# **Dokumentation und Kommunikation**

# **Kommunikation von Softwarearchitekturen Warum so wichtig?**

• Verständnis und Zusammenarbeit

• Einheitliche Vision

• Risikominimierung

• Bessere Entscheidungsfindung

• Kunden- und Stakeholder-Engagement

• Dokumentation und Nachvollziehbarkeit

**Understanding and Collaboration**:

* Clear communication of the software architecture enables members of the development team to understand the structure and design of the system and collaborate effectively.

 **Unified Vision**:

* By communicating the architecture, a unified vision is created for the entire development team. Everyone shares a common understanding of how the system is built and how it should function.

 **Risk Mitigation**:

* Early communication of the architecture helps identify and address potential risks and issues before they escalate into larger problems.

 **Better Decision Making**:

* Clear communication of the architecture allows decision-makers to make informed decisions about design and development guidelines based on the project's goals and requirements.

 **Customer and Stakeholder Engagement**:

* Communicating the architecture allows customers and stakeholders to be involved in the development process, helping them better understand their requirements and ensuring that the end product meets their expectations.

 **Documentation and Traceability**:

* Clear communication of the architecture facilitates appropriate documentation of the system and its decisions, making it easier to trace and understand for future developers and maintenance activities.

# **Arten der Kommunikation von Softwarearchitekturen**

• Dokumentation

• Architekturdiagramme

• Präsentationen

• Prototypen und Proof-of-Concepts

• Code und Kommentare

• Workshops und Brainstorming-Sitzungen

# **ISO/IEC/IEEE 42010-2022**

• “Systems and Software Engineering – Architecture Description“

• Norm, die sich mit der **Architekturbeschreibung von Systemen befasst**

• Legt Grundsätze und Best Practices fest, um effektive **Architekturbeschreibungen zu erstellen, zu kommunizieren und zu verwalten**

**• Definiert Schlüsselkonzepte** wie **Architekturansichten**, **Architekturrahmenwerke,** **Architekturbeschreibungssprachen** und **Architekturbeschreibungsmethoden.**

• Bietet **Richtlinien für die Auswahl und Anpassung von Architekturansätzen und -methoden**, um den spezifischen Anforderungen und Kontexten von Projekten gerecht zu werden.

* **Key Concepts**: ISO/IEC/IEEE 42010-2022 defines important concepts such as architecture views (different perspectives on the architecture), architecture frameworks (structures and guidelines for describing architectures), architecture description languages (formal languages for describing architectures), and architecture description methods (approaches for creating architecture descriptions).

**Guidelines for Selection and Adaptation**: The standard provides guidance on how to select and adapt architecture approaches and methods to meet the specific requirements

# **Konzeptionelles Modell zur Beschreibung von Softwarearchitekturen nach ISO/IEC/IEEE 42010-2022**

O imagine care conține text, diagramă, Plan, Paralel

Descriere generată automat

# **Stakeholder**

role, position, individual or organization having a right, share, claim or other interest in an architecture entity or its architecture that reflects their needs and expectations

O imagine care conține text

Descriere generată automat

# **Concern nach ISO/IEC/IEEE 42010-2022**

**Definition:**

• Ein Concern ist ein Interesse, das sich auf das System, seine Entwicklung oder seinen Betrieb bezieht.

* Funktionale Anforderungen Nicht-funktionale Anforderungen
* Zum Beispiel Wartbarkeit oder Benutzerfreundlichkeit

**Dienen als Grundlage für die Erstellung und Strukturierung der Architektursichten**

**Lenken den Fokus der Architekturarbeit auf relevante Aspekte**

**Unterstützen Kommunikation mit Stakeholdern**

**Entscheidend für die Validierung und Bewertung der Architektur**

# **Aspects nach ISO/IEC/IEEE 42010:2022**

**Definition:**

Aspekte sind bestimmte Perspektiven oder Blickwinkel, unter denen das System analysiert und beschrieben wird

• Helfen dabei, verschiedene Concerns der Stakeholder systematisch zu adressieren

• Beispiele: Sicherheits-Aspekt, Leistungs-Aspekt, WartbarkeitsAspekt, Benutzerfreundlichkeits-Aspekt

# **Viewpoint nach ISO/IEC/IEEE 42010-2022**

**Definition:**

Ein Viewpoint ist ein definierter Satz von Konventionen und Regeln zur Darstellung und Beschreibung bestimmter Concerns eines Systems.

