



SiLEST

**Entwicklung und Erprobung von Methoden und Werkzeugen
für den Test und die Sicherheitsanalyse eingebetteter
Echtzeitsoftware**



SiLEST – **S**oftware **i**n the **L**oop for **E**MBEDDED **S**oftware **T**EST

- ▶ **Laufzeit: 01.01.2004 – 31.12.2006**
- ▶ **Ziel: Entwicklung und Erprobung von Werkzeugen für den Test und die Sicherheitsanalyse eingebetteter Echtzeitsoftware**
- ▶ **Verbundpartner:**
 - **Deutsches Zentrum für Luft- und Raumfahrt e.V., Simulations- und Softwaretechnik (DLR, SISTEC)**
 - **Fraunhofer Gesellschaft zur Förderung der angewandten Forschung e.V., Institut für Rechnerarchitektur und Softwaretechnik (FhG FIRST)**
 - **Ingenieurgesellschaft Auto und Verkehr GmbH (IAV)**
 - **webdynamix GmbH**
- ▶ **Fördermittel: 1.485.316 €**
- ▶ **Gesamtmittel: 1.991.540 €**



Übersicht

1. Motivation zur Nutzung einer SiL-Testumgebung
2. Ziele von SiLEST
3. Testansatz
 - Schichtenmodell einer SiL-Testumgebung
 - Virtuelle Simulationszeit
 - Testmethodik
4. Zusammenfassung



Motivation (1/2)

- ▶ Einsatz von Software in kritischen Umgebungen
- ▶ Überprüfung der softwarebeeinflussten Systemsicherheit
 - ⇒ Anforderungen an die Software in eingebetteten Systeme
 - Überprüfen der Funktionsfähigkeit von Sensoren/Aktuatoren
 - Tolerant gegenüber Alterungserscheinungen von Sensoren/Aktuatoren
 - Robust gegen Sensor-/Aktuatorausfälle



Motivation (2/2)

- ▶ **Test der Software muss im Regelkreis geschehen**
- ▶ **Die Robustheit einer Software muss nachgewiesen werden**
- ▶ **Nachteile einer Hardware in the Loop Testumgebung**
 - **Unflexibel, fehleranfällig**
 - **Teuer im Betrieb/Aufbau**
 - **Keine Debuggingmöglichkeiten**

