ARCADIA

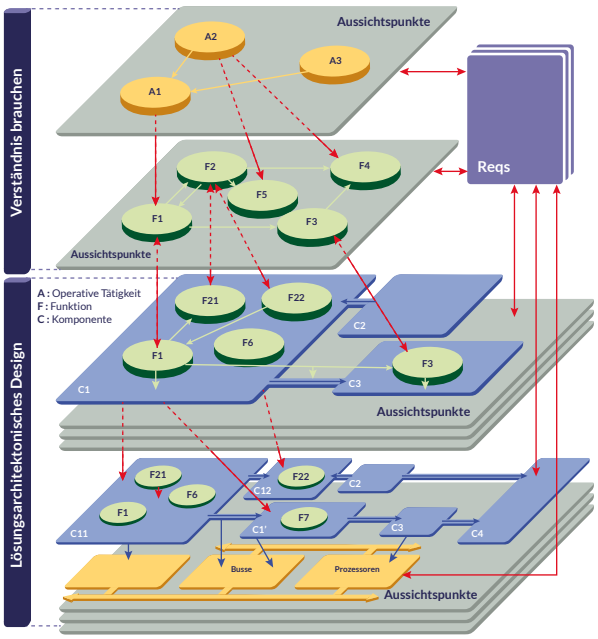
eclipse.org/capella

EINE SOFTWAREUNTERSTÜTZTE METHODE ZUR

# Definition, Analyse, Entwicklung & Validierung von System-, Software- und Hardware-Architektur

## Unterstützt effektive Zusammenarbait im Engineering


## Erlaubt die Validierung einer Lösung gegenüber den betrieblichen Anforderungen Vereinfacht Auswirkungsanalysen vereinfachen




**Betriebliche Analyse**
Was der Systemnutzer erreichen will

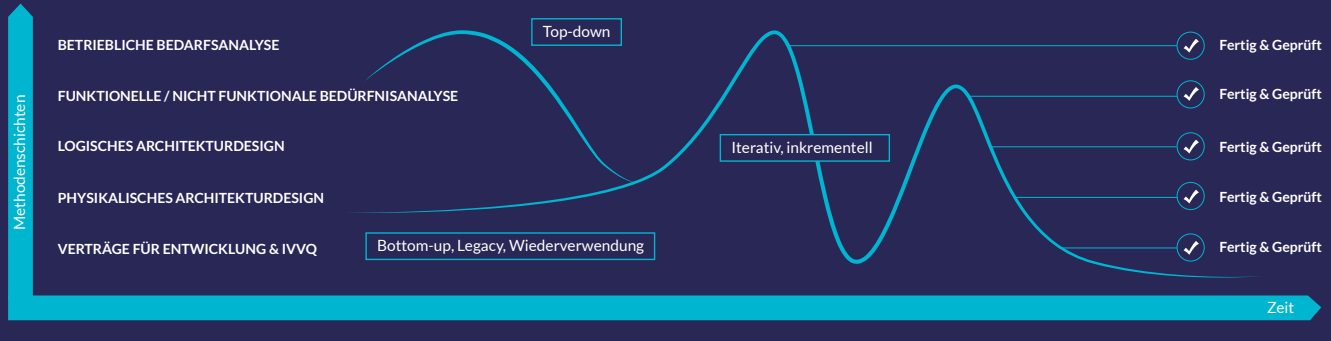
**Funktionaler & nicht-funktionaler Bedard**
Was days System für den Benutzer bewältigen soll

**Logische Architektur**
Wie das System funktion-ieren wird, um die Anforderungen zu erfüllen

**Physikalische Architektur**
Wie das System entwickelt und erstellt wird

## Kompatibel mit dem meisten Prozessen

top-down bottom-up, Iterativ, Legacy-basiert, gemischt...



