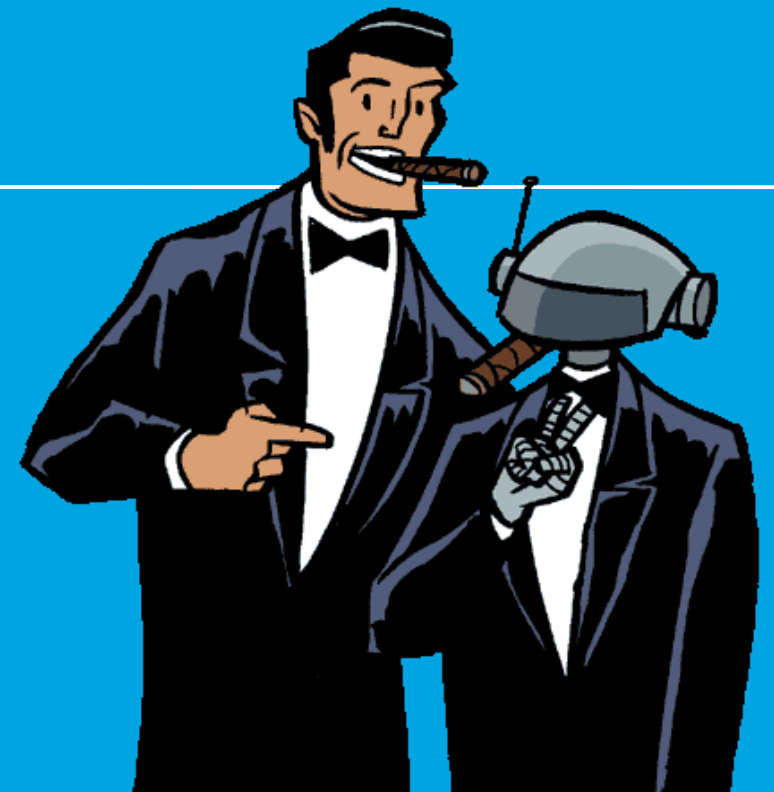


Begriffe zum Thema:

Test

Frank Listing
f.listing@microconsult.com

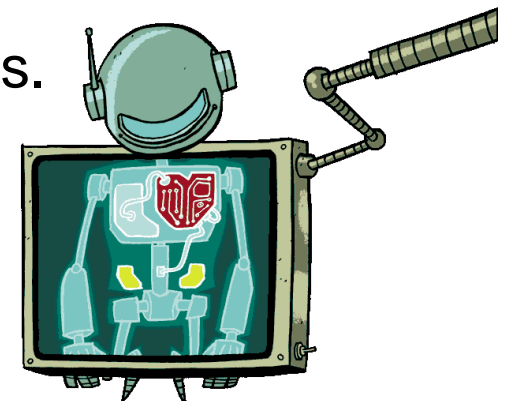


Testen:

- ist der Prozess, in welchem ein Programm ausgeführt wird, um die korrekte Umsetzung von Anforderungen nachzuweisen und noch vorhandene Fehler zu finden.

Tests:

- **sind geplante Vorgänge.**
- **sind wiederholbar** (Reproduzierbarkeit der Fehler).
- orientieren sich an klar **definierten Qualitätsmerkmalen.**
- setzen **klare Anforderungen** (Requirements) voraus.



Warum wird getestet?

