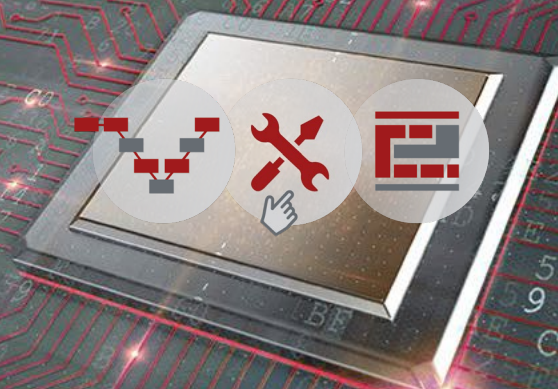


ARAMiS II Abschlussveranstaltung
20.09.2019 Stuttgart



Modellbasierte Multicore-Softwareentwicklung

Die ARAMiS II Entwicklungsprozesse

Stefan Kunz, Continental und Timo Sandmann, KIT

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Die Bereitstellung eines systematischen und strukturierten Ansatzes zur Entwicklung von Multicore Software und Plattformen („Strukturierter Multicore Entwicklungsprozess“)

„... Die in TP2 entwickelte Methodik wird dabei von allen Arbeitspaketen in TP3 in den dort betrachteten Teilschritten angewendet bzw. implementiert. Ebenso fließen sowohl Erkenntnisse aus den Arbeitspaketen von TP3, TP4 und TP5 in die Weiterentwicklung der übergeordneten Methodik ein wie auch Informationen über die Architekturen in Form von Modellen ...“

(VHB ARAMiS II)

STRUKTURIERTER MULTICORE ENTWICKLUNGSPROZESS

Bereitstellung eines systematischen und strukturierten Ansatzes zur Entwicklung von Multicore Software und Plattformen



NEUE INDUSTRIELLE PLATTFORMEN



Entwicklung und Erweiterung von etablierten industriellen Plattformen unter Berücksichtigung Multicore spezifischer Anforderungen.



NEUE METHODEN UND WERKZEUGE FÜR DEN ENTWICKLUNGSPROZESS

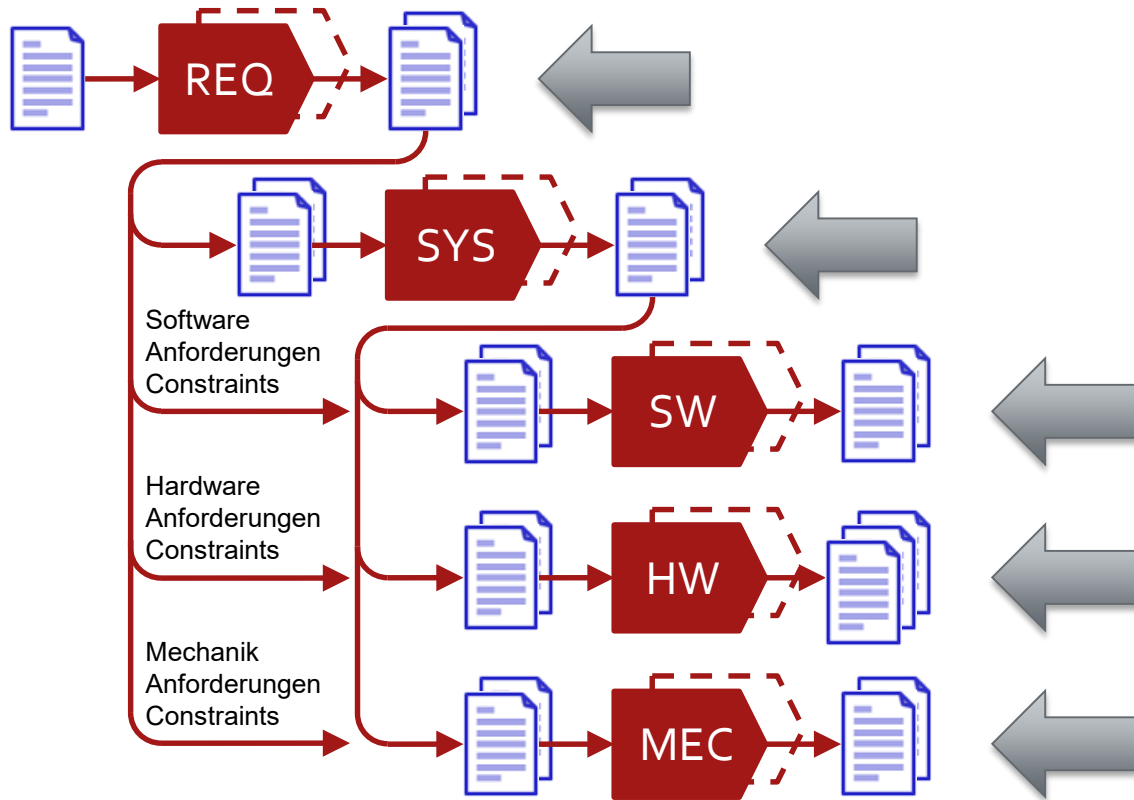
Entwicklung von Methoden und Werkzeugen, welche den strukturierten Multicore Entwicklungsprozess unterstützen