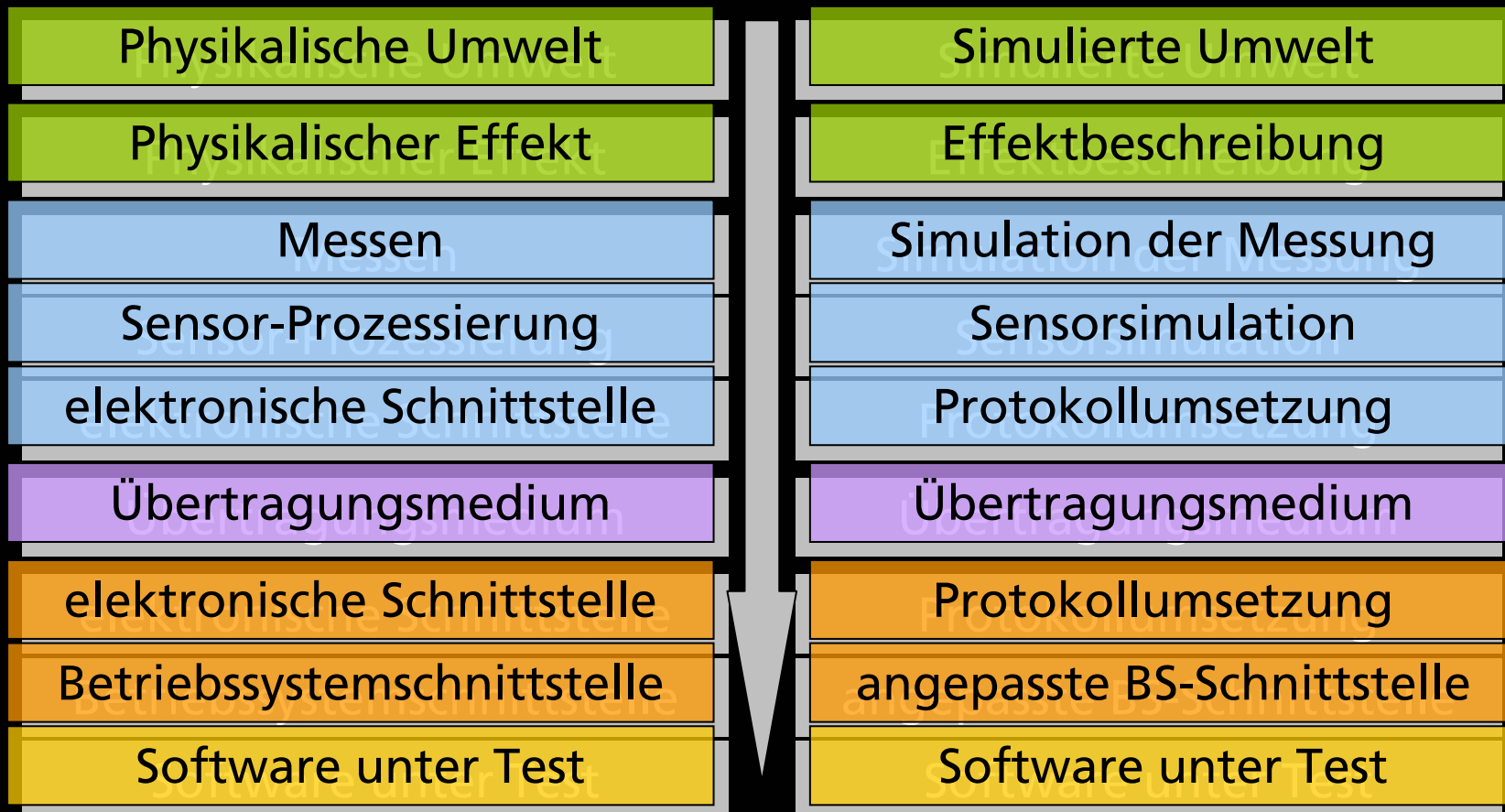


Ziele von SiLEST

- ▶ **Verallgemeinerter Testprozess für SiL-Softwaretests**
- ▶ **Tailoringregeln für Anwendungsdomänen (Automobil, Raumfahrt)**
- ▶ **Verallgemeinerte Testumgebung**
- ▶ **Schnittstellenspezifikation für Simulationsmoduln**
- ▶ **Vorbereiten der Standardisierung der Schnittstellen für die Sensor- und Aktuator-Simulationsmoduln**
- ▶ **Generierung von Testfällen aus Ausfallszenarien**
- ▶ **Automatisierung des Softwaretests eingebetteter Systeme**
- ▶ **Abgrenzen der Einsatzmöglichkeiten HiL ↔ SiL-Softwaretests**



Schichtenmodell eines Sensors und einer SiL-Testumgebung





Virtuelle Simulationszeit

- ▶ Grundlage: **Die Zeit ist nur sichtbar zu Ein- und Ausgabeoperationen**
- ▶ Die Software unter Test sieht nicht die Realzeit, sondern eine virtuelle Zeit
- ▶ während der Ein- und Ausgabeoperationen:
 - Synchronisation zwischen Software unter Test und Simulation
 - Korrektur der Board-Zeit
- ▶ Scheduling im eingebetteten System läuft wie im realen System
- ▶ Simulation muss keine Echtzeitanforderungen erfüllen
- ▶ Start/Stop-Betrieb der Software unter Test und Simulation ist möglich

- ▶ Geringfügige zeitliche Änderungen durch Pipelining- und Caching-Effekte



Testmethodik

- 1. Laden eines Zustands der Software unter Test und der Simulation oder Initialisierung für einen Neustart**
 - 2. Modifikation der Sensor/Aktuatorsimulation durch Einschalten und Ausschalten von Fehlerfällen in der Simulation**
 - 3. Betrieb bis zu einem vorgegebenen Zeitpunkt**
 - 4. Speichern des Zustands der Software unter Test und der Simulation**
- ▶ Testfälle werden bestimmt durch Nominal-/Fehlerszenarien der umgebenden Hardware**
 - ▶ Betriebs- und Fehlerszenarien lassen sich baumartig aufbauen**
 - ▶ Nachweis der Robustheit der Software unter Test ist möglich**



Zusammenfassung

▶ Ergebnisse von SiLEST:

- ✓ **Anpassbarer SiL-Testprozess für Software eingebetteter Systeme ausgehend von Fehlerfällen der umgebenden Hardware**
- ✓ **Abgrenzung der Einsatzmöglichkeiten zwischen HiL- und SiL-Tests**
- ✓ **Spezifikation von Simulationsmodul-Schnittstellen der Umgebungssimulation**
- ✓ **Repository mit standardisierten Simulationsmoduln**
- ✓ **Verbesserte Debuggingmöglichkeiten zur Fehlersuche/-korrektur**
- ✓ **Nachweis der Robustheit gegenüber Sensor-/Aktuatorausfällen**