- Menschen machen Fehler
- Viele Menschen machen viele Fehler



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

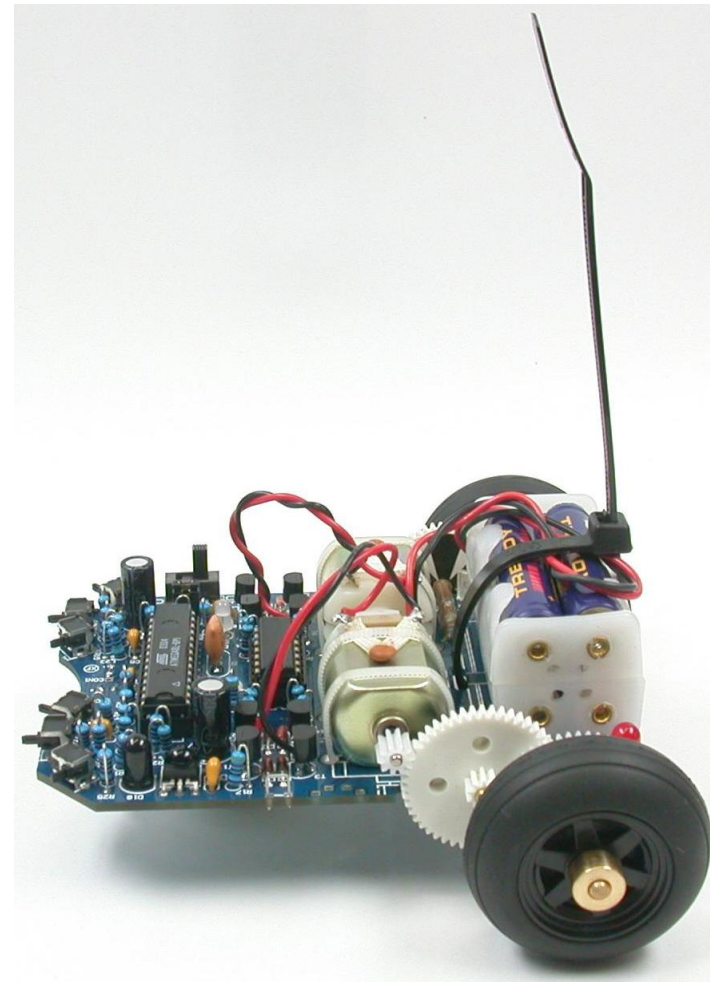
Validieren:

- ist das Prüfen auf die fachliche Korrektheit.
- Tut das System das Richtige?
- Ein Fachmann prüft die Spezifikation oder das gesamte System, ob es für den geplanten Einsatz zweckmäßig ist.



Verifizieren:

- ist das **Prüfen gegen die Spezifikation**.
- Ist das Produkt so, wie es in der Anforderungsspezifikation beschrieben wurde?



Die **FMEA** (Fehlermöglichkeits- und Einflussanalyse):

- ist eine analytische Methode der Zuverlässigkeitstechnik zum Finden von potentiellen Problemen.
- ist eine **Methode** für die **vorsorgende Fehlerverhütung**:
 - Eingrenzung und Strukturierung des betrachteten Systems
 - Analyse auf potentielle Fehlerursachen, Fehlerarten und Fehlerfolgen
 - Risikobeurteilung
 - Maßnahmen und Lösungsvorschläge
 - Restrisikobeurteilung bzw. -bewertung

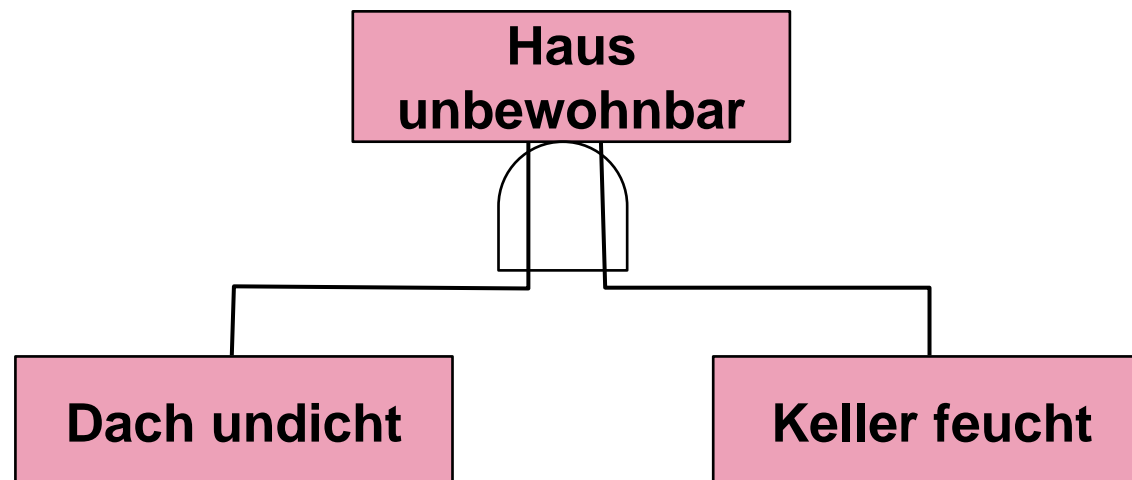
→ Es wird versucht herauszufinden, was alles schief gehen kann.