Generischer Entwicklungsprozess – Motivation





- Generischer ARAMiS II-Entwicklungsprozess **und** Methoden
- V-Modell Stand der Technik in der Industrie
- Internationale Standards
 - Automotive: ISO 26262 „Functional Safety for Road Vehicles“
 - Avionik: DO 178C „Software Considerations in Airborne Systems and Equipment Certification“
 - IEC 61508 „Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems“
 - ISO 15288 „Systems and software engineering – System life cycle process“
 - ISO 12207 „Systems and software engineering – Software life cycle“

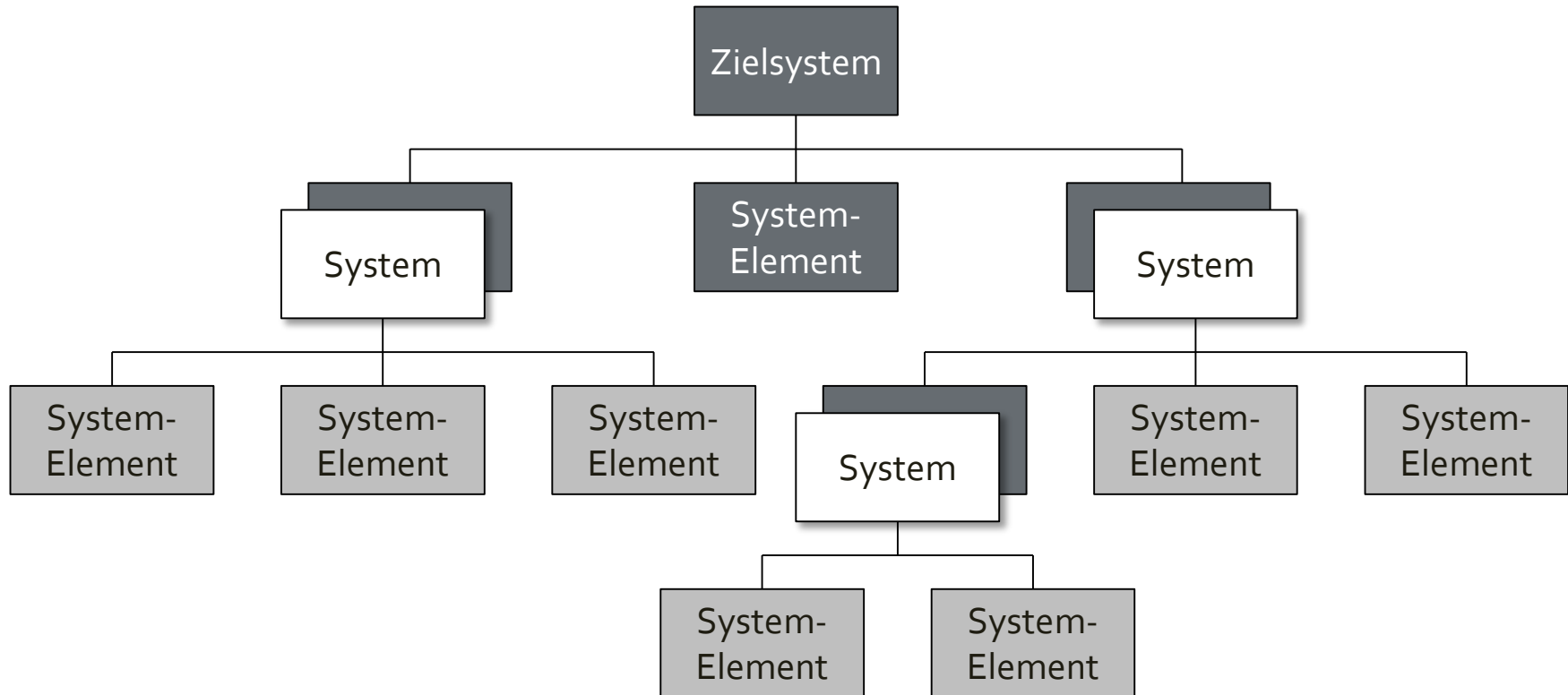
Welche **Granularität** der Komponenten eines Softwaresystems ist für die **Ausnutzung** der Leistungsfähigkeit einer Multicore-basierten Plattform erforderlich?

