

Softwarearchitektur und Design

Block 3c – Vorgehen

SS2014

DI Dr. Gottfried Bauer

LV-Typ: VO, UE

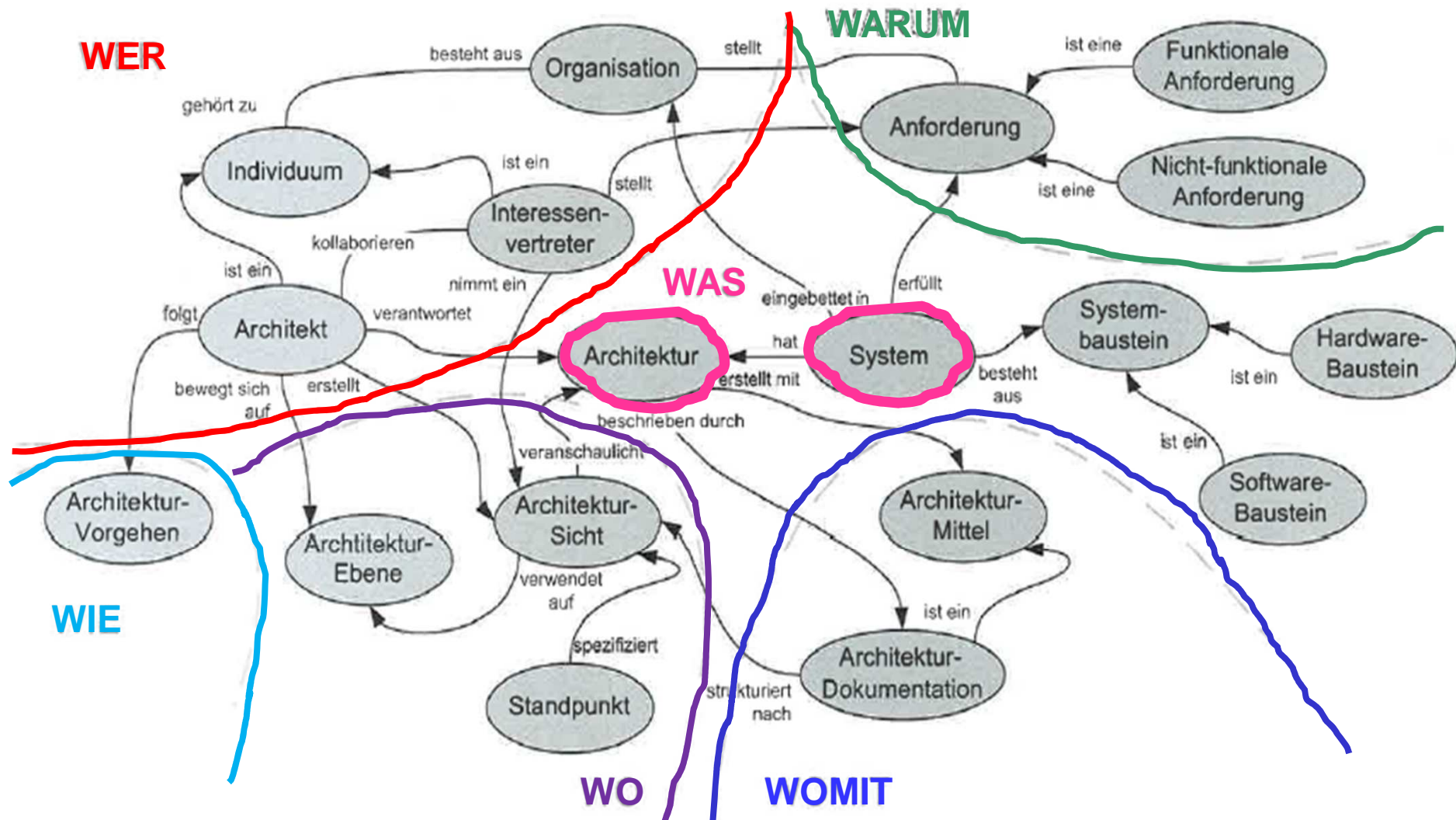
Semester: 2

LV-Nummer: **S 2012 ILV**

LV-Bezeichnung: Softwarearchitektur und Design

Mindmap zur SW-Architektur

SAD
Mindmap zu Arch.