Verschiedene Viewpoints sind dafür konzipiert, die Interessen und Bedenken unterschiedlicher Stakeholder zu adressieren.

O imagine care conține text, Font, captură de ecran, algebră

Descriere generată automat

# **Architecture View nach ISO/IEC/IEEE 42010-2022**

**Definition:**

Eine Architecture View ist eine Darstellung eines Systems aus der Perspektive eines bestimmten Architecture Viewpoints.

Konkrete Repräsentation eines Aspekts, die dazu dient, bestimmte Concerns der Stakeholder zu adressieren

**Beispiele von Architcture Views:**

UML-Diagramme: Klassendiagramme, Komponentendiagramme, Sequenzdiagramme, …

O imagine care conține text, captură de ecran, proiectare

Descriere generată automat

Annahme: Es ist nicht möglich, die Merkmale und Qualitätseigenschaften eines komplexen Systems in einem einzigen verständlichen Modell zu erfassen, das von allen Beteiligten verstanden wird und für sie von Nutzen ist.

O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat

# **Sichtenmodelle(View models)**

**Definition:**

Strukturierte Darstellungen eines Systems, die verschiedene Aspekte der Architektur aus unterschiedlichen Perspektiven oder Sichten zeigen.

* Umfassen eine Sammlung von verschiedenen Architektur-Sichten, die zusammen eine umfassende und kohärente Beschreibung eines Systems liefern.
* Die Norm ISO/IEC/IEEE 42010 bietet eine Grundlage zur Erstellung solcher Sichtenmodelle

# **4+1 Sichtenmodell nach Philippe Kruchten**

O imagine care conține text, diagramă, captură de ecran, Font

Descriere generată automat

# **Logical View:**

Fokus auf die Strukturierung des Systems in Module und deren Interaktion

# **Process View:**

Fokus darauf**, wie die Module zur Laufzeit interagieren** **und wie Daten durch das System fließen**

# **Development View:**

**Struktur des Quellcodes** sowie Details zum BuildProzess und anderer Aspekte des Entwicklungsprozesses

# **Physical View:**

Physikalische Verteilung des Systems auf Hardware sowie Aspekte wie Netzwerktopologie

# **Scenarios:**

Zeigt, die verschiedenen Sichten zusammenarbeiten anhand verschiedener Szenarien

# **Siemens‘ Four View Model**

• **Conceptual View / Konzeptionelle Sicht**

* Hauptkomponenten und deren Interaktionen
* Wichtige Architekturmuster, die angewendet werden

Views: Zum Beispiel einfache Komponentendiagramme

• **Module View / Modul- oder logische Sicht**

* Detaillierte Beschreibung der Module und ihrer Schnittstellen
* Views: Zum Beispiel Klassen-, Paket-, oder Komponentendiagramme

• **Execution View / Laufzeitsicht**

* Kommunikationswege, Verhalten in spezifischen Szenarien
* Views: Zum Beispiel Sequenzdiagramme, Aktivitätsdiagramme oder Zustandsdiagramme

• **Physical View / Physikalische Sicht**

* Hardwarekomponenten und deren Konfiguration
* Zuordnung der Softwarekomponenten zu Hardwarekomponenten
* Views: Zum Beispiel Deployment-Diagramme

# **Sichtenmodell nach arc42**

O imagine care conține diagramă, Dreptunghi, Plan, captură de ecran

Descriere generată automat

# • Kontextabgrenzung

Einbettung des Systems in seine Umgebung System als Blackbox

# • Bausteinsicht

Zerlegung in Subsysteme, Komponenten, Frameworks, ...

