Entwicklungsbegleitender Test mechatronischer Systeme

Dr. Olaf Maibaum

Deutsches Zentrum
für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

Übersicht

- Test von Regelungssoftware
- Testansätze
  - MiL
  - SiL
  - PiL
  - HiL
- Vergleich der Testansätze
- Testautomatisierung
- Testfälle
- Simulation
- Rolle des System Engineering
- Zusammenfassung

Test von Regelungs-Software

- Regelungs-Software ist in ein technischen System eingebettet
- Funktionales Testen alleine ist nicht ausreichend
  - Einbeziehung von
    - Interaktion mit der physikalischen Umgebung
    - Timing der Regelungs-Software
- Regelungs-Software muss „in the Loop“ getestet werden.

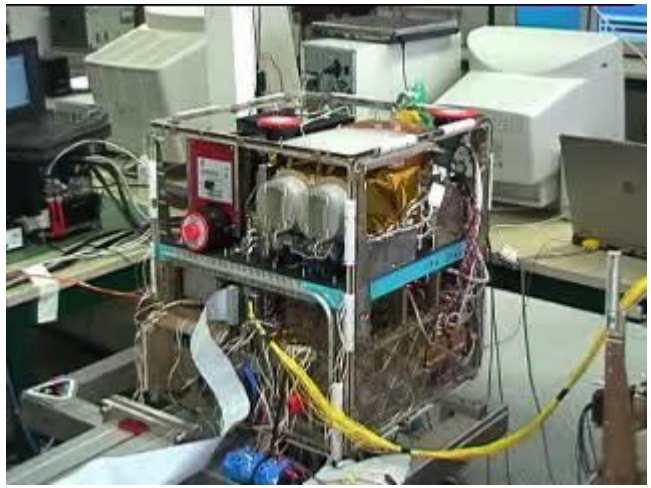


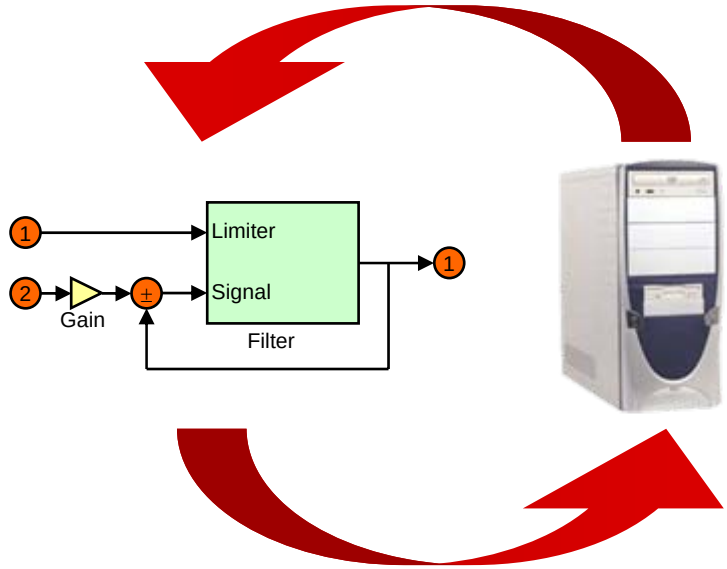
Bild: DLR

Herausforderungen für den Test

- Beschränktes bzw. kein Benutzerinterface
- Beschränkte System-Ressourcen
  - Speicher
  - Prozessorleistung
  - Debugging-Interfaces
- Echtzeitfähigkeit
  - Test-Instrumentierung ändert Zeitverhalten
  - System kann nicht angehalten werden
- Test von langen Laufzeiten
- Fehlermaskierung durch zeitliche Jitter
- Elektromagnetische Effekte im Test-Bed
- Test von gealterten Aktuatoren und Sensoren

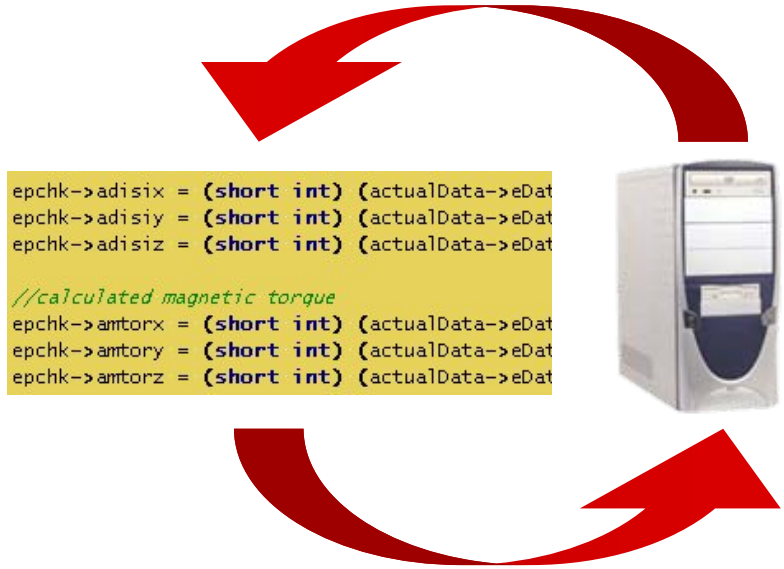
Model in the Loop (MiL) Test

- Test eines Modells der Steuerfunktionalitäten
- Das zu testende Modell ist eingebettet in eine Simulation von
  - Kommunikationsschnittstellen
  - Aktuatoren und Sensoren
  - physikalische Umweltmodell
- Einsatzzweck:
  - Test von Algorithmen



Software in the Loop (SiL) Test

- Test einer oder mehrere Steuersoftware-Modulen
- Die zu testenden Steuersoftware-Module sind eingebettet in eine Simulation von
  - Kommunikationsschnittstellen
  - Aktuatoren und Sensoren
  - physikalische Umweltmodell
- Einsatzzweck:
  - Modultest



Processor in the Loop (PiL) Test

- Test einer Steuersoftware integriert auf dem µ-Controller
- Die auf dem µ-Controller integrierte Steuersoftware ist eingebettet in eine Simulation von
  - Kommunikationsschnittstellen
  - Aktuatoren und Sensoren
  - physikalische Umweltmodell
- Einsatzzweck:
  - Integrationstest
  - Robustheitstest

Hardware in the Loop (HiL) Test: Simulation

- Test einer Steuersoftware auf der Zielplattform
- Die auf der Zielhardware integrierte Steuersoftware ist über die Hardwareschnittstellen eingebettet in eine Simulation von
  - Aktuatoren und Sensoren
  - physikalische Umweltmodell
- Einsatzzweck:
  - Integrationstests
  - Abnahmetest

