

# Agenda – Requirements Engineering

## Requirements Elicitation

- Erhebung (nicht)-funktionaler Anforderungen
- Erfassung aller Stakeholder
- Erfassung der Workflows der wichtigsten Stakeholder (Administrator, Engineer, User, usw.)
- Vorbereitung für das Meeting:
  - - Artefakte zur Erhebung funktionaler Anforderungen (Use-Case, User Stories)
  - - Artefakte zur Erhebung nicht-funktionaler Anforderungen (NFA-Template)
  - - Vorläufiges System einführen in welchem Informationen gemanaged werden können

## Requirements Analysis

- Konflikt-Analyse zu Stakeholdern (Kano-Modell)
- Klassifizierung der Anforderungen (IEEE Std. 830-1998 + evtl. mehr Dimensionen)
- Konsistenzprüfung (Widersprüche auflösen)
- Abstraktionshierarchien definieren (Ziele des Systems, Stakeholder, User, usw.)

## Requirements Specification/Documentation

- Nicht-funktionale Anforderungen spezifizieren (ISO 25010)
- Funktionale Anforderungen spezifizieren (IEEE 29148, Use Cases/Stories, usw.)
- Story Maps und Epic-Strukturen etablieren
- Lastenheft (Kundenansicht) und Pflichtenheft (Anbieteransicht)

## Requirements Verification/Validation

- Prototyping zur Validierung mit Stakeholdern
- Reviews mit Stakeholdern
- Checklisten zu Anforderungen (Eindeutig, messbar, testbar, nachvollziehbar, usw.)

## **Requirements Management**

- Traceability der Informationen (vertikal, horizontal)
- Versionsverwaltung
- Change Control Policy einrichten (Wer hat welche Rechte bei Anforderungsänderung, Freigabe, Bewertung usw.)

## **Fragen**

- Weitere Stakeholder nennen
- Aktuellen Workflow der Stakeholder (Administrator, Engineer, usw.)
- Beispiel oder eine Referenzinstanz eines AAS ? Oder bekommen wir eine konkrete Instanz gestellt, mit der wir arbeiten können?

# Lastenheft – Vorschlag

## 1. Allgemeines

Zielgruppe: Bosch-interne Entwickler|innen und Partnerunternehmen  
Hintergrund: Synchronisation von Asset Administration Shells und Zugriffskontrolle via Eclipse Dataspace Connector.

## 2. Projektziele und Nicht-Ziele

Ziele:

- Synchronisation von AAS-Daten zwischen mehreren Systemen.
- Automatische Konfiguration von EDC-Zugriffsregeln auf Basis der AAS-Konfiguration.
- Optionale Konfigurations-UI zur einfachen Bedienung.
- Monitoring Option zur Nachverfolgung von Performance und Synchronisationsstatus

Nicht-Ziele:

- Entwicklung eines vollständigen industriellen Produktivsystems.

## 3. Ist-Zustand

Derzeit sind AAS und EDC separate Systeme ohne zentrale Verwaltung oder Synchronisationsmechanismen. Zugriffskontrolle erfolgt manuell, Synchronisation ist nicht vorhanden.

## 4. Anforderungen

Funktionale Anforderungen:

- Definition und Verwaltung von Synchronisationsregeln zwischen AAS.
- Automatische Synchronisation über REST-API (bidirektional).
- Automatisierte Erstellung und Pflege von EDC-Zugriffsrechten.
- Unterstützung verschiedener AAS-Metamodelle.

Nicht-funktionale Anforderungen:

- Performante Synchronisation (innerhalb Zeitintervall)
- Skalierbarkeit auf mehrere parallele AAS-Verbindungen.
- Sicherheit (Rollenbasierter Zugriff)
- Erweiterbarkeit

## 5. Benutzer & Rollen

- Entwickler: Konfiguration der Synchronisationsregeln.
- Administratoren: Zugriff auf Logs, Monitoring und Management der EDC-Regeln.
- (Optional) Nutzer: UI zur grafischen Regel-Erstellung und -Bearbeitung.

## **6. Abnahmekriterien**

- Synchronisation zwischen mindestens zwei AAS funktioniert zuverlässig.
- Zugriff auf Daten wird korrekt nach EDC-Regeln gesteuert.
- System reagiert performant.
- Optional: UI und Monitoring laufen stabil und benutzerfreundlich.