# • Laufzeitsicht

* Interaktion von Laufzeitinstanzen
* Generelle Abläufe innerhalb der Software

# • Verteilungssicht

* Technische Umgebung und Infrastruktur
* Hardwarekomponenten und deren Interaktion / Netzwerktopologie
* Deployment-Artefakte

O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat

# **Ebenen in arc42**

Ebenen werden in arc42 verwendet, um Architekturdokumentation in unterschiedliche Granularitätsstufen zu unterteilen

* + So kann man spezifische Teile der Dokumentation besser reprasentieren

# **Makroarchitektur**

* Gesamtstruktur des Systems
* Hauptkomponenten und ihre Funktionen
* Grobe Interaktionen und Schnittstellen

# **Mittlere Ebene**

* Aufteilung des Systems in Subsysteme/Komponenten
* Interaktionen zwischen Subsystemen/Komponenten
* Schnittstellen und Kommunikationswege

# **Mikroarchitektur**

* Detaillierte Beschreibung der Bausteine oder Komponenten
* Interne Struktur und Abhängigkeiten
* Schnittstellen und Kommunikationswege

O imagine care conține text, diagramă, Plan, schiță

Descriere generată automat

# **UML – Unified Modeling Language**

Jede Sicht braucht geeignete Beschreibungstechniken

Mit einer Diagrammart lassen sich nicht alle Aspekte aller Sichten gleichermaßen abbilden.

Für jeden Viewpoint werden unterschiedliche Views benötigt.

**DEFINITION:**

Mit der „Unified Modeling Language (UML)“ steht eine grafische Modellierungssprache zur Verfügung, welche zur Dokumentation von Softwaresystem verwendet werden kann.

• Sprache zur Erstellung von Views

**Diagramme in UML lassen sich in zwei Hauptgruppen einteilen:**

**• Strukturdiagramme**

Klassendiagramm, Komponentendiagramm, Verteilungsdiagramm,

Kompositionsstrukturdiagramm, Paketdiagramm, Objektdiagramm, Profildiagramm

**• Verhaltensdiagramme**

Aktivitätsdiagramm, Anwendungsfalldiagramm, Interaktionsübersichtsdiagramm,

Kommunikationsdiagramm, Sequenzdiagramm, Zeitverlaufsdiagramm, ZustandsdiagrammO imagine care conține text, captură de ecran, Font, proiectare