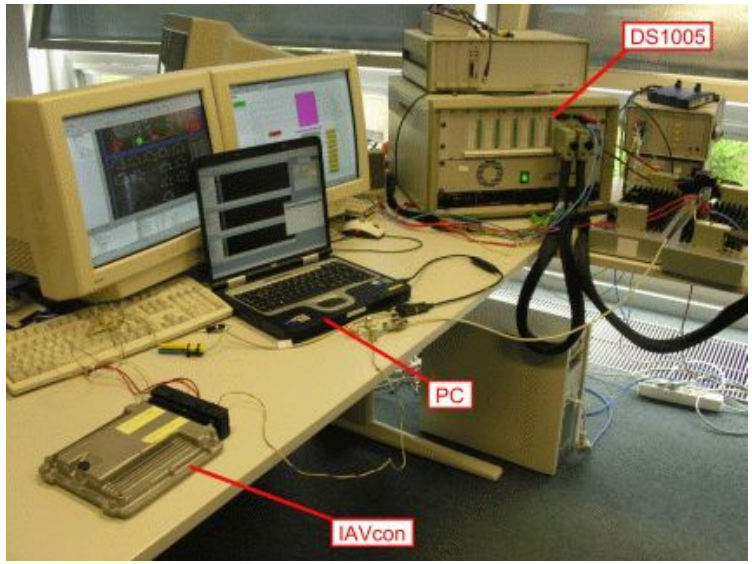
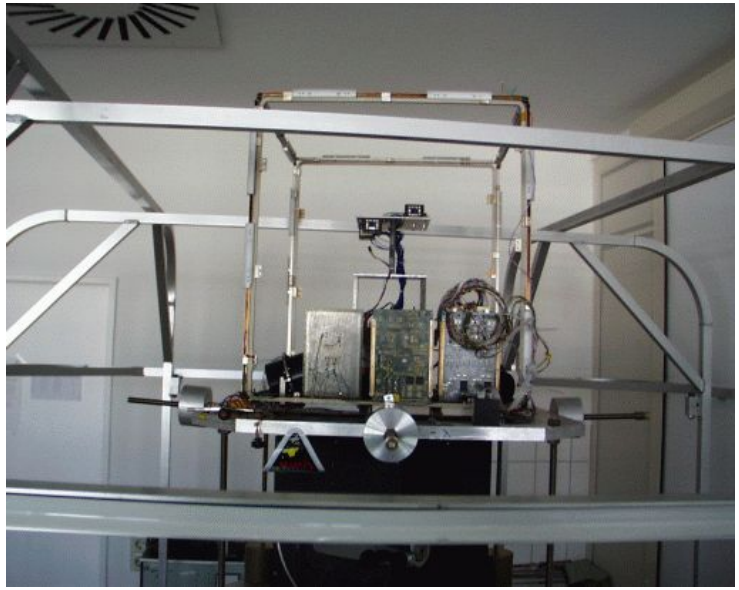


Bild: IAV

Hardware in the Loop (HiL) Test: Engineering Model

- Test einer Steuersoftware auf der Zielplattform
- Die auf der Zielhardware integrierte Steuersoftware ist eingebettet in eine Labormuster des realen Systems

- Einsatzzweck:
  - Integrationstests
  - Abnahmetest



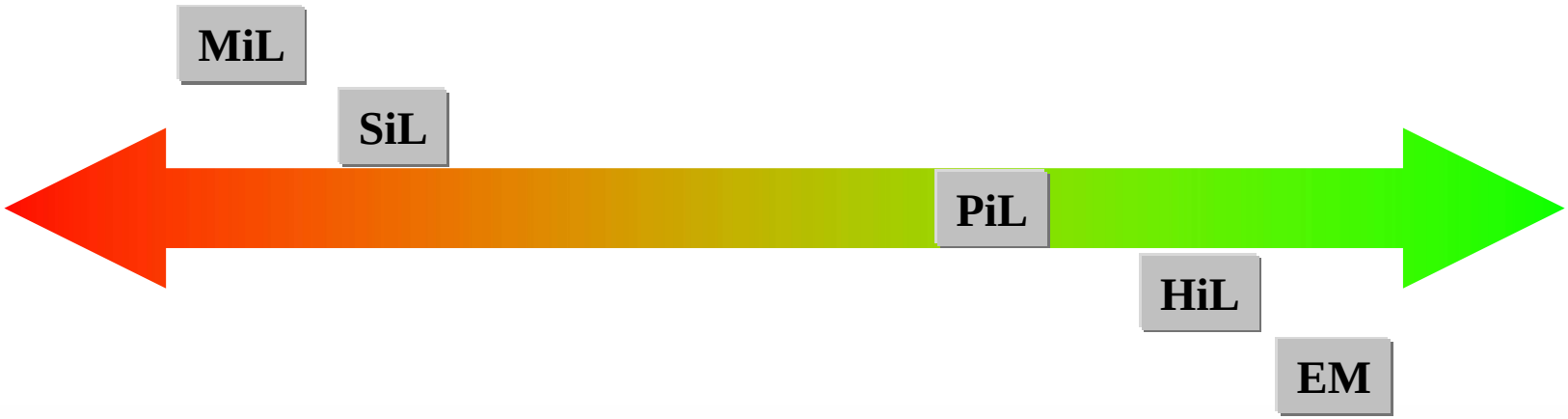
Picture: DLR

Überprüfung der Dynamik

- Alle Testansätze sind zum Test der Dynamik geeignet
- Bei MiL und SiL bestehen Einschränkung durch die fehlende komplette Integration
- Alle Ansätze außer der Test am Engineering Model haben
  - Rundungsfehler
  - Quantisierungsfehler

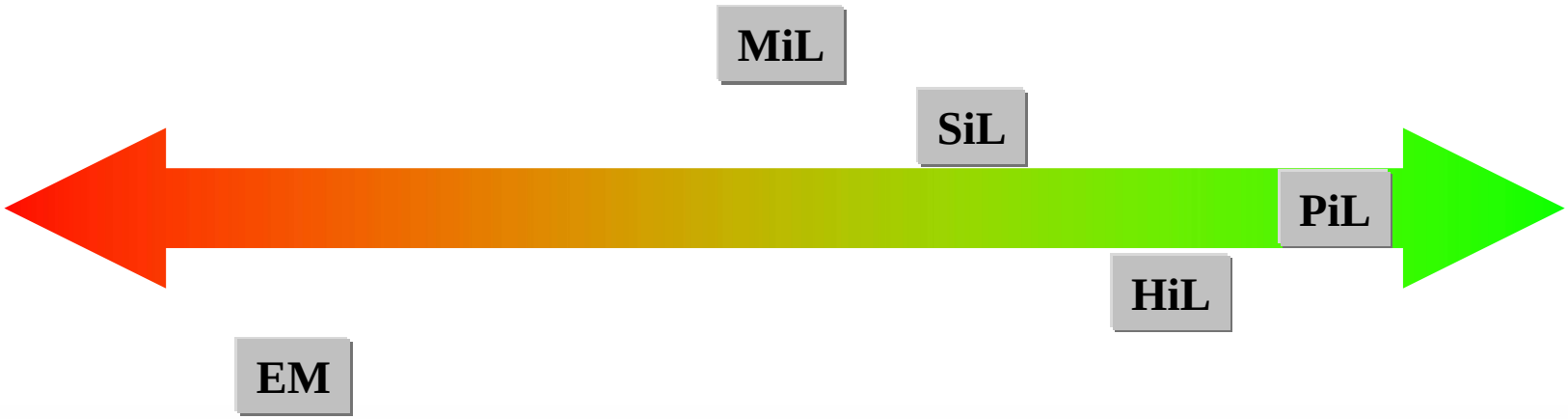
Echtzeitverhalten

- Nur der Test am Engineering Model zeigt das reale Zeitverhalten
- HiL an einer Simulation verlangt Echtzeitfähigkeit der Simulation
- PiL kann Abweichungen im µs-Bereich haben
- SiL und MiL erlauben keine Aussagen zum Echtzeitverhalten



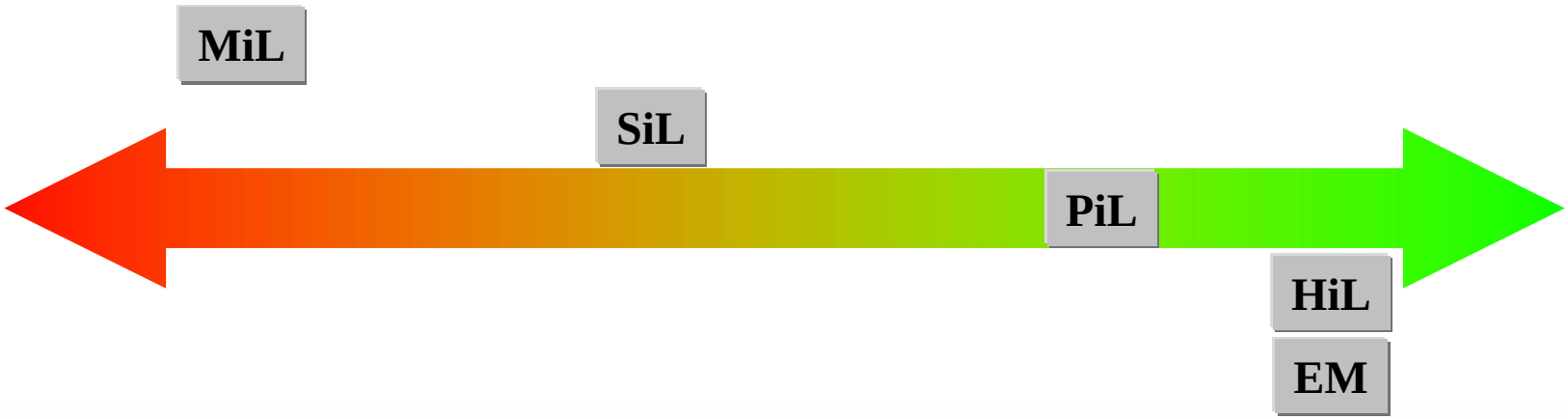
Fehlersimulation

- Nur am Engineering Muster nicht möglich bis auf fehlende Kabelverbindungen
- Bei MiL keine Möglichkeit zur Failure-Injektion im Test-Objekt
- Bei SiL existieren Einschränkungen bezüglich teilintegrierte Systeme
- HiL S. besitzt Einschränkung bezüglich Fehler im Kabelbaum