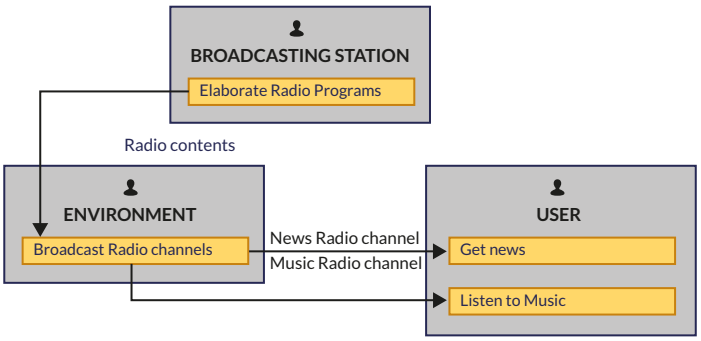
METHODISCHE SCHRITTE | AUFGABEN | BEISPIELMODELL

BEDARF

## Analyse der betrieblichen Anforderungen des Kunden

**Was die Systemnutzer erreichen wollen**

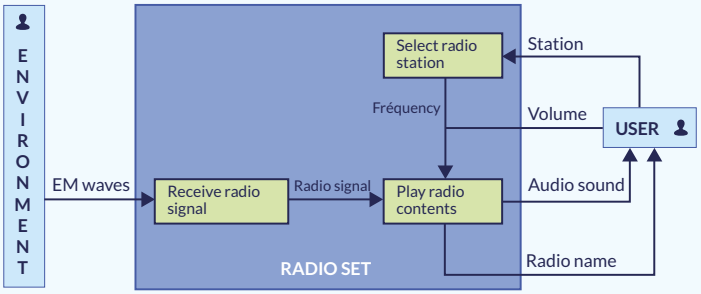
- ✓ Operational capabilities definieren
- ✓ Analyse der betrieblichen Anforderungen durchführen




## Analyse der System SW- und HW-Anforderungen

**Was das System für die Nutzer leisten soll**

- ✓ Capability trade-off analysieren
- ✓ Funktionale und nichtfunktionale Analyse durchführen
- ✓ Anforderungen formalisieren und konsolidieren

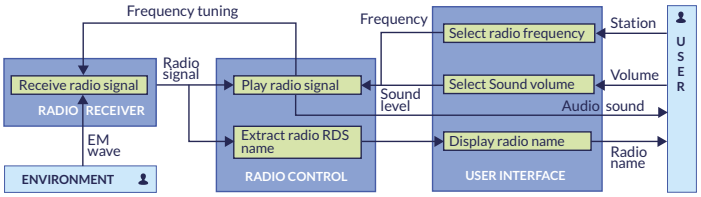



LÖSUNGEN

## Design der logischen Architek

**Wie das System funkionerieren wird, um Anforderungen zu erfüllen**

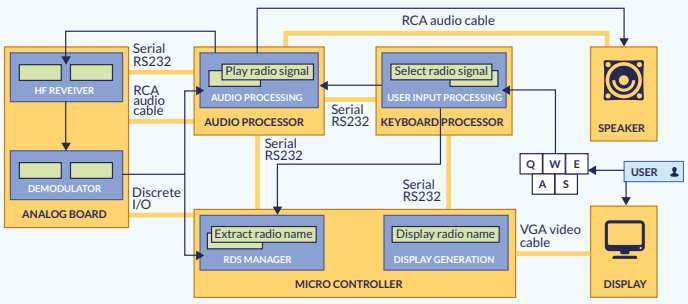
- ✓ Architekturrichtlinien und Sichten auf die Architektur definieren
- ✓ Mögliche Aufteilungen der Komponenten erstellen
- ✓ Den besten Kompromiss für die Architektur auswählen




## Design der physikalischen Architektur

**Wie das System entwickelt und erstellt wird**

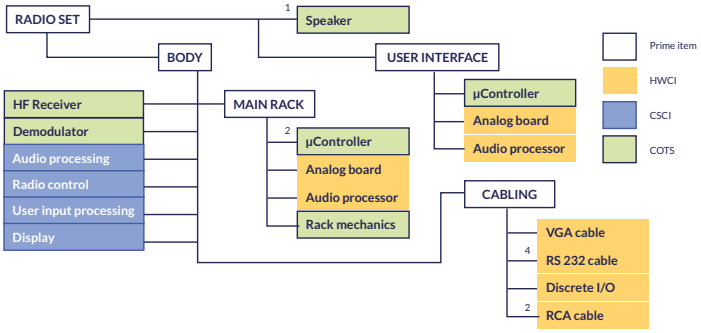
- ✓ Architekturmuster definieren
- ✓ Wiederverwendung bestehender Assets berücksichtigen
- ✓ Eine physikalische Referenzarchitektur entwickeln
- ✓ Validieren und prüfen