Descriere generată automat

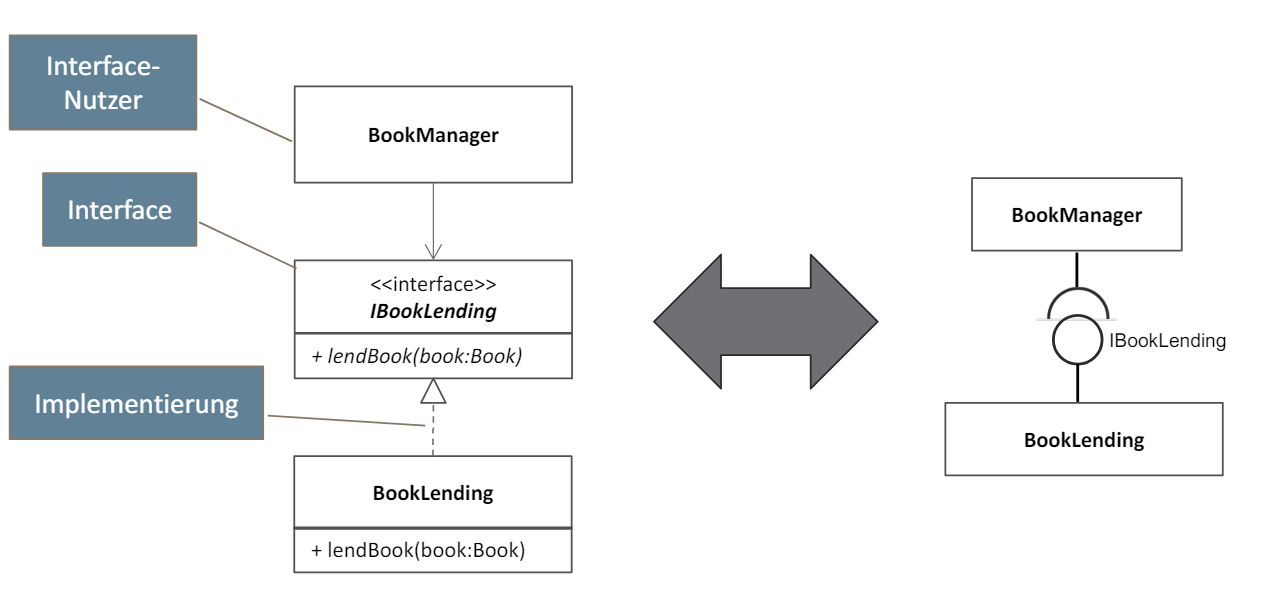
# **UML Klassendiagramm**

• Grafische Darstellung von Klassen, Schnittstellen, Attributen und Methoden sowie deren Beziehungen zueinander.

• Mit Klassendiagrammen lassen sich unter anderem auch Datenbank-Strukturen und Domänenmodelle darstellen.

• Wird häufig verwendet in

Bausteinsicht / Module View und Conceptual View

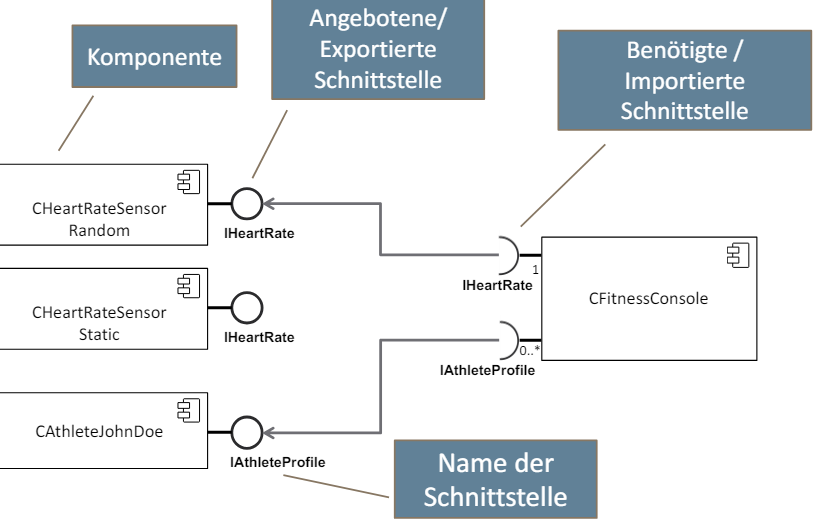


# **UML Komponentendiagramm**

• Grafische Modellierung von Komponenten und Subkomponenten sowie deren Verdrahtung.

• Wird häufig verwendet in

* Bausteinsicht
* Module View
* Kontextabgrenzung



O imagine care conține text, diagramă, Font, linie

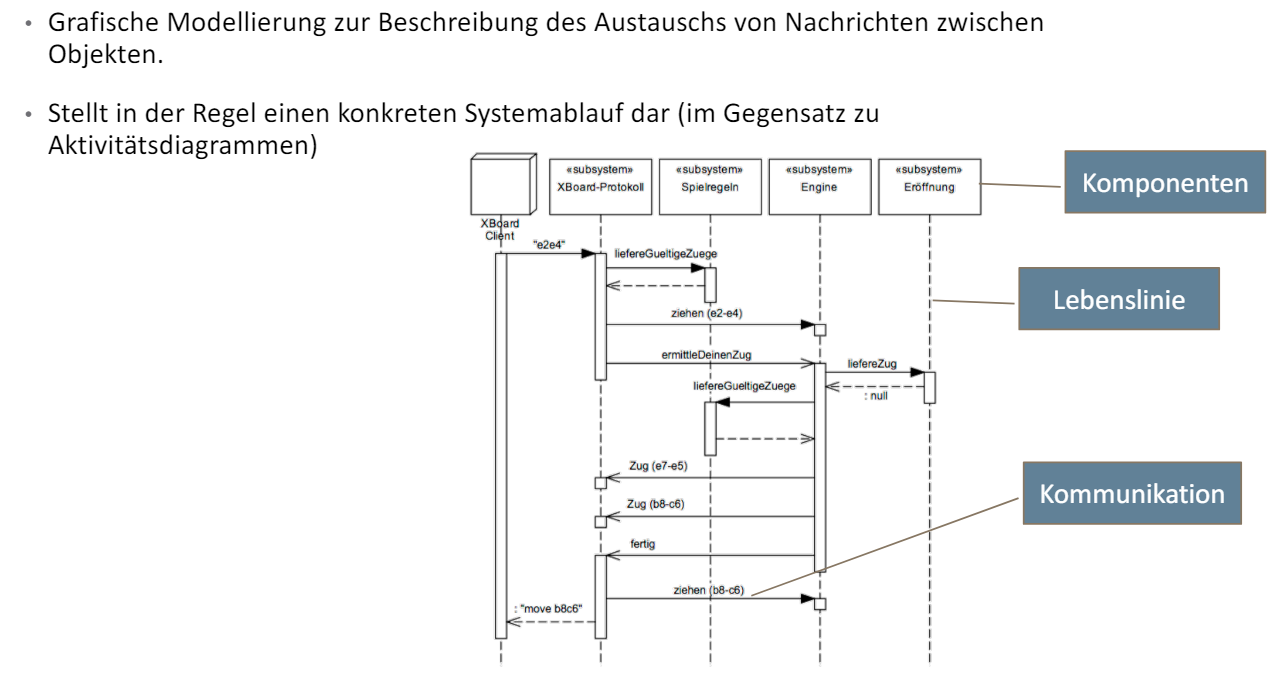
Descriere generată automat

# **UML Verteilungsdiagramm**

O imagine care conține text, captură de ecran, diagramă, Font

Descriere generată automat• Grafische Modellierung der Verteilung von Komponenten auf Rechnerknoten

# **UML Sequenzdiagramm**



# O imagine care conține text, captură de ecran, diagramă, Font Descriere generată automat**UML Aktivitätsdiagramm**

# **Konsistenz zwischen Diagrammen**

O imagine care conține text, diagramă, Plan, Desen tehnic

Descriere generată automat

O imagine care conține text, diagramă, Plan, Desen tehnic

Descriere generată automat

# **Dokumentation von Architekturentscheidungen**

# **Architekturentscheidungen**

• Mit UML werden insbesondere Ist- und Sollzustand einer Architektur dokumentiert

**• Hauptarbeit eines Architekten ist es, Architektur-relevante Entscheidungen zu treffen.**

• Diese Entscheidungen können

* Auf dem Ist-Zustand beruhen
* Auf einer Auswahl von möglichen Plänen und Soll-Zuständen beruhen
* Auf zahlreichen weiteren Faktoren beruhen

Diese Entscheidungen selbst müssen dokumentiert werden

# **Architekturentscheidungen zu dokumentieren, hilft dabei**

* Entscheidungen Team-übergreifend sichtbar zu machen
* Entscheidungen im Team und im Unternehmen durchzusetzen
* Die Qualität von Architekturentscheidungen zu messen
  + Welche Entscheidungen hatten welche Auswirkungen?
* Die Entstehung einer künftigen Ist-Architektur nachzuvollziehen
* dem Team Guidelines für detailliertere Entscheidungen zu geben

• Feedback-Loops helfen, indem die getroffenen Entscheidungen regelmäßig hinterfragt und bewertet werden.

# **Architectural Decision Records**

**Definition:**

Bewährte Möglichkeit, Architekturentscheidungen zu dokumentieren sind Architectural Decision Records (ADRs).

**O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat**

# **Allgemeine Aspekte zur Dokumentation**

**Weitere übliche Dokumenttypen**

O imagine care conține text, captură de ecran, Font

Descriere generată automat

# **Praxisregeln zur Dokumentation**

**• Schreiben aus Sicht des Lesers**

**• Unnötige Wiederholungen vermeiden**

**• Mehrdeutigkeit vermeiden**

**• Standardisierte Organisationsstruktur bzw. Templates und SatzSchablonen**

**• Begründen Sie wesentliche Entscheidungen schriftlich**

**• Überprüfung der Gebrauchstauglichkeit**

**• Übersichtliche Diagramme**

**• Regelmäßige Aktualisierungen**

**Eine Architekturbeschreibung sollte so einfach wie möglich sein, aber so umfangreich wie notwendig.**