Test von integrierter Software

- Bei MiL ist Software kein Testobjekt
- SiL wird nur teilintegriert angewandt
- Integration zwischen MiL und SiL Testobjekten möglich
- Bei PiL fehlende Integration der Firmware
- HiL und Engineering Model ohne Einschränkungen

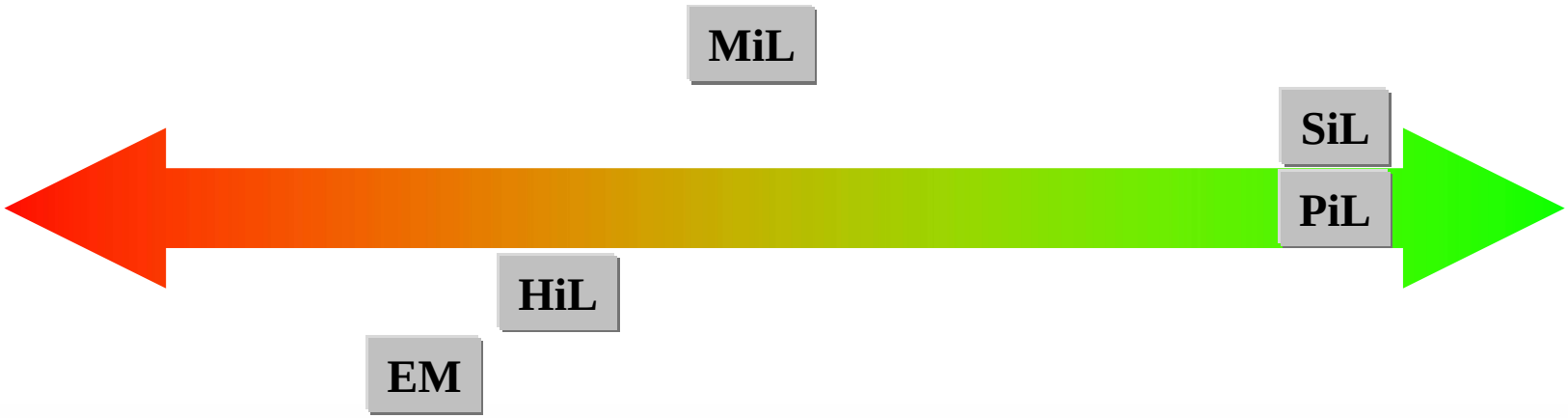


Regessionstests

- Regressionstest erfordern Automatisierung
  - Engineering Model erfordert händisches Einrichten des Teststands
  - HiL mit einer Simulation erfordert vorhergehende Verkabelung und Überprüfung des Teststands
- Keine Einschränkungen bei MiL, SiL und PiL (Nightly Build)

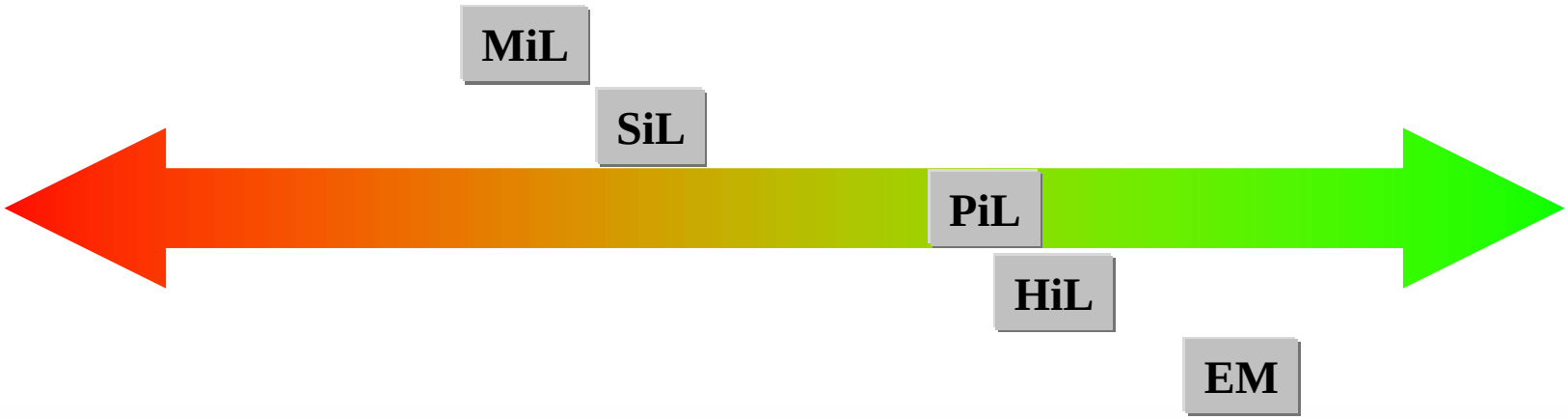
Code Debugging

- Bei HiL Debugging nur über aufgezeichnete Log-Dateien
- Engineering Model besitzt keine geeignete Schnittstelle außer Housekeeping-Daten
- MiL nur Debugging von Algorithmen
- SiL und PiL erlauben jederzeit eine Inspektion des gestoppten Systems