Inhalt - Block 3c

SAD
Inhalt - Block 3c

■ 6 – **WIE**

Architektur-Vorgehen

Vision-Entwurf-Umsetzung-Kommunikation;
Entwurfsentscheidungen

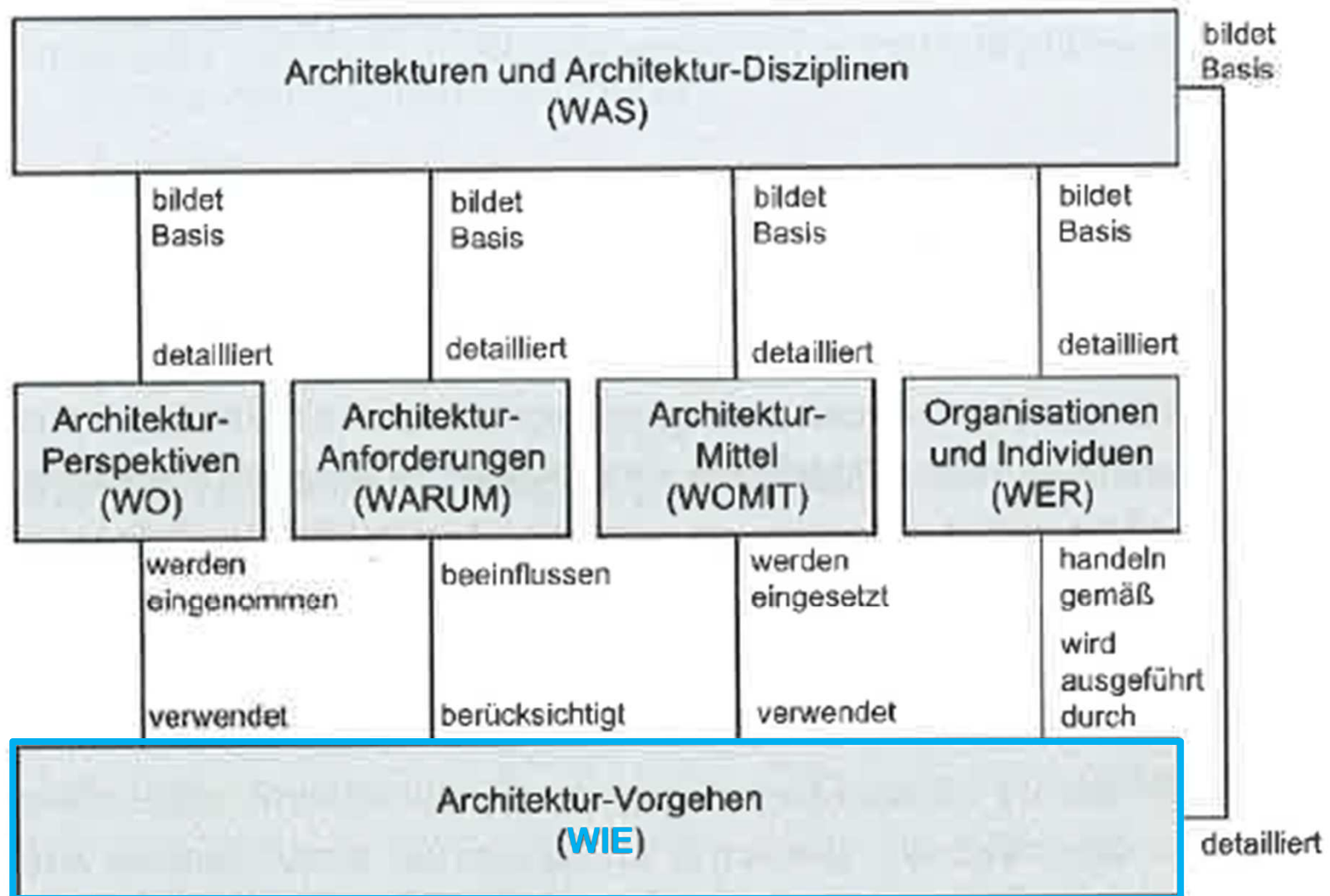
Architektur-Organisation - **WIE**

**SAD
Vorgehen**

**Architektur -
Vorgehen
WIE**

Architektur-Vorgehen - WIE-1

SAD
Vorgehen

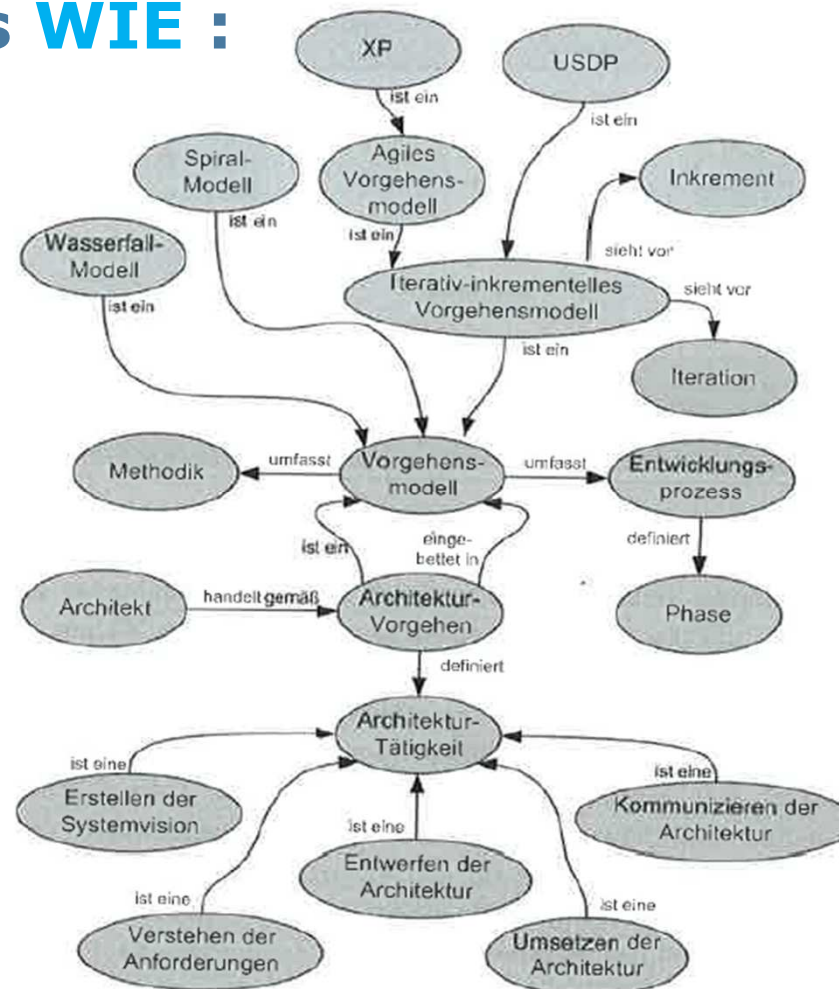


O.Vogel et.al., Software-Architektur, Grundlagen-Konzepte-Praxis, Spektrum, 2009

Architektur-Vorgehen - WIE-2

SAD
Vorgehen

Konzepte des WIE :



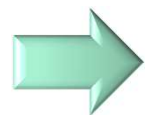
Architektur-Vorgehen - **WIE-3**

**SAD
Vorgehen**

Was meint Vorgehen und **WIE**:

- **Der Architekt muss die Architekturmittel und seine Erfahrung zielgerichtet und bewusst einsetzen**
- **Software-Architektur:**
 - umfasst neben der Software-Strukturen auch den Weg dorthin
 - ...
 - Projekt, Vorgehensmodell spielen eine Rolle, Hauptaufgabe und Herausforderung ist es, eine Architektur zu definieren, die die funktionalen und nichtfunktionalen Anforderungen abdeckt
 - ...

Welches sind wichtige architektonische Tätigkeiten und Schritte zum Ziel ...