Die **FTA** (Fehlerbaumanalyse):

- ist ein Verfahren, mit der die **Wahrscheinlichkeit** eines **bestimmten Fehlers** bestimmt wird.
- Grundlage sind alle möglichen Ursachen eines Fehlers.
- Die Einzelfehler werden anhand der Booleschen Algebra logisch miteinander kombiniert.

→ Hier weiß man, was schief gehen könnte. Man versucht herauszufinden, wann es passiert und wie oft.



Der **Whitebox-Test**:

- basiert auf der **inneren Struktur** des **Testobjektes**.
- Testfälle werden aus der Struktur des Testobjektes heraus entwickelt und nicht nach funktionalen Anforderungen:
 - **Kontrollfluss-orientiert** (Programmablauf)
 - **Datenfluss-orientiert** (Initialisierung und Nutzung der Daten)

Ziel des **Whitebox-Test**:

- ist die **Abdeckung** der **Pfade** und **Bedingungen**
→ Nachweis der Robustheit des Testobjektes



Die **Codeüberdeckung** (Code Coverage):

- ist ein Maß, das angibt, wie viel der möglichen Programmpfade mit einem Test durchlaufen wurden.

ACHTUNG: 100% Codeüberdeckung sind nicht gleich 100% Codeüberdeckung.

Es gibt verschiedene **Maßzahlen** für die **Codeüberdeckung**, z.B.:

- **Anweisungsüberdeckungstest (C0)**
- **Zweigüberdeckungstest (C1)**
- **Modifizierte Verzweigungs- und Bedingungsüberdeckung (MCDC)**

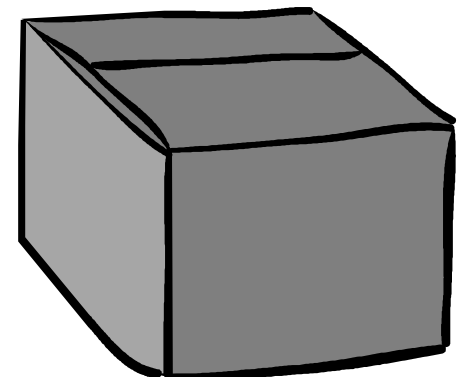
Hinweis: Vor dem Test muss genau definiert werden, welches **Überdeckungsmaß** erreicht werden soll.

Der **Blackbox-Test**:

- betrachtet das Testobjekt nur an seinen äußeren Schnittstellen.
- **Testfälle** für den Blackbox-Test werden anhand der Spezifikationen des Testobjektes entwickelt:
 - **Anforderungsspezifikation**
 - **Architekturspezifikation**
 - **Handbuch**

Ziel des **Blackbox-Tests**:

- ist eine möglichst umfassende und redundanzarme Prüfung der Funktionalität.



Äquivalenzklassen:

Zerlegung der Eingabeparameter und Wertebereiche der Ausgabeparameter in Äquivalenzklassen.

- **gültige Äquivalenzklassen** => Funktionstests
- **ungültige Äquivalenzklassen** => Stabilitätstests

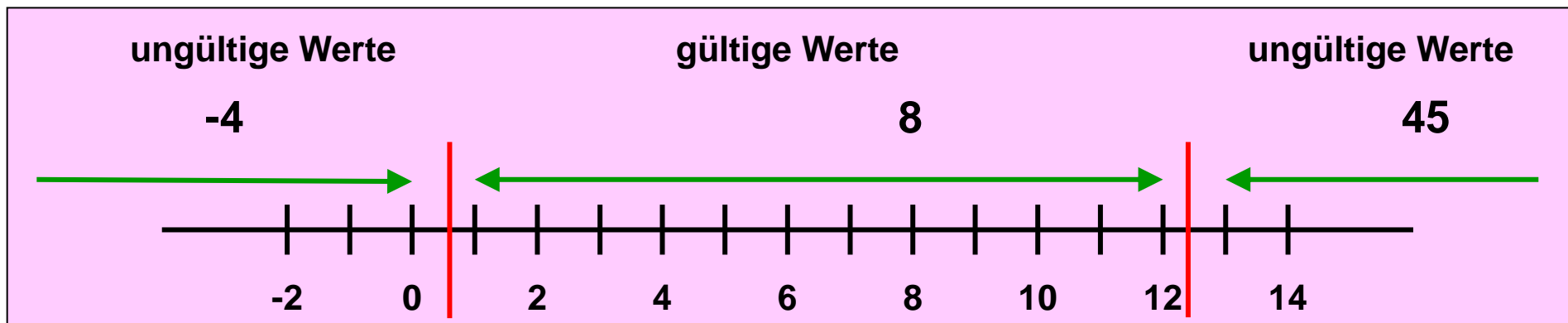
Annahme: Für jeden Repräsentanten einer Klasse reagiert das Programm gleich.