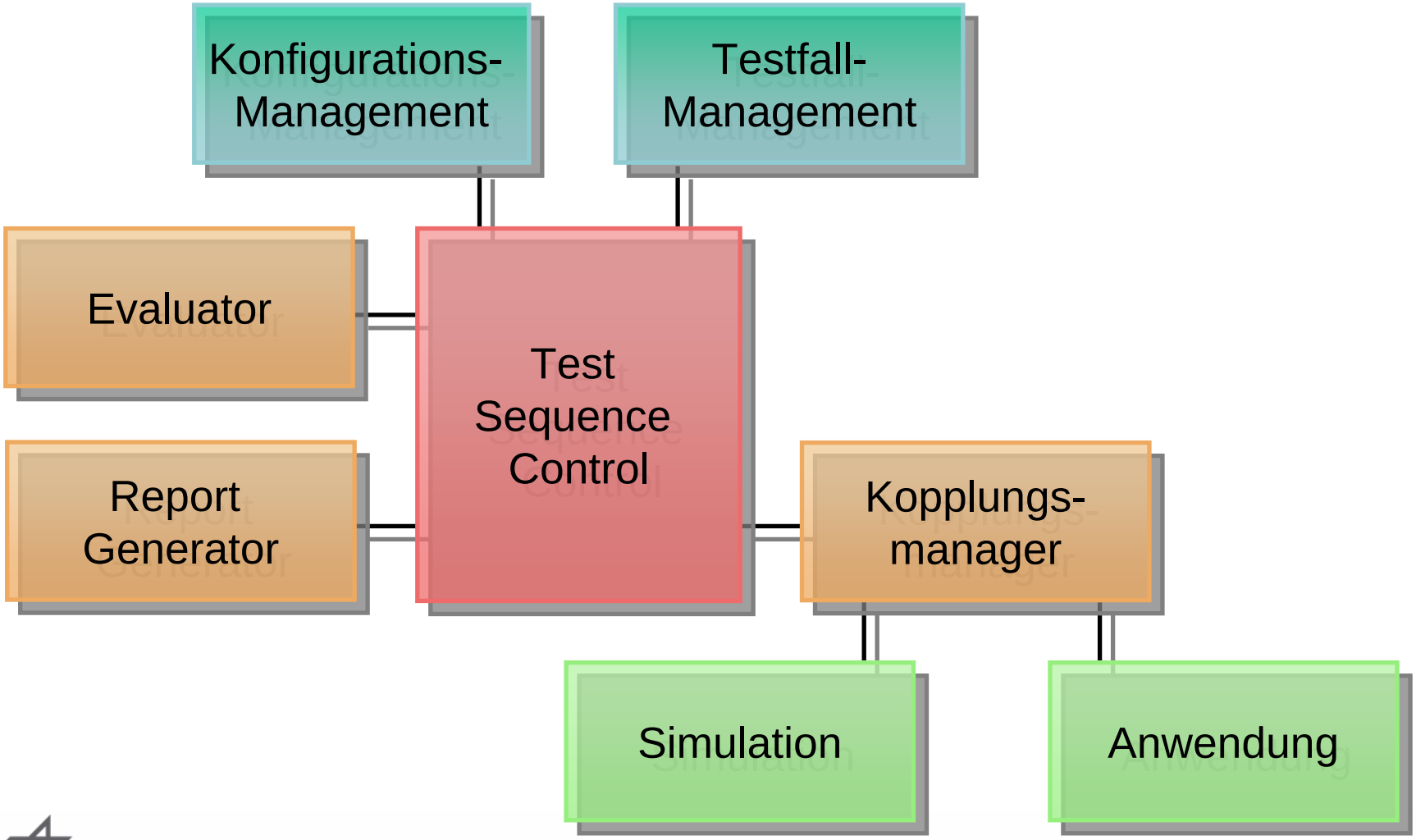


Deckung zwischen Testergebnissen und wirklichen Verhalten

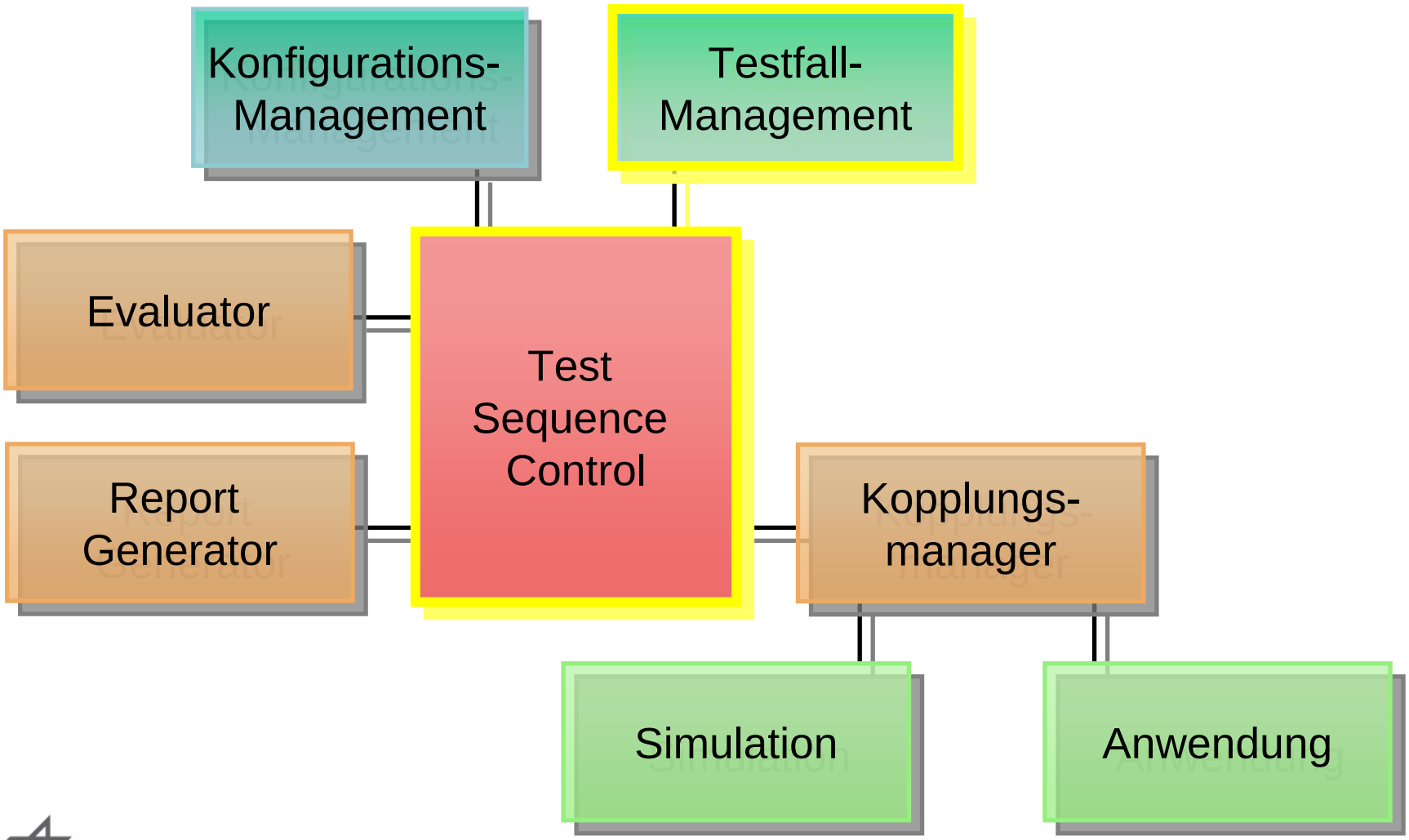
- Kein Testansatz zeigt komplett die Realität
  - EM nutzt idealtypisches System
  - Laborumgebung und Einsatzort sind nicht identisch
    - z.B. Drücke, Gravitation, Luftwiderstand, Magnetfeld, …
  - PiL und HiL besitzen Quantisierungs- und Rechenfehler
  - Fehlendes Messrauschen bei Einsatz von Simulationen