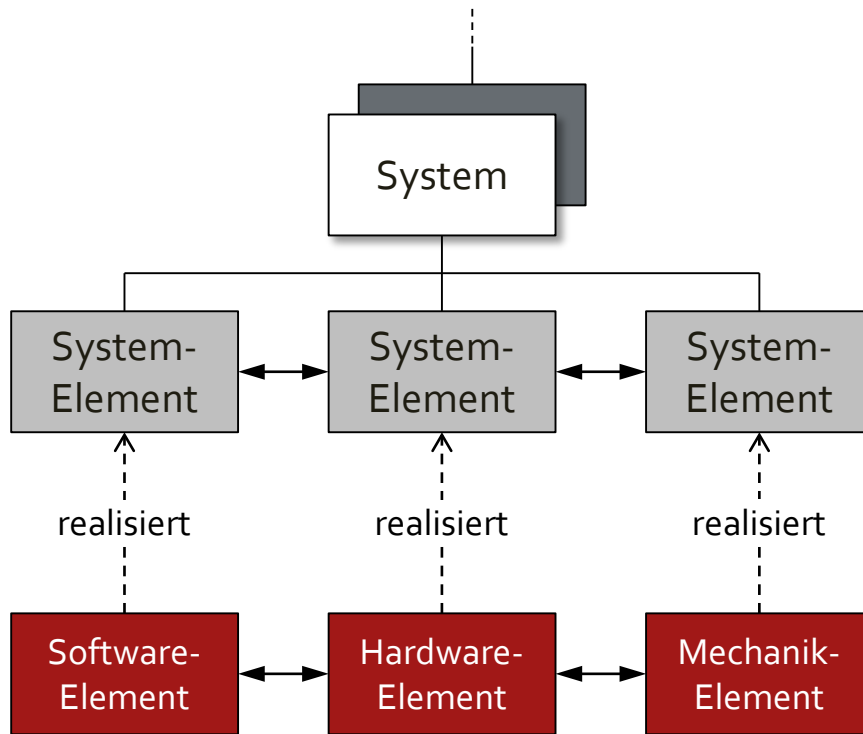
Generischer Entwicklungsprozess – Überblick



REQ	Anforderungen
SYS	Systemarchitektur
SW	Software
HW	Hardware
MEC	Mechanik

-  Task
-  Verifikation und Validierung VV
Zertifizierung
-  Artefakt /
Ausgabeprodukt
-  Validierung des "rechten
Flügels" des V-Modells





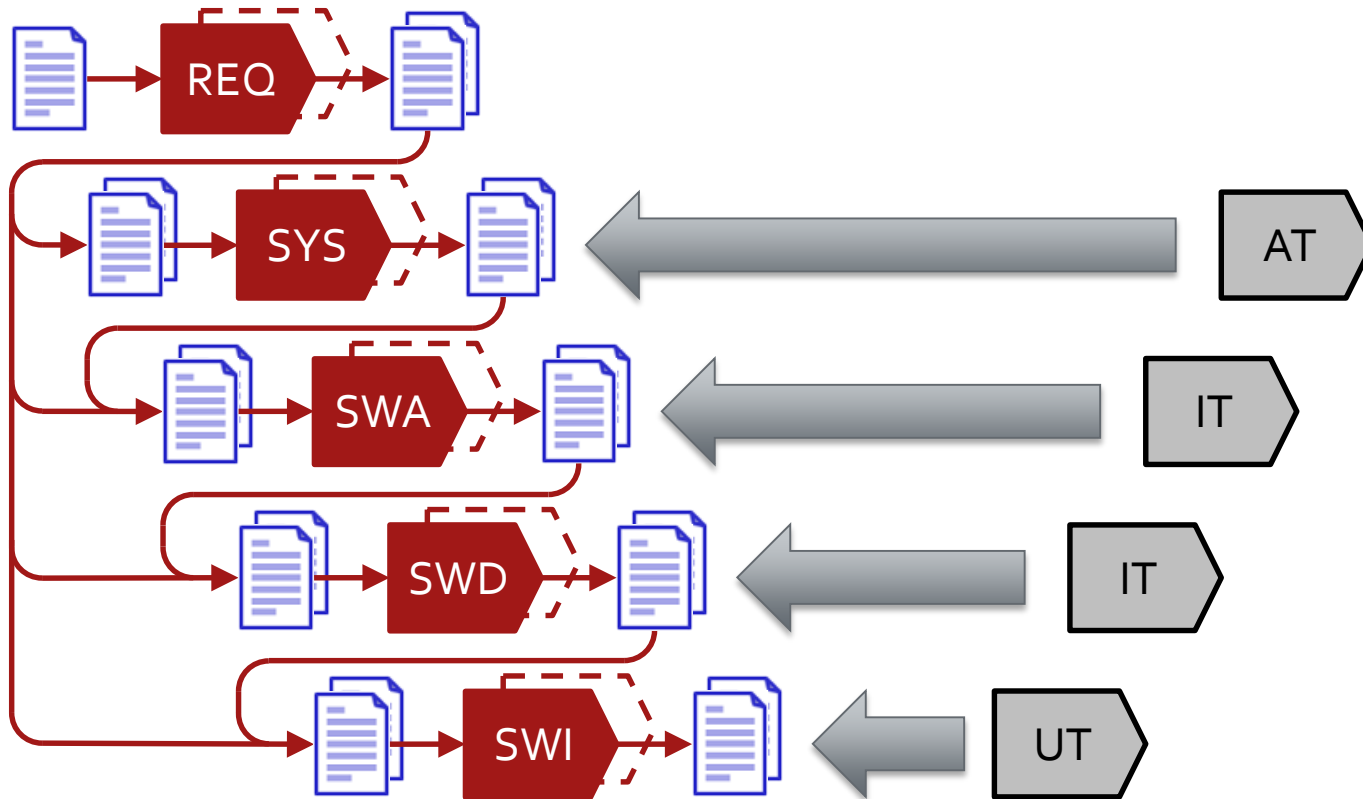
Entscheidung, welche Elemente des Systems realisiert werden durch:

