Reicht der Systemtest als einziger Test aus?



Der Systemtest baut auf den Daten aus der Analyse auf.

Reicht der Systemtest als einziger Test aus?

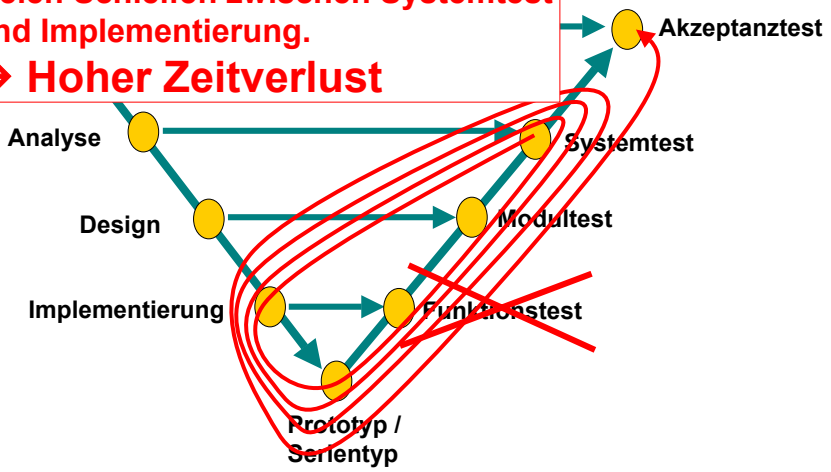
Untersuchungen zeigen:
80% der Projekte machen keine Funktionstests
und
wissen damit nicht, ob ihre
Funktionen robust sind!

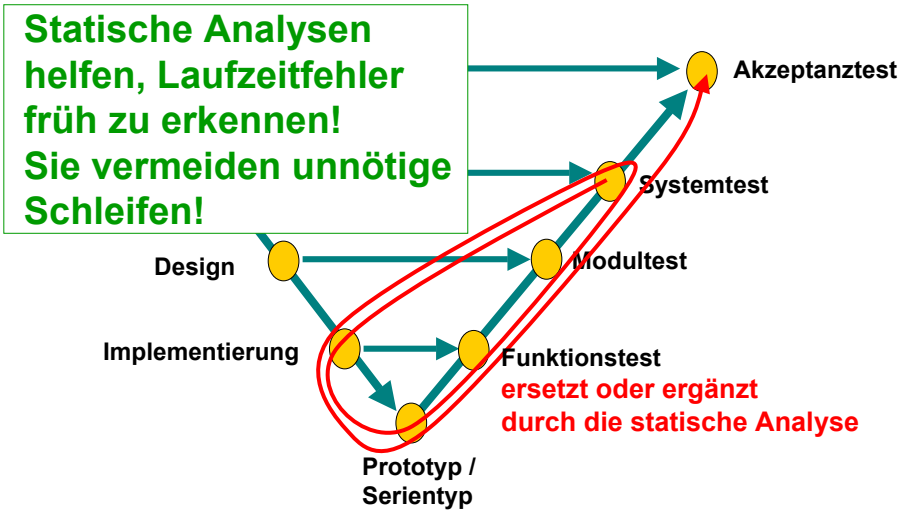


Der Systemtest baut auf den Daten aus der Analyse auf.

Mangelhafte Robustheit führt zu vielen Schleifen zwischen Systemtest und Implementierung.

→ Hoher Zeitverlust





Die statische Analyse ist ein
Prüfverfahren! (kein Test)

D.h. der Code wird zur Prüfung nicht ausgeführt.

Mit der statischen Analyse lassen sich Laufzeitfehler erkennen.

Die statische Analyse hilft einfache Fehler zu vermeiden.