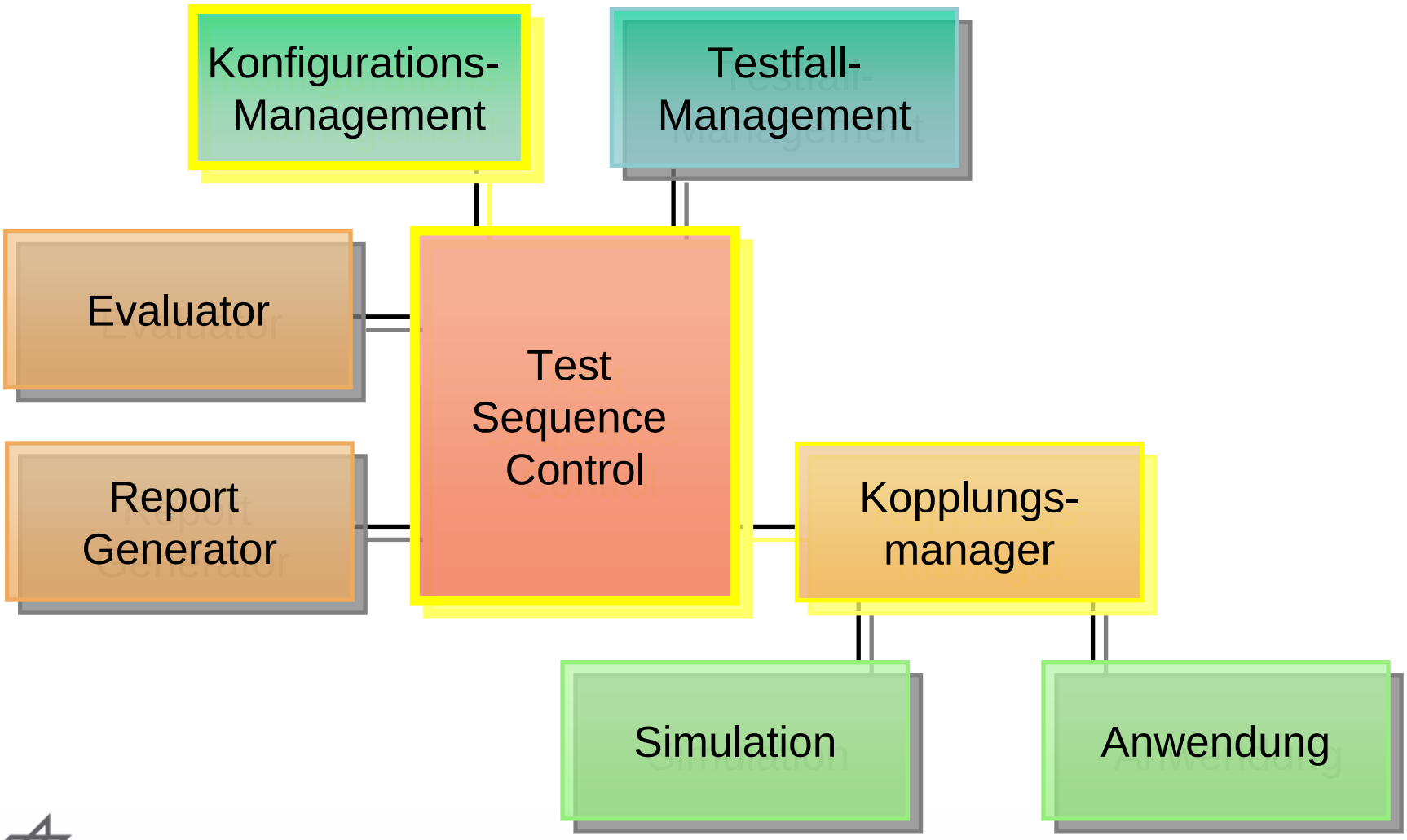
Test-Automation: Test Sequence Control



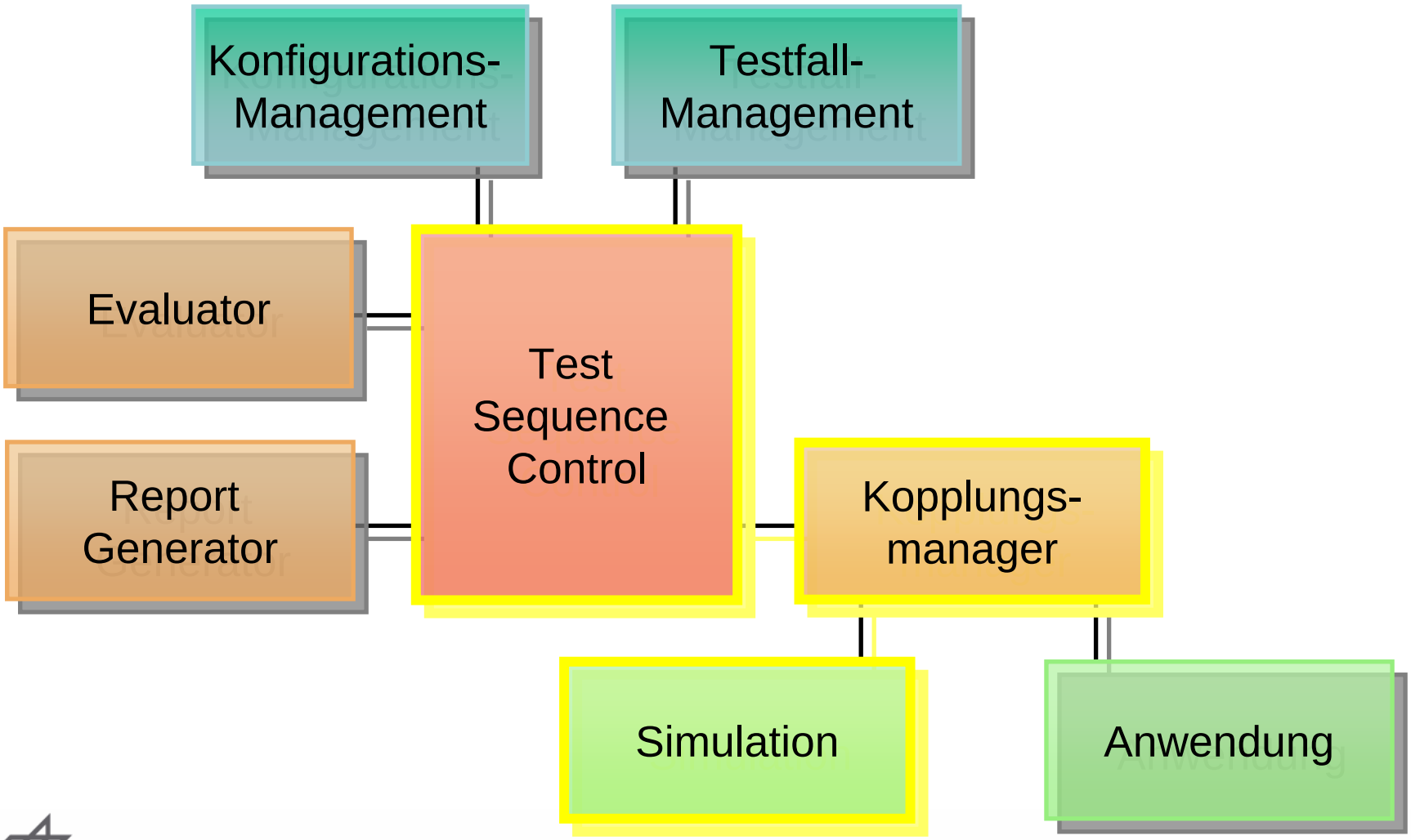
Test-Automation: Testfälle holen



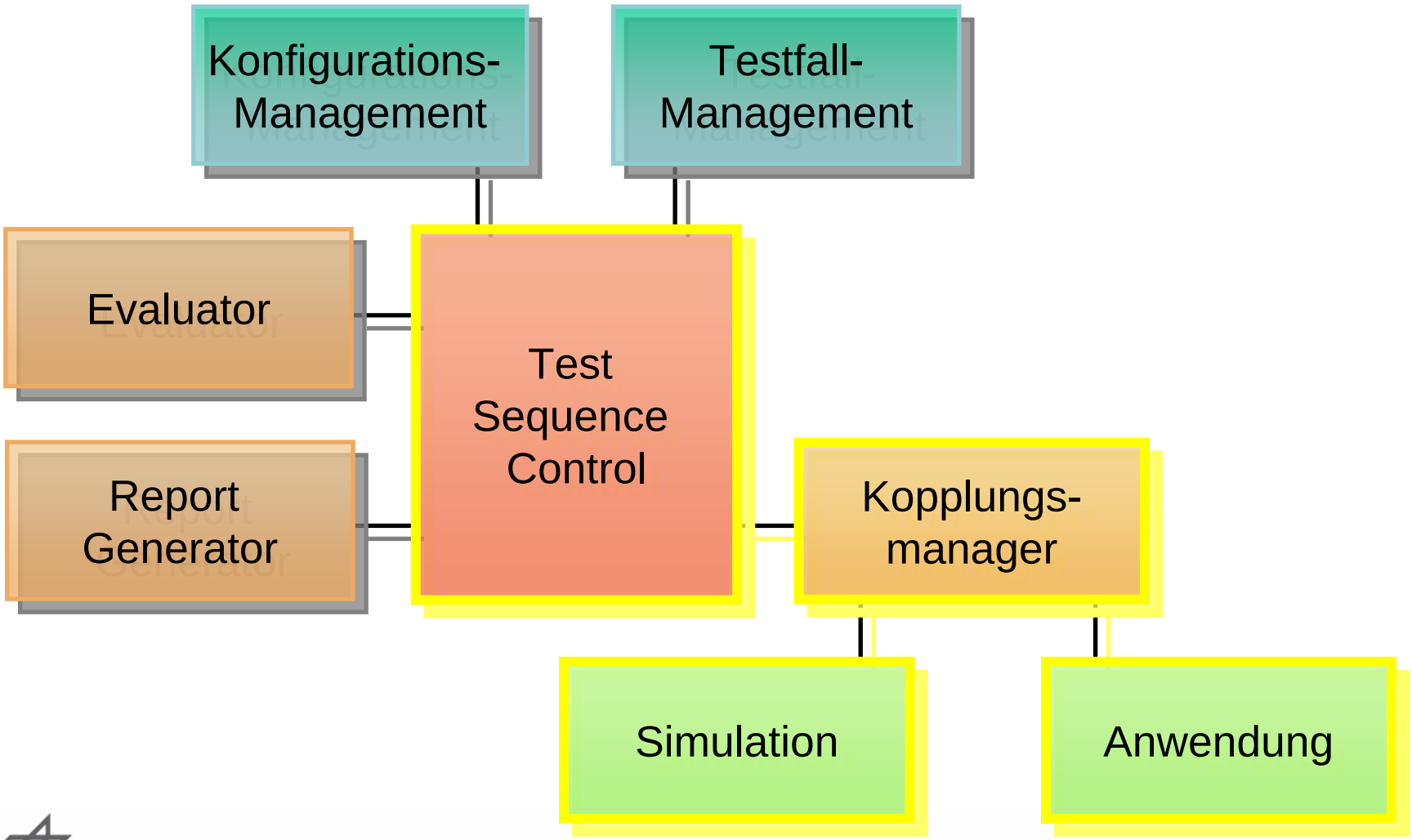
Test-Automation: Initialisierung Testsystem