- Software – Software-Plattformen
- Hardware – **Multicore**-basierte Plattformen
- Mechanik

unter der Berücksichtigung von funktionaler Sicherheit, Security, Zeitanforderungen, Performanzanforderungen, usw.

SW und HW Fähigkeiten müssen zur Analysemöglichkeit durch Modelle beschrieben werden

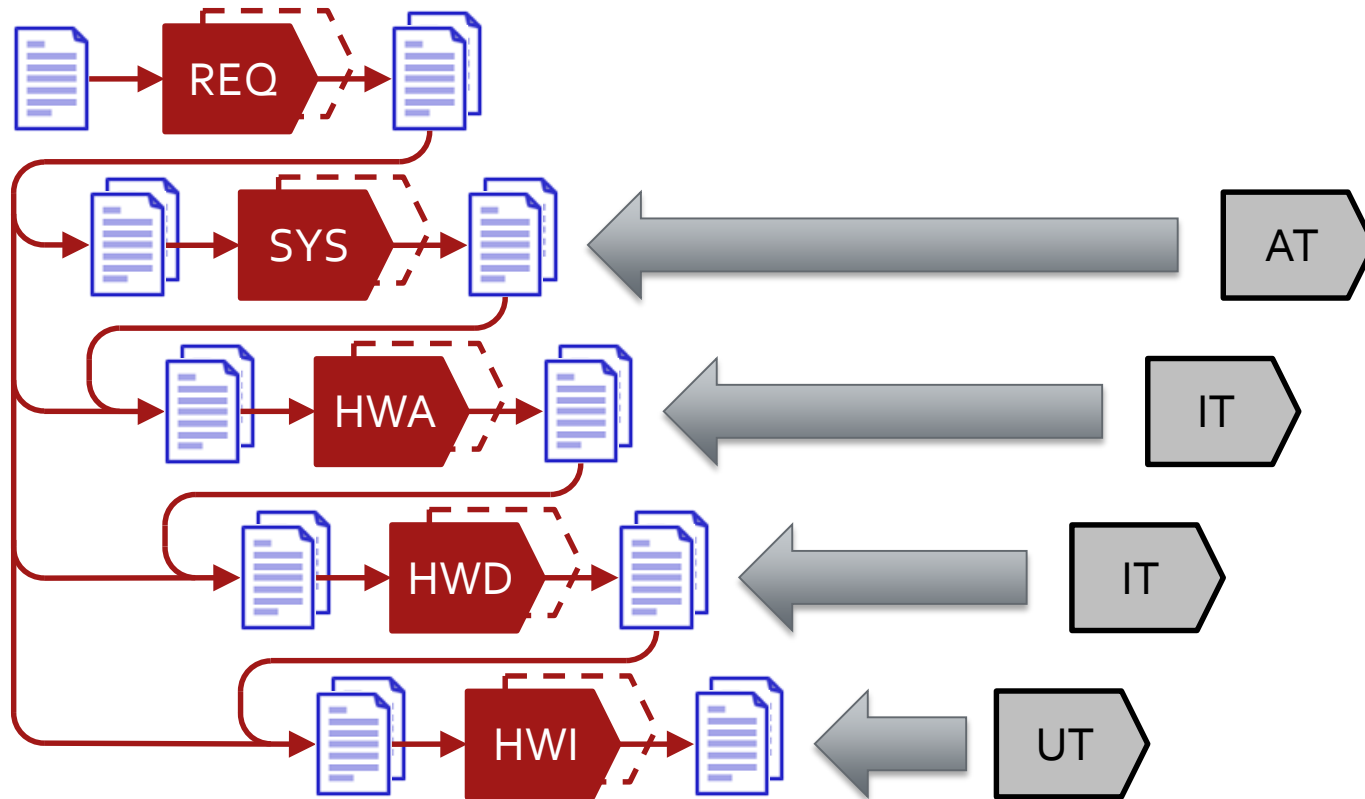
Generischer Entwicklungsprozess – Software