Reviews

Automatische Analyse

- Compiler
- LINT oder Misra-C Checker
- PolySpace

Reviews

+ Hohe Fehlerfindungsrate (60%-90%)
+ Auch auf Dokumente anwendbar
+ Wissensverteilung im Team

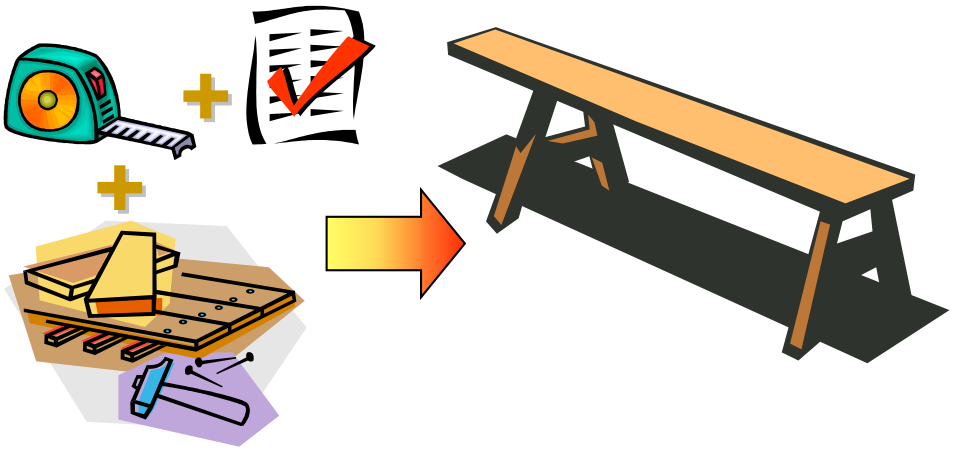
- Hoher Zeitaufwand beim Code-Review

Analysewerkzeuge

+ Der Code enthält keine Syntaxfehler (Compiler)
+ Überprüfung des Codes auf die Einhaltung von Standards und Programmierrichtlinien (Lint, MISRA-C Checker)
+ Abstrakte Interpretation des Source Codes (Polyspace)
+ Geringer Zeitaufwand

- Funktioniert nur für Code

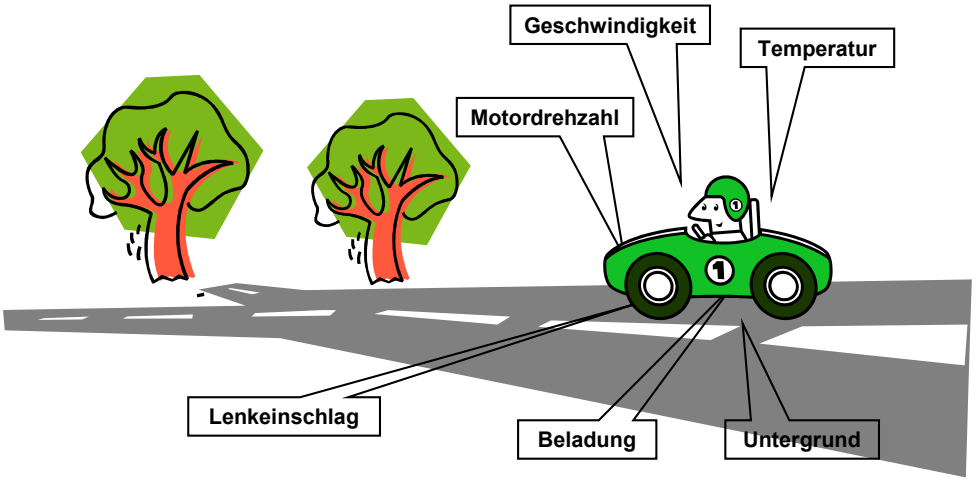
Kein Systemtest ohne statische Analyse!!!



Die Robustheit des Prüflings lässt sich durch statische Analyse stark verbessern.

"Wir hätten unser System gerne viel mehr getestet!"

MICROCONSULT



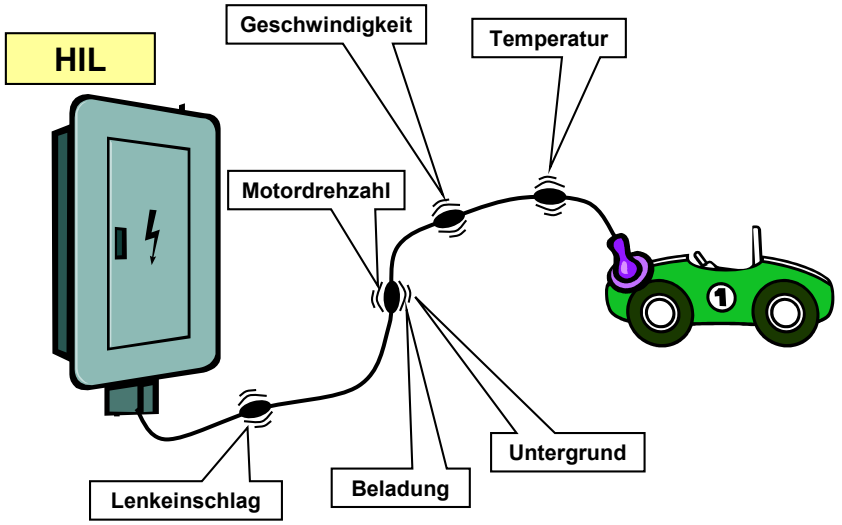
In der realen Umgebung wirken viele Faktoren auf ein System ein.