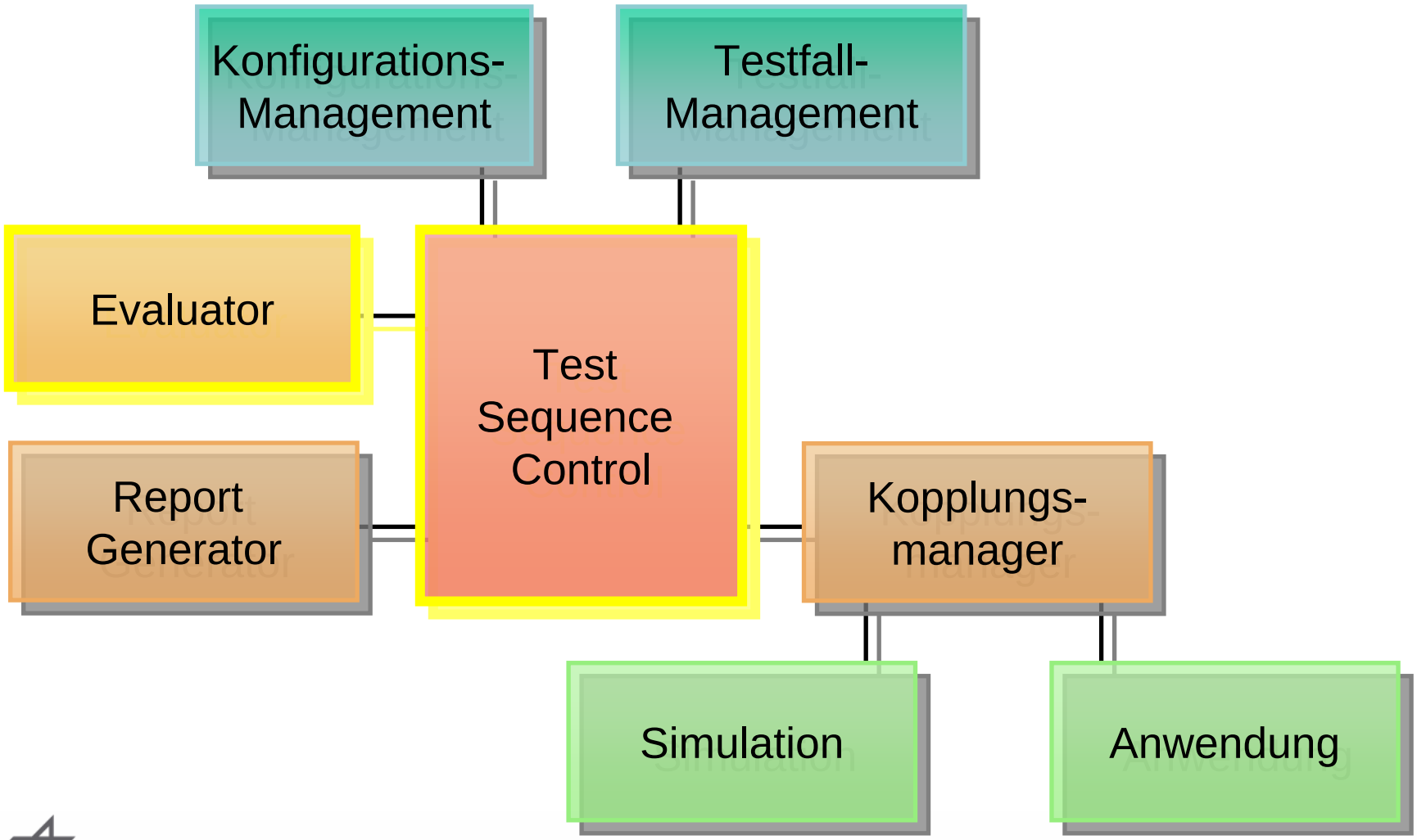
Test-Automation: Kalibration



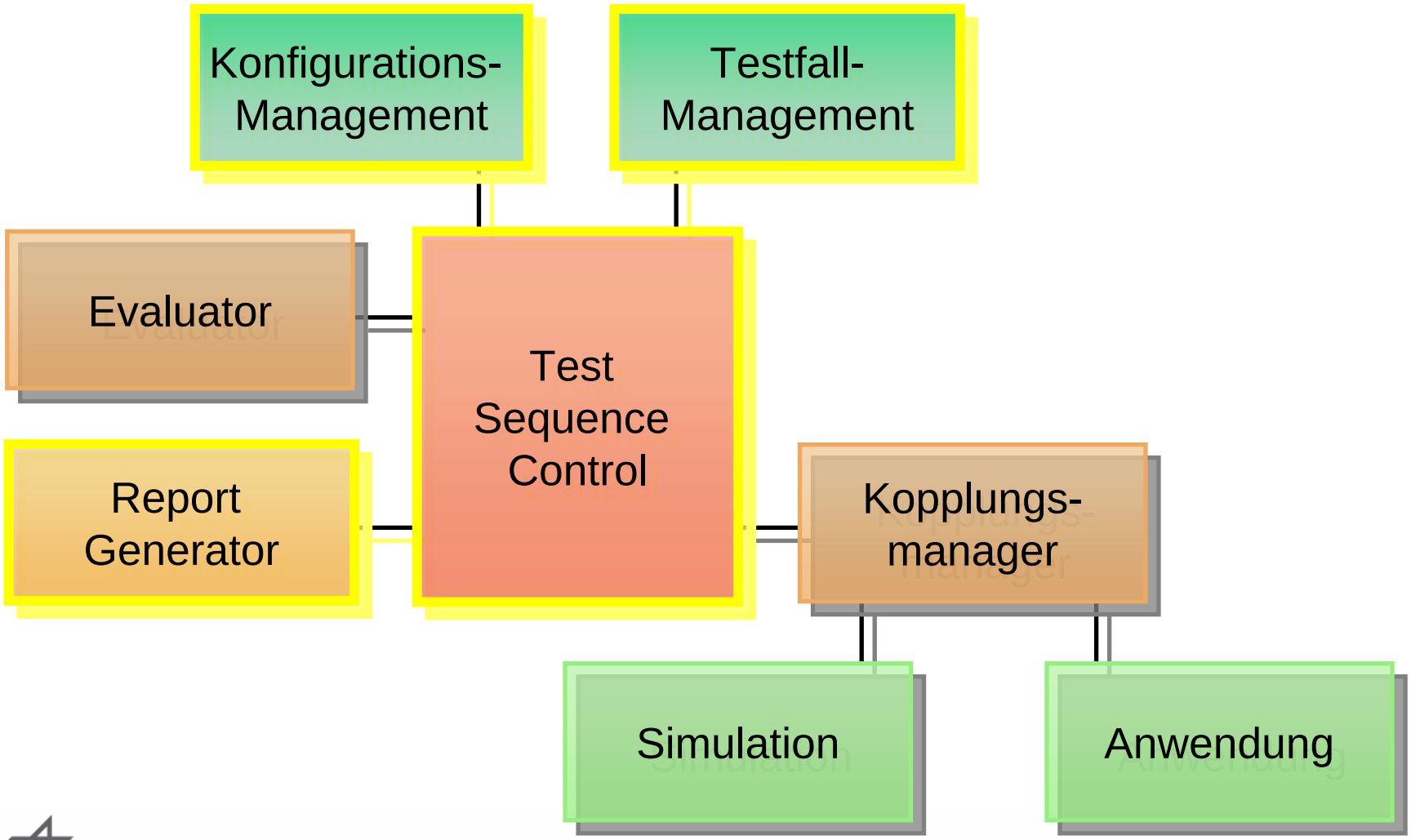
Test-Automation: Test Sequence Control



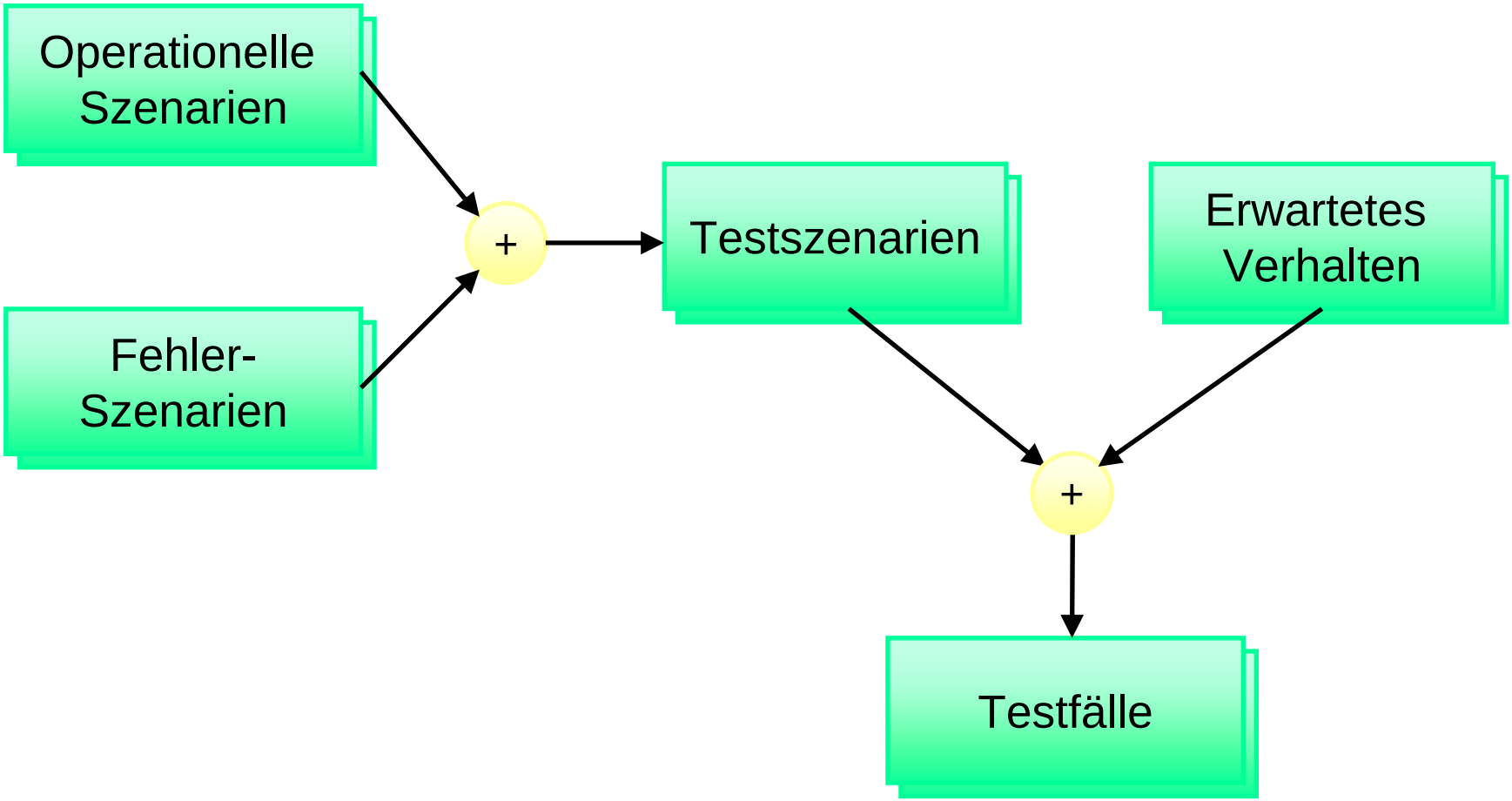
Test-Automation: Testauswertung



Test-Automation: Testabschluss



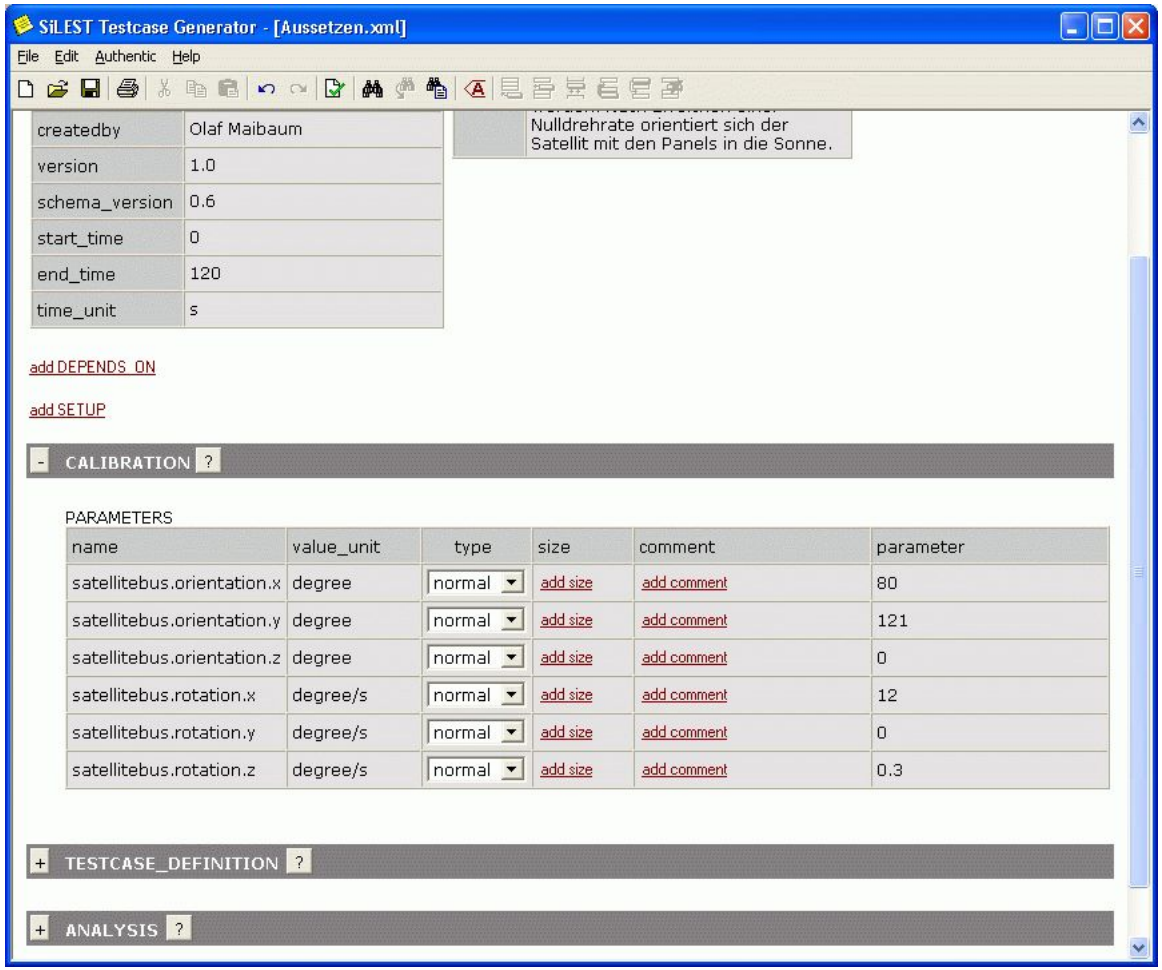
Testfallerstellung



XML-TestfallFormat

- Im Testfall enthalten sind
  - Allgemeine Beschreibung
  - Abhängigkeiten zu anderen Testfällen
  - Globale Einstellungen (z.B. Gleichungslöser in der Simulation)
  - Kalibrationsdaten
  - Testfallbeschreibung
    - Definition von Signalen (analoge und digitale Wertverläufe)
    - Definition von Nachrichten
    - Definition von Fehlertriggern (ersetzen oder modifizieren)
  - Analyse
    - Obere und untere Grenzen
    - Definition von unscharfen Signalverläufen

Testfall-Generator

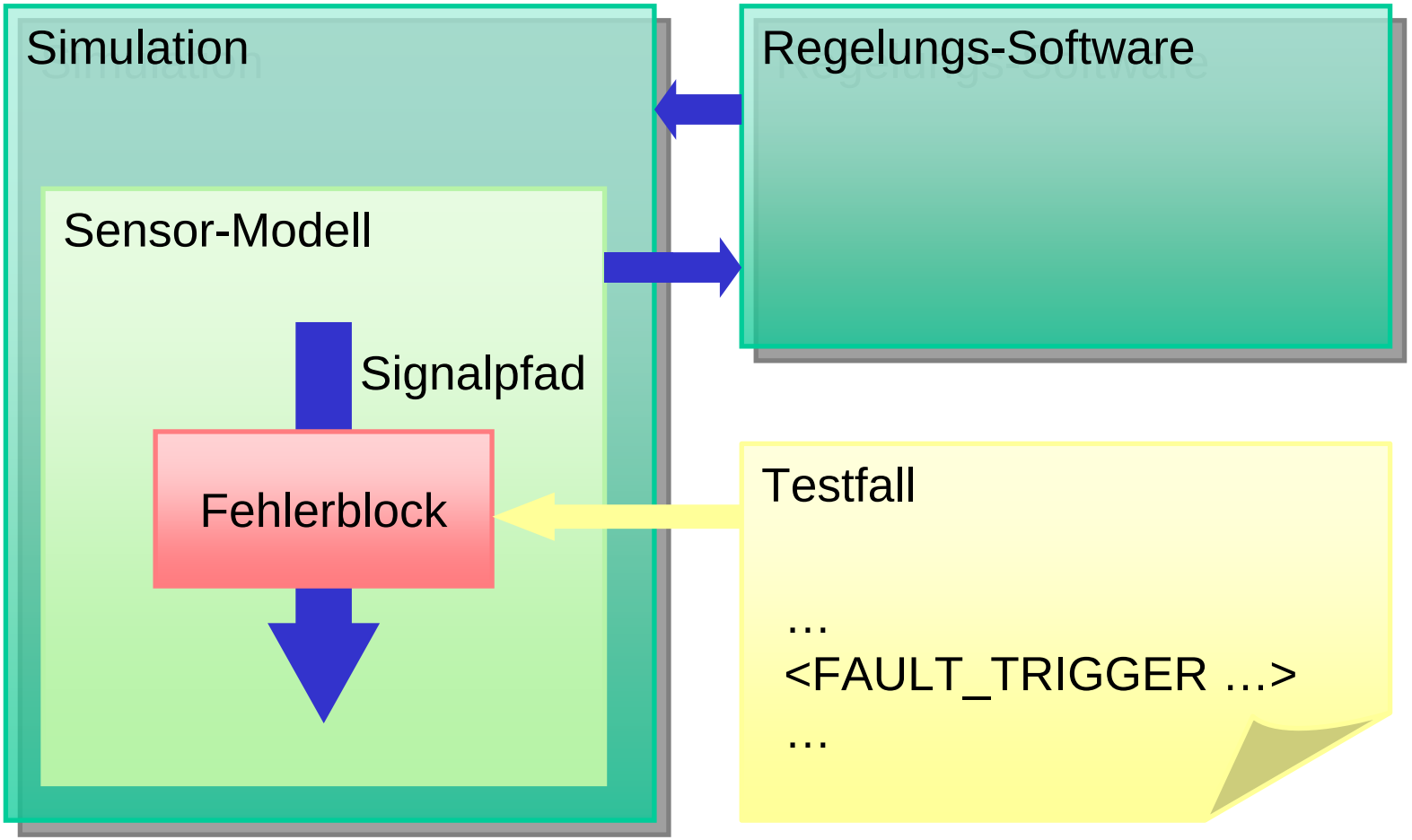


Umgebungssimulation

- Bestandteile der Simulation sind
  - Dynamikmodell des Systems