## Entwicklungs-veinbarungen

**Was von jedem Entwickler/Zulieferer erwartet wird**

- ✓ Définition d'une stratégie IVVQ des composants
- ✓ Définition et mise en œuvre d'une structure de répartition du produit et de contrats pour l'intégration des composants



## Konzepte

- Operational Capabilities
- Aktoren, operational entities
- Activities
- Interaktionen zwischen Activities & Aktoren
- In Activities und Interaktionen verwendete Informationen
- Betriebliche Prozesse als Abfolge von activities
- Szenarien für dynamisches Verhalten

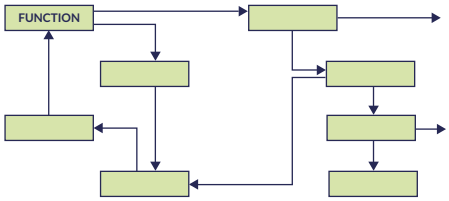
- Aktoren und System, Capabilities
- Funktionen des Systems & der Aktoren
- Datenaustausch zwischen Funktionen
- Datenflüsse durch Functional Chains
- In Funktionen & im Datenaustausch verwendete Informationen, Datenmodell
- Szenarien für dynamisches Verhalten
- Modes & States (Zustandsmachinen)

*GLEICHE KONZEPTE, ZUZÜGLICH:*
- Komponenten
- Komponentenaports und Schnittstellen
- Zuweisung der Funktionen zu den Komponenten
- Definition der Komponentenschnittstellen durch Zuweisung von Functional Exchanges

*GLEICHE KONZEPTE, ZUZÜGLICH:*
- Verhaltenskomponenten verfeinern logische Komponenten und implementieren das Funktionsverhalten
- Implementierungskomponenten stellen die Resourcen für die Verhaltenskomponenten zur Verfügung
- Physikalische Links zwischen Implementierungskomponenten

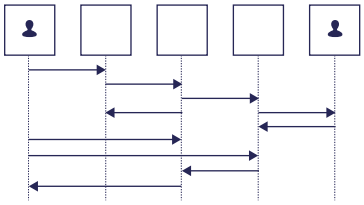
- Baum aller Konfigurationsobjekte
- Teilenummern, Anzahl
- Entwicklungsvereinbarungen (erwartetes Verhalten, Schnittstellen, Szenarien, Ressourcenverbrauch, nichtfunktionale Eigenschaften...)