Testfall: Auswahl eines Repräsentanten aus jeder Äquivalenzklasse.

Ziel der Äquivalenzklassen:

- **Verringerung** der **Anzahl** der **Testfälle** durch geschickte **Äquivalenzklassenbildung** bei gleicher Testqualität.

Beispiel für Äquivalenzklassen:
Ein Parameter erwartet einen Monat als Zahl.



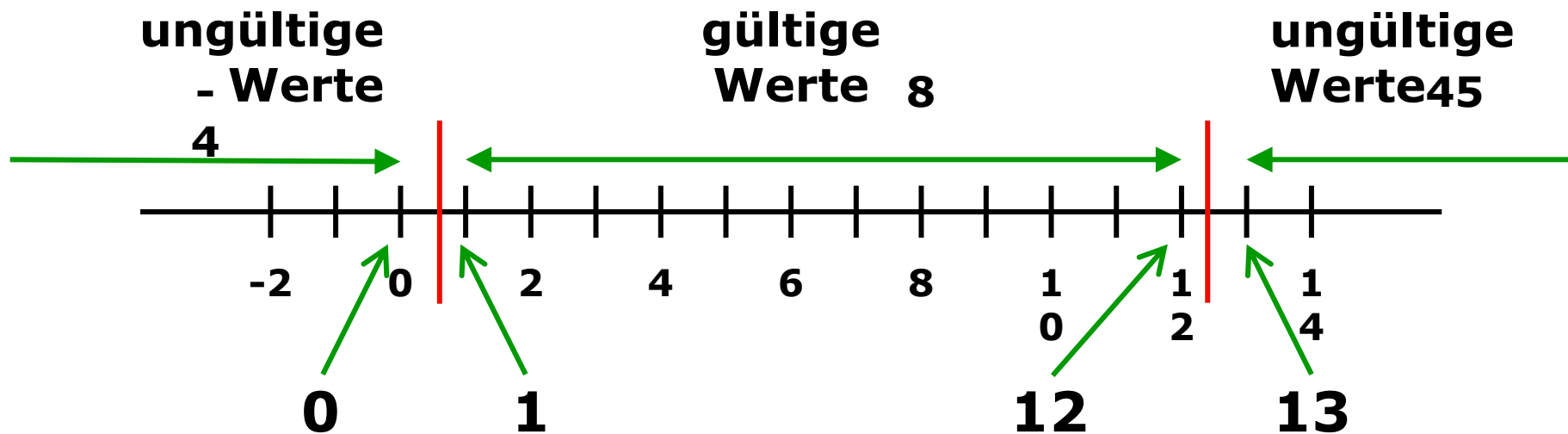
→ 3 Testfälle

Die **Grenzwertanalyse**:

- basiert auf den **Äquivalenzklassen**
- betrachtet die **Grenzen der Wertebereiche**
- dient der **Annäherung** vom **gültigen** als auch vom **ungültigen Bereich**
- ist nur möglich, wenn die Elemente der **Äquivalenzklassen** auf natürliche Weise geordnet werden können
- beachtet auch **Grenzen des Wertebereiches** der **Datentypen**