  - Aktuator-Modelle
  - Sensor-Modelle
- Fehlersimulationen ist Bestandteil der Aktuator und Sensor-Modelle

- Wiederverwendung von Simulationsmodellen
  - Simulationsmodell-Repository
  - Dokumentation der Modelle und ihrer Schnittstellen
  - Dokumentation der Modellverwendung, -verifikation und Pflege

Fehlersimulation



Rolle des Systems-Engineering

- Systems-Engineering liefert ein Strukturmodell des Systems
  - Beschreibung der Hardware-Komponenten
  - Eigenschaften der Hardware-Komponenten
  - Beziehung zwischen den Hardware-Komponenten

- Das Strukturmodell wird verwendet für
  - Auswahl der Simulationsmodelle aus dem Simulationsmodell-Repository
  - Verknüpfung und Kalibrierung der Simulationsmodelle
  - Durchführung einer FMEA
    - Auswahl von kritischen Komponenten zur Testfallerstellung

Zusammenfassung

- Jeder Testansatz hat sein Einsatzgebiet
  - MiL wird zum Test der Algorithmen angewandt
  - SiL dient dem Modultest und dem Test von teilintegrierten Software
  - PiL dient dem Test und Debugging integrierter Software
  - HiL dient dem Akzeptanztest mit harten Zeitanforderungen
- HiL kann durch andere Testansätze nicht ersetzt werden
- Mit PiL kann die Robustheit eines Systems gegenüber Hardware-Defekten getestet werden
- Automatisierte Regressionstests sind bis einschließlich PiL möglich

- Aufbau einer Simulation ist Kostentreiber für „in the Loop“-Tests
- „In the loop“-Tests benötigen eine Simulationskultur im Unternehmen

- Verbundprojekt zwischen
  - Deutsches Zentrum für Luft und Raumfahrt e.V. (DLR)
  - Ingenieurgesellschaft Auto und Verkehr GmbH (IAV)
  - Technische Universität Berlin
  - Webdynamix GmbH
  - Fraunhofer FIRST

- Projektziel war
  - Definition und Anwendungserprobung eines SiL Testprozesses

- Gefördert durch das Bundesministerium für Bildung und Forschung in der Forschungsoffensive „Software Engineering 2006“

Diskussion

www.silest.de