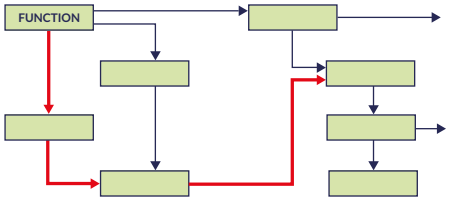
## MITTEL ZUR BESCHREIBUNG



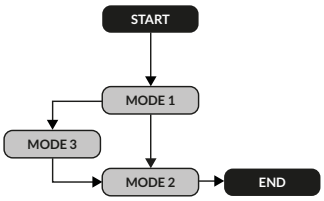

**Datenfluss:** Funktionen,Operational Activities, Interaktionen und Datenaustausch

**Szenarien:** Actors, System, Komponenteninteraktionen und Datenaustausch

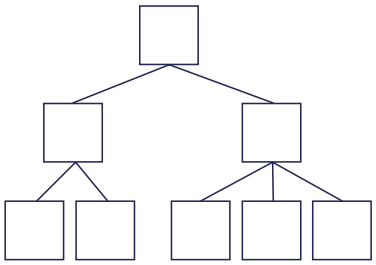





**Functional chains, Betriebliche Prozesse** durch Funktionen und Operational Activities

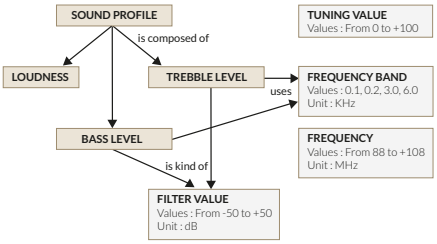


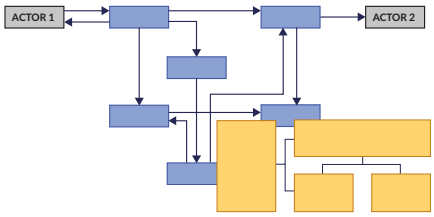

**Modes & States** von Aktoren, System und Komponenten



**Aufteilung** von Funktionen & Komponenten

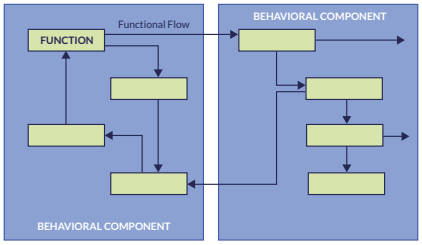
**Datenmodell**: Datenfluss & Szenarien, Definition und Erläuterung der Schnittstellen



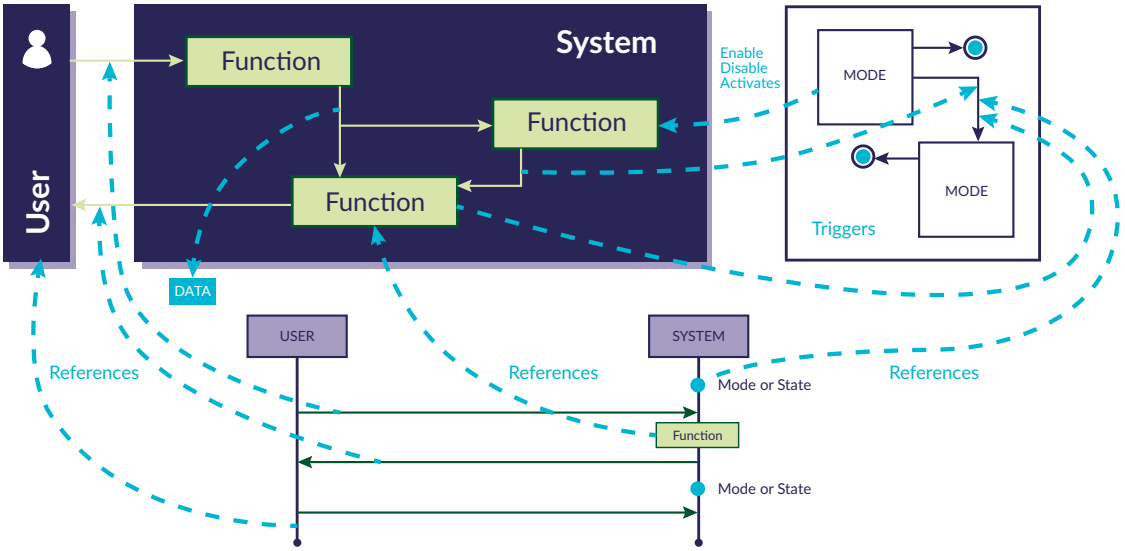




**Komponentenverbindungen:** Alle Arten von Komponenten

**Zuweisung** der Operational Activities zu Aktoren, der Funktionen zu den Komponenten, der Verhaltenskomponenten zu den Implementierungskomponenten, der Datenflüsse zu den Schnittstellen, der Elemente zu den Konfigurationsobjekten