- Anforderungen
- Modelle

REQ	Anforderungen
SYS	Systemarchitektur
SWA	Software-Architektur
SWD	Software-Design
SWI	Software-Implementierung
UT	Unit-Tests
IT	Integrations-Tests
AT	Akzeptanz-/System-Tests

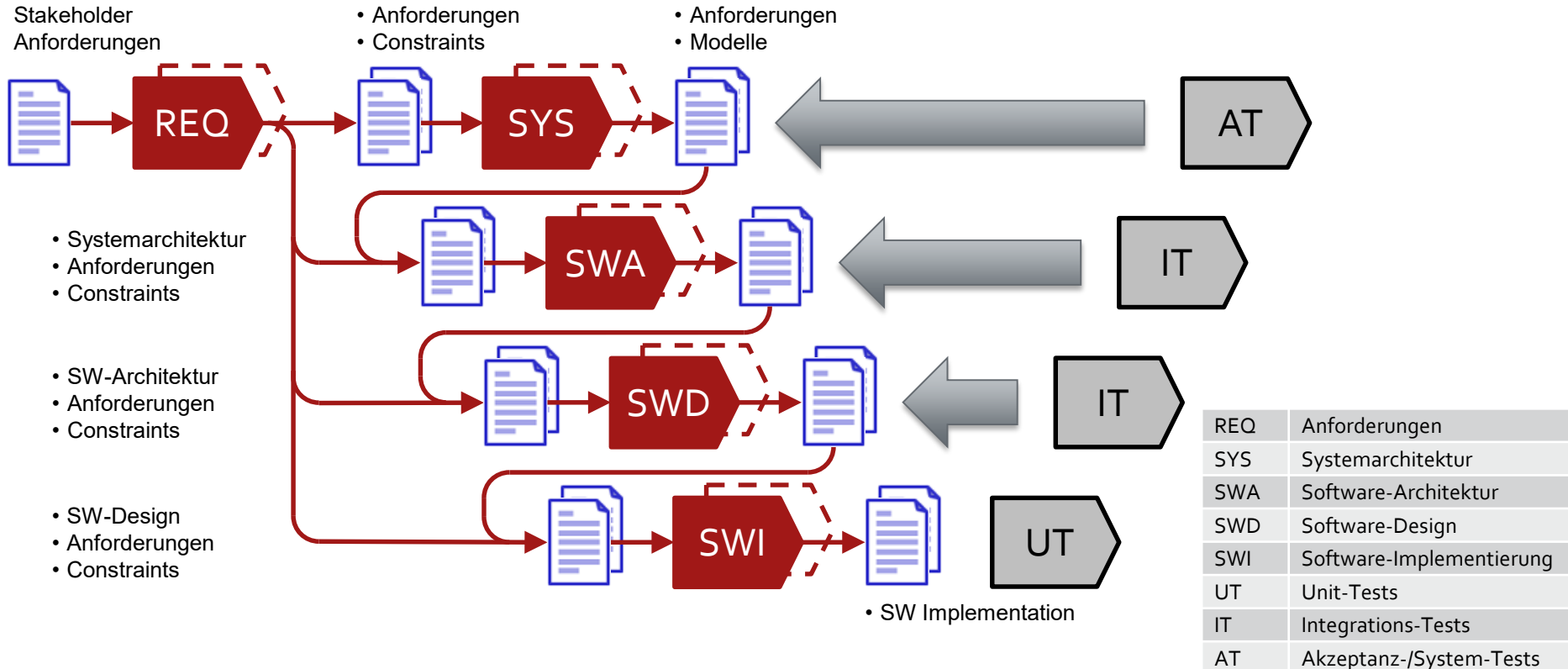
Generischer Entwicklungsprozess – Hardware