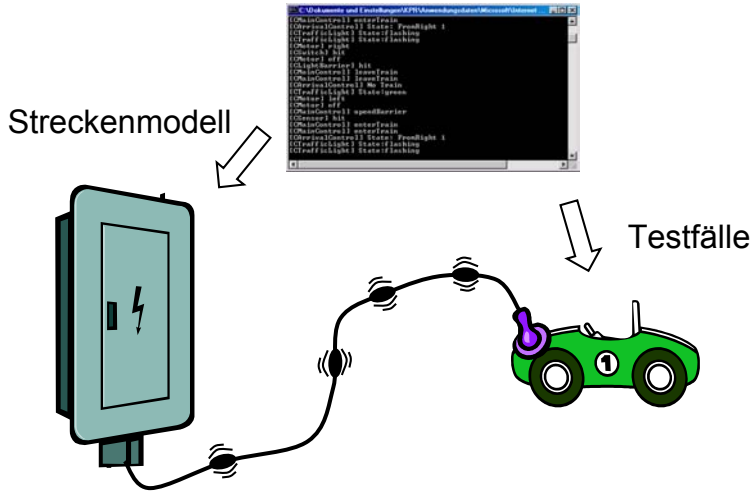
Alle Tests in dieser Umgebung durchzuführen ist wirtschaftlich nicht vertretbar.

Manche notwendige Tests sind auch nicht real ausführbar, sondern müssen simuliert werden (Auto: CAN-Nachlauf, AKW: Super-GAU)

Für den Systemtest einer Komponente ist es auch nicht sinnvoll, diese in ein Auto einzubauen und dann zu testen.

Hier muss ein Weg gefunden werden, um die reale Umgebung zu simulieren.





Vollständiges Testen ist unmöglich („Austesten“).

Testen ist eine kreative und anspruchsvolle Tätigkeit.

Tests müssen geplant sein.

Das Testziel muss definiert sein (Qualitätsmerkmale).

Testen ist destruktiv.

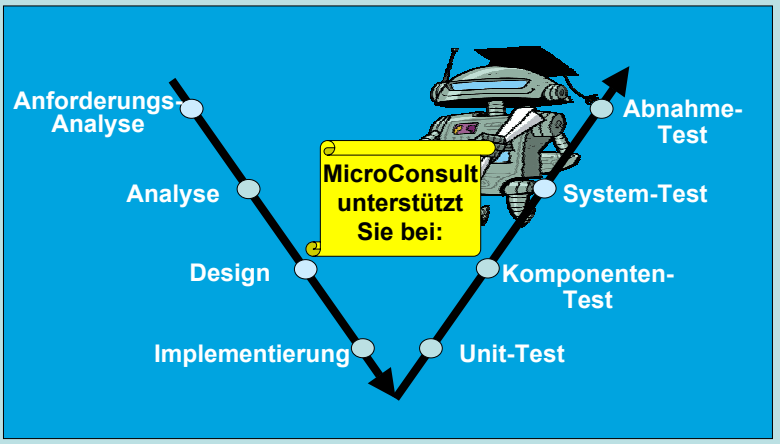
Testen erfordert Unabhängigkeit.

- Frage der Prioritäten: *Termin- oder Qualitätsziele?*

Zu jedem Testfall gehört ein Soll-Resultat.

- Nur ein *auffällig* falsches Resultat springt ins Auge!

Training, Coaching, Engineering



HW-/SW-Technologien, Tools, Methoden, Prozess, Team