Äquivalenzklassen

Werte z.B. -4, 8, 45



Grenzwertanalyse

Werte: 0, 1, 12, 13

Der **Abnahmetest – Akzeptanztest**

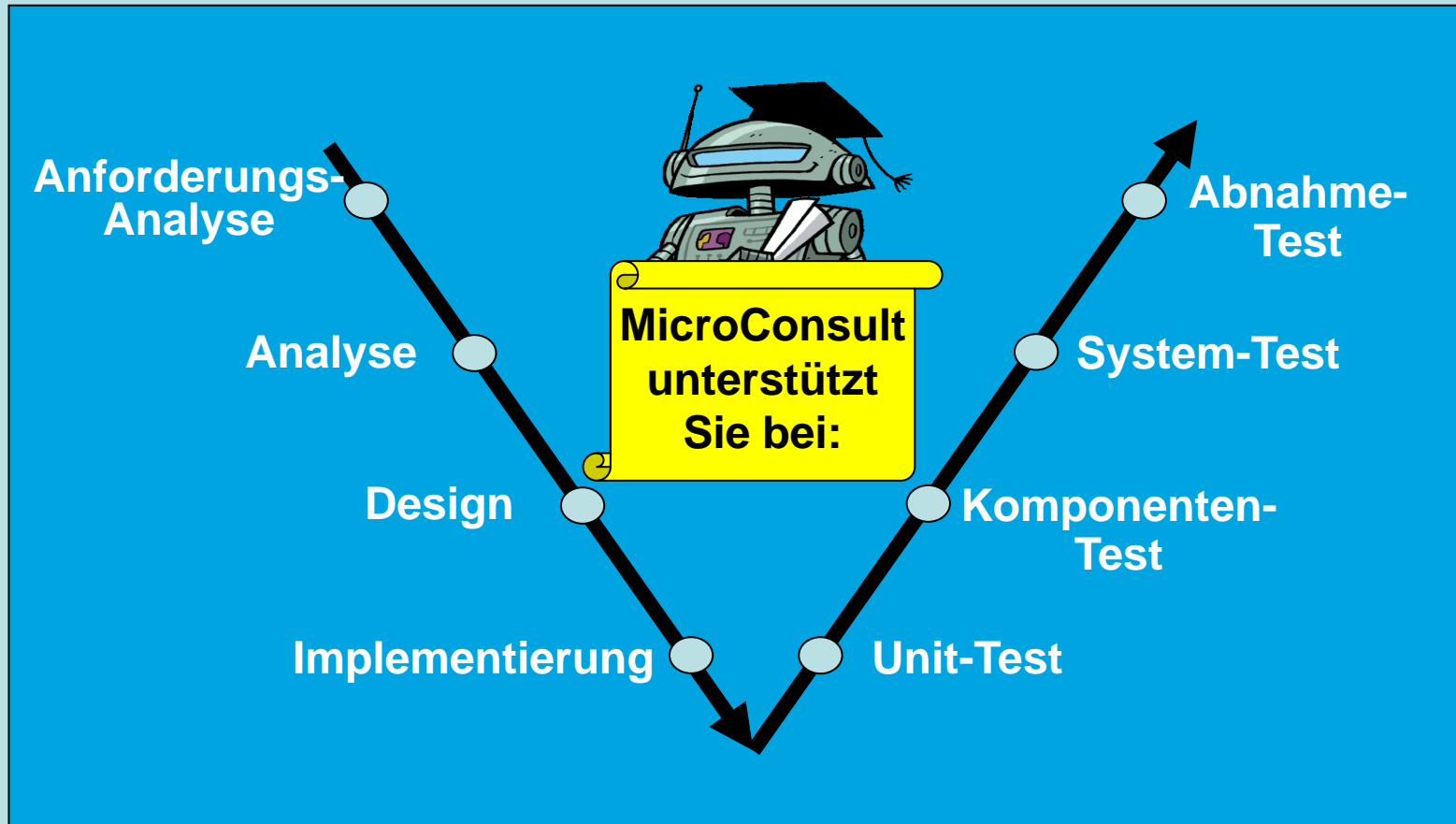
(UAT: User Acceptance Test):

- ist der **Test** auf **vertragliche Akzeptanz**.
- verwendet als **Basis** für die Testfälle das **Lastenheft** des Auftraggebers.
- soll sicherstellen, dass das Produkt alle **geforderten Anforderungen** wie Abnahmekriterien, Standards und Gesetze **erfüllt**.
- wird auf das **gesamte System** angewendet und **vom Kunden oder mit dem Kunden durchgeführt**.
- **endet** mit dem **formalen Abnahmeverfahren**.

Ziel des Abnahmetests:

- **Übereinstimmung** der **Leistung** mit der **Spezifikation**:
- Der Auftraggeber bestätigt, dass das Produkte bzw. die Leistung alle spezifizierten Anforderungen erfüllt.

Beratung, Training, Workshops, Coaching, Consulting



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

Vortrag per E-Mail anfordern: info@microconsult.de
 Betreff: „Vortrag embedded world 2012“