DYNAMISCHES VERHALTEN

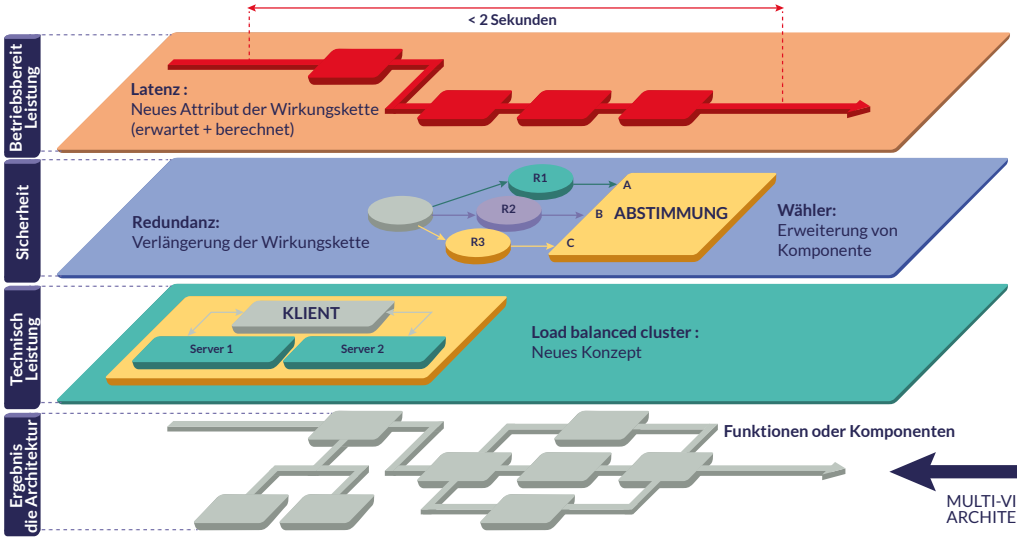



NICHTFUNKTIONALE SICHTEN AUF DIE ARCHITEKTUR (BEISPIEL)

# Lösungen gegenüber nichtfunktionalen & industriellen Anforderungen prüfen und verifizieren

| Methodische Schritte | Beispiele für Performance Anforderungen | Beispiele für Sicherheitsanforderungen |
|---|---|---|
| BETRIEBLICHE ANFORDERUNGSANALYSE | Max. Reaktionszeit auf Gefährdung | Ungewollte Ereignisse |
| FUNKTIONELLE/NICHTFUNKTIONELLE BEDARFSANALYSE | Functional Chain (FC) zur Reaktion auf Gefährdung. Maximale erlaubte Latenz auf der FC | Mit Ereignissen verbundene kritische Functional Chains |
| DESIGN DER LOGISCHEN ARCHITEKTUR | Komplexität der Verarbeitung und des Datenaustauschs Zuweisung der Functional Chains | Sichern der Functional Chains durch reduntante Wege |
| DESIGN DER PHYSIKALSCHEN ARCHITEKTUR | Resourcenverbrauch auf der FC Folgende Berechnungslatenz | Häufige Fehlermöglichkeiten Fehlerfortpflanzung auf der FC |
| ENTWICKLUNGSVEREINBARUNGEN UND IVVQ | Zugewiesene Resourcen zur akzeptablen Latenz | Benötigter Zuverlässigkeitslevel |

- ✓ Kosten und Planung
- ✓ Schnittstellen
- ✓ Leistung
- ✓ Wartbarkeit
- ✓ Betriebs-/Informationssicherheit
- ✓ ...
- ✓ IVVQ
- ✓ Produktpolitik




**ARCHITEKTUR-CHECK**

- Berechnen Sie die Funktionszykluszeit entsprechend der implementierten Komponente& Kommunikationsperf
- Leiten Sie die erreichte Latenz ab und vergleichen Sie diese mit der erwarteten
- Vergleichen Sie das Redundanzniveau mit der Kritikalität der Funktionskette
- Weiterleitung von HW-Fehlern an funktionale Ketten
- Berechnen Sie die Last basierend auf Funktionen und prüfen Sie die Angemessenheit
- Sowohl bei der Rechenleistung als auch bei der Kommunikationsbandbreite

MULTI-VIEWPOINT ARCHITEKTURPRÜFUNG

THALES