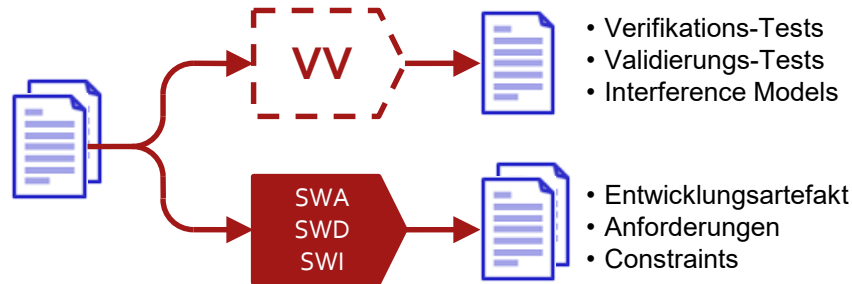
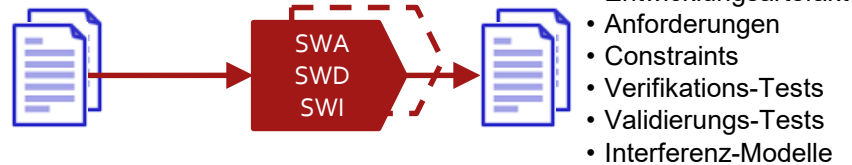
- Anforderungen
- Modelle

REQ	Anforderungen
SYS	Systemarchitektur
HWA	Hardware-Architektur
HWD	Hardware-Design
HWI	Hardware-Implementierung
UT	Unit-Tests
IT	Integrations-Tests
AT	Akzeptanz-/System-Tests

Generischer Entwicklungsprozess




- Entwicklungsartefakt
- Anforderungen
- Constraints




Architektur, Design und Implementierung:

- Dekomposition
- Komposition
- Partitionierung
- Parallelisierung
- Deployment

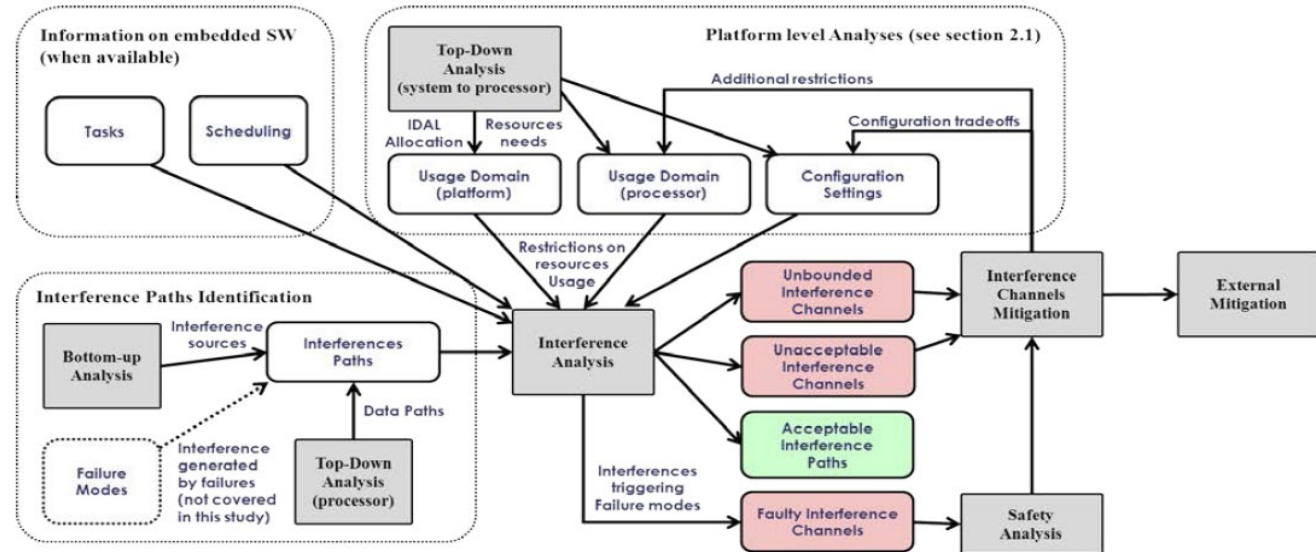
 Task

 Verifikation und Validierung VV
Zertifizierung

 Artefakt /
Ausgabeprodukt

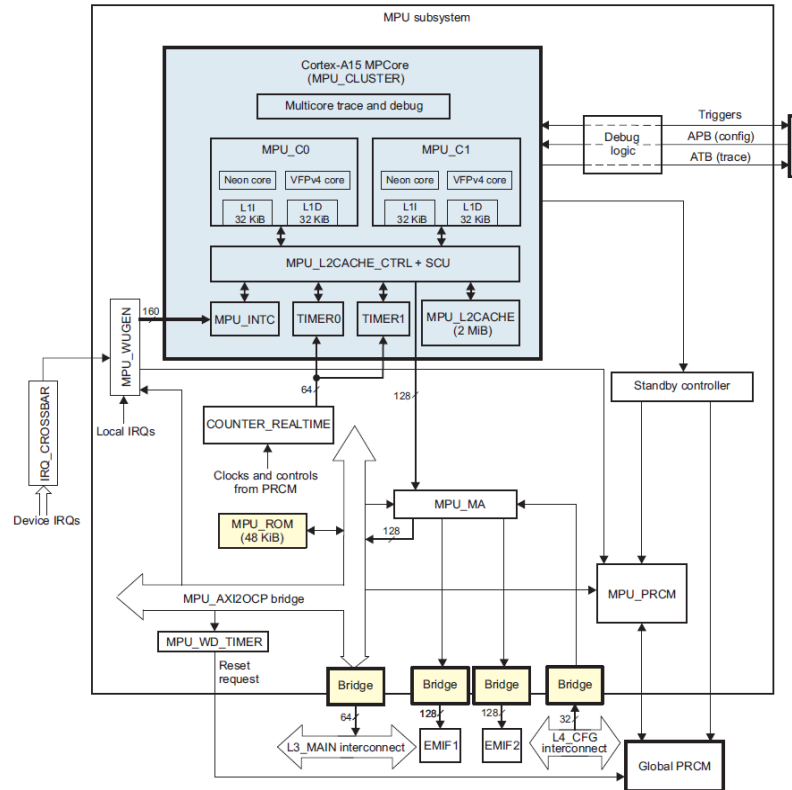
Modelle und Interferenz-Modelle

- Hardware- und Software-Modelle
- Interferenz-Modelle
- Fehlermodelle

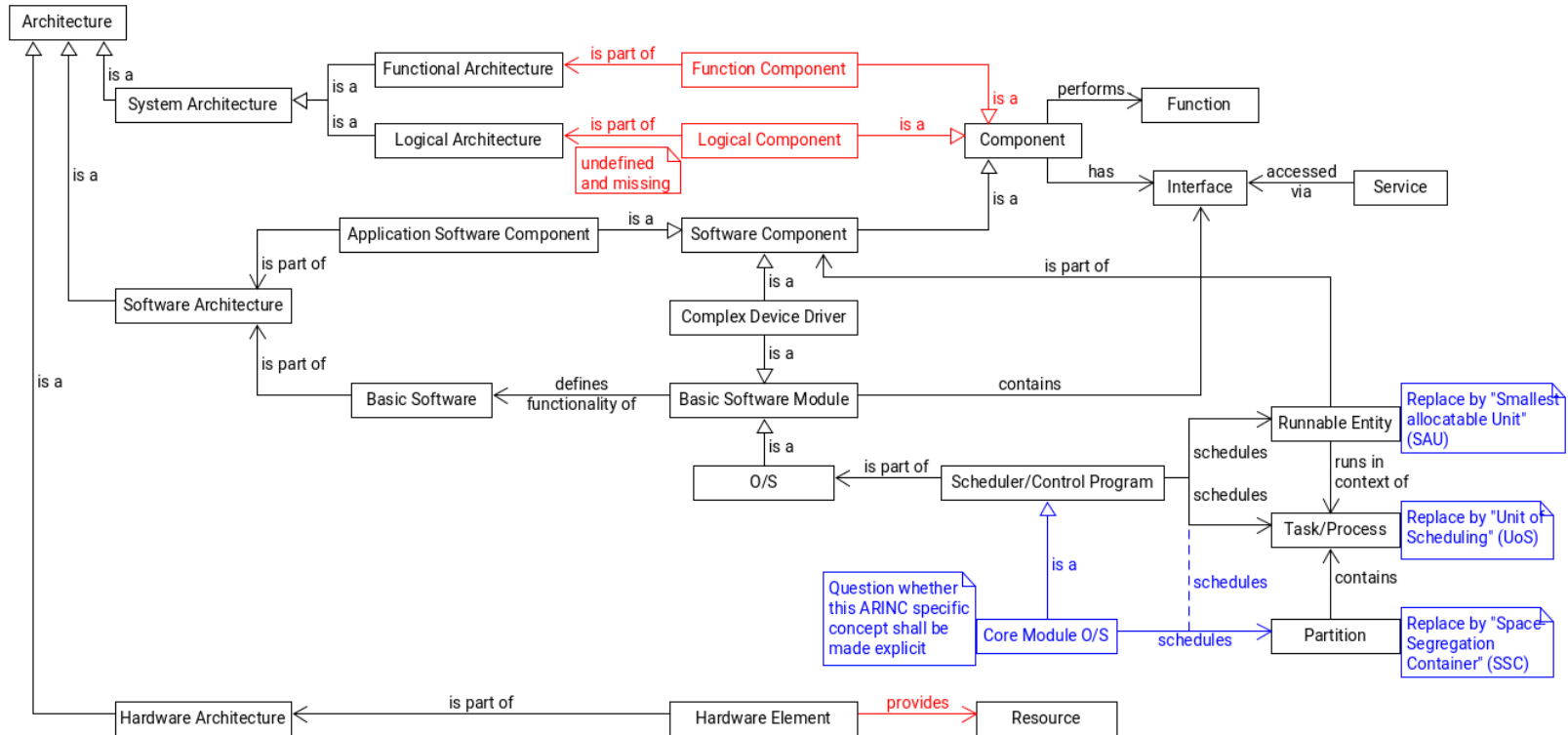


Interferenz-Modelle

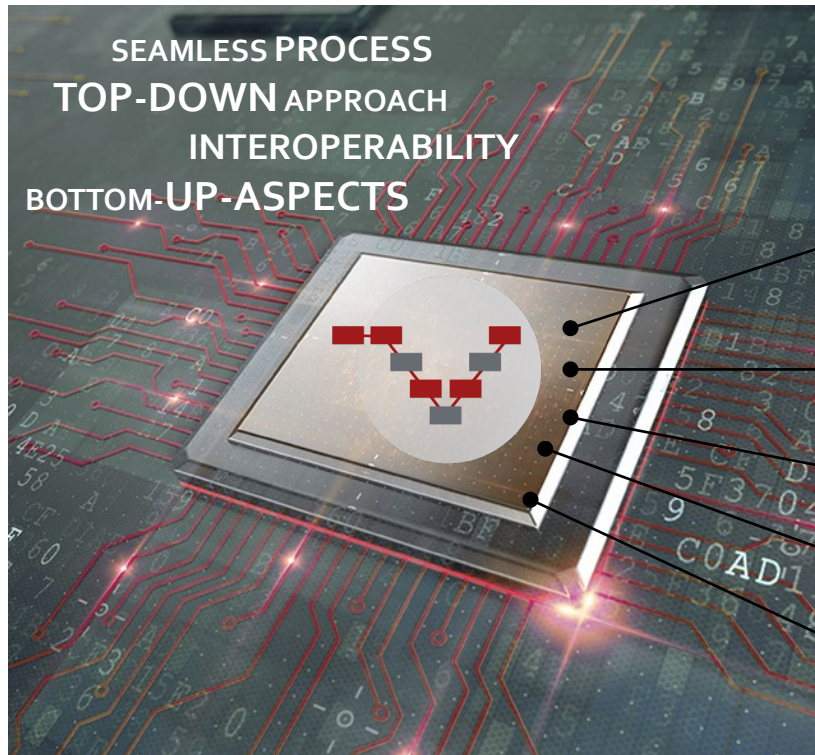
- Architektur-Ebene
- Ursachen
- Safety-Auswirkungen
- Interferenzklasse
- Effekt
- Eindämmung



Domänenübergreifende Terminologie



- Software- und Hardware-Modelle sowie Methoden zum Entwurf, zur Implementierung, Bewertung, Analyse, Verifizierung und Validierung von Lösungen
- Modelle, die die Zusammenhänge zwischen Software- und Hardware-Elementen beschreiben: Abhängigkeiten, Allokationen, Auslastung, usw.
- Interferenzmodelle: Ausführung, Safety, usw.
- Tools und Interoperabilität sind **entscheidend** für eine systematische und automatisierte Systementwicklung



Strukturierte Multicore-Entwicklung



Festlegung eines generischen und nahtlosen Entwicklungsprozesses für Multicore-Systeme



Modellbasierter "Top-Down" Entwicklungsprozess, der unnötige Entwicklungsschleifen verhindert



... aber auch bottom-up Aspekte berücksichtigt

Umsetzung durch Methoden und Werkzeuge in TP3

Evaluation in den Use Cases

- Generische Multicore-Entwicklungsprozess, ausgerichtet an den Vorgaben aktueller **Industriestandards**
 - Erweiterungen für die **Multicore**-spezifischen Tätigkeiten
 - Formale Spezifikation erforderlicher **Artefakte** für Zertifizierung/Qualifizierung von Multicore-Systemen
- Definition einer einheitlichen, **domänenübergreifenden Terminologie**
- Durchgängige Verwendung eines **gemeinsamen Metamodells**
 - Entwicklungsprozess wird durch entsprechende Tools/Toolchains anwendbar
- Evaluation der Anwendung in den Use Cases