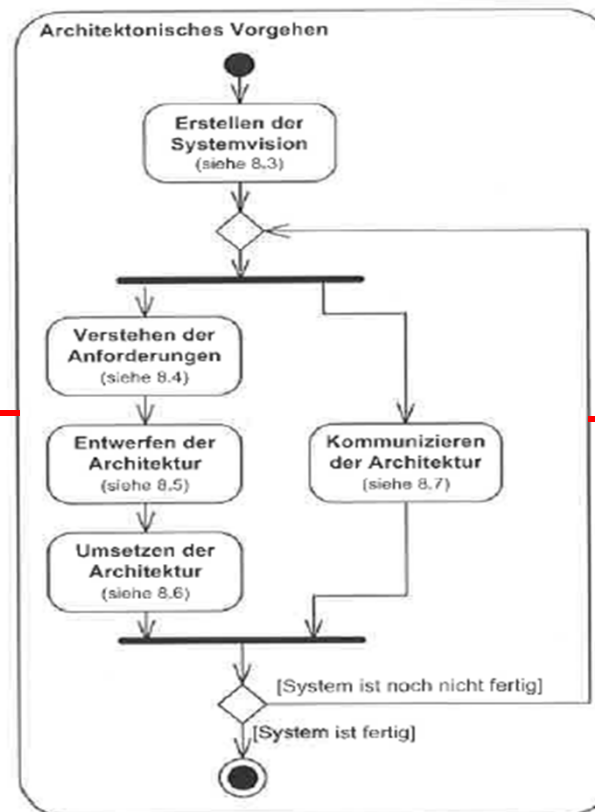


Architektur-Vorgehen - WIE-4

SAD
Vorgehen

Tätigkeiten und Schritte im Vorgehen:

„Architekt ist über den
ganzen Zyklus hinweg
involviert“



Iterativ-inkrementelles
Vorgehen

➡ **Tätigkeiten und Erarbeitung von Sichten ...** ➡

Architektur-Vorgehen - **WIE-5**

Erstellen der Systemversion: (Anforderungssicht, Logische Sicht)

- Architekt sollte mitwirken und ist (architektonischer) Berater

Verstehen der Anforderungen: (Anforderungssicht)

- Identifikation, Priorisierung und Detaillierung von **architekturelevanten** Anforderungen; Verfeinerung von Anforderungen; eventuell Prototypen
- Auseinandersetzung mit nichtfunktionalen Anforderungen wichtig

Bis hier: Abstecken des Gestaltungsspielraums für den Architekten



Ab hier: Überlegungen – Alternativen – Entscheidung für Architektur

Entwerfen der Architektur:

(Logische Sicht, Impl.-Sicht, Verteilungssicht, Prozesssicht)

- In dieser Phase entsteht die eigentliche Architektur: es sind Alternativen für mögliche Architekturen zu betrachten und zu beurteilen und eine Entscheidung für eine umzusetzende Architektur zu treffen
- Anwendung vielfältiger Architektur-Mittel (siehe **WOMIT**)

Architektur-Vorgehen - WIE-6

SAD
Vorgehen

Umsetzen der Architektur:

(Umsetzungssicht)

Umsetzung bedeutet technische Manifestation des Entwurfs:
vom Entwurf zur Entwicklung

- Umsetzung kann umfassen:
 - lediglich Definition von Entwicklungsrichtlinien
 - manuelle Reviews
 - Etablierung einer Infrastruktur, die die architekturkonforme Umsetzung gewährleistet

Kommunizieren der Architektur:

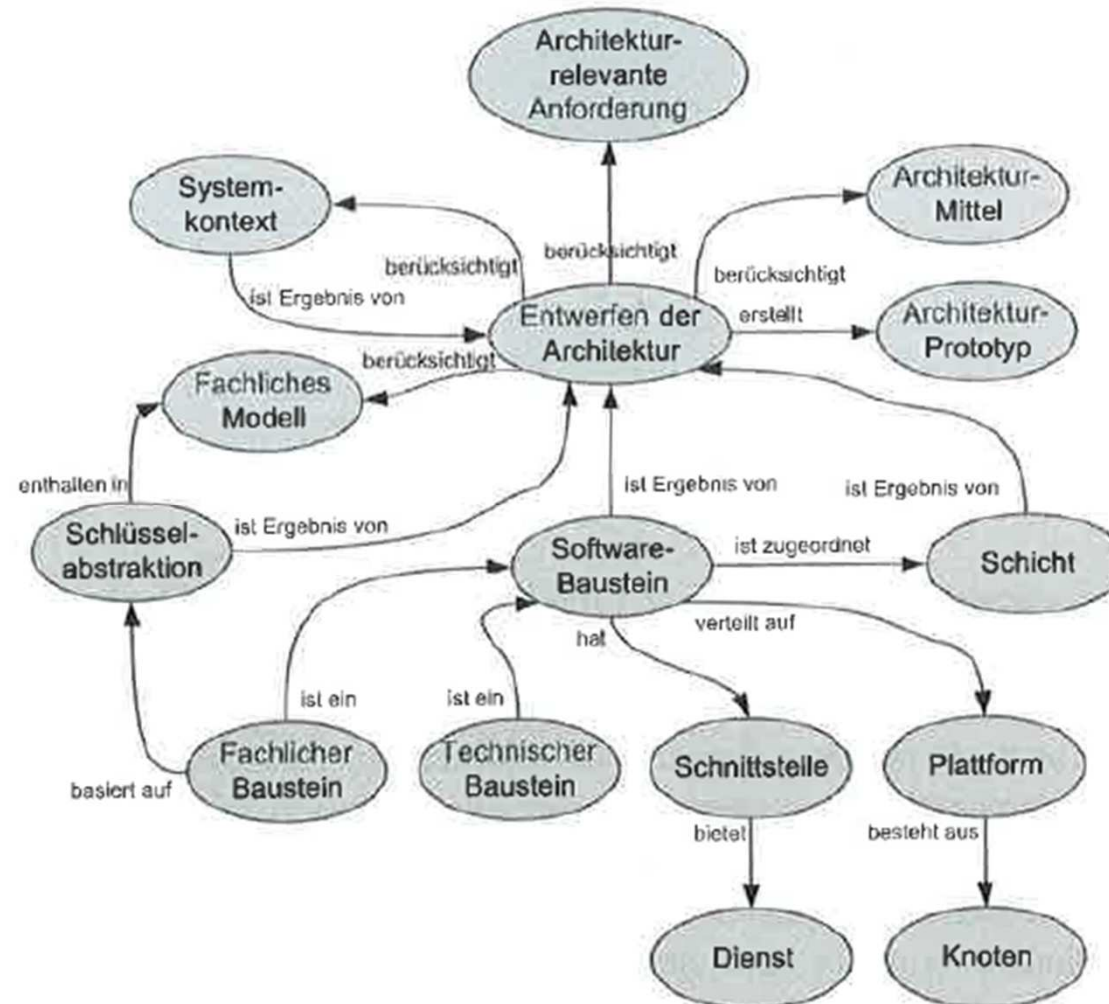
(alle Sichten)

- Architekt muss laufend die Architektur Interessensvertretern (Entwickler, Projektleiter, Benutzer, Kunden, ...) kommunizieren
- gutes Verständnis der Architektur vermitteln
- Unklarheiten erkennen und beheben, Review-Schritte durchführen, ...
- Dokumentierte Architekturnichtlinien vorteilhaft (z.B. Namenskonventionen)
- über: gute Architekturdokumentation (inkl. architektonische Entscheidungen)

Architektur-Vorgehen - WIE-7

SAD
Vorgehen

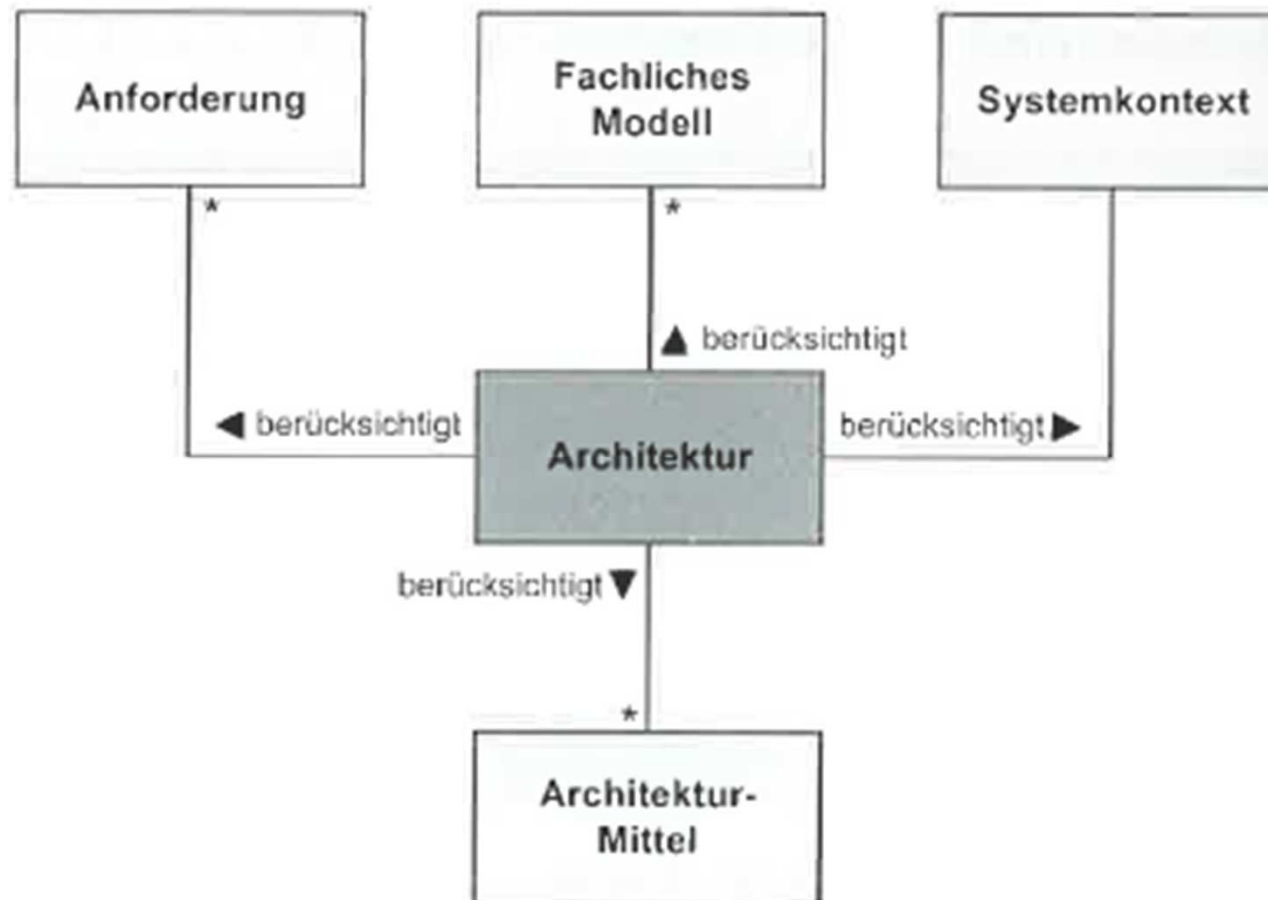
Konzepte des Entwerfens:



Architektur-Vorgehen - WIE-8

SAD
Vorgehen

Entwerfen der Architektur - Einflussfaktoren:



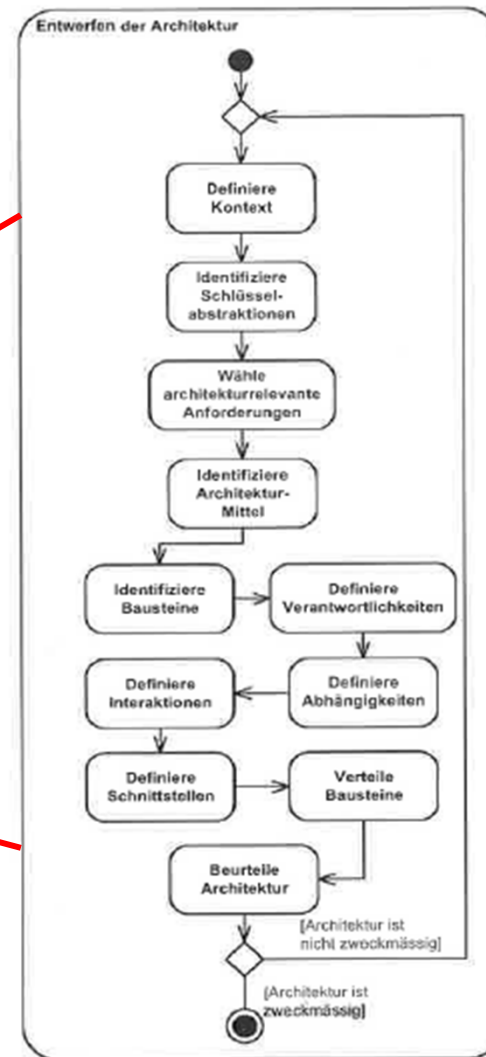
Architektur-Vorgehen - WIE-9

SAD Vorgehen

Entwerfen der Architektur - Aktivitätsdiagramm:

**Erster „Durchlauf“
bzw. erstes Inkrement =
Architektur Vision
(Wichtiges steht
rahmenartig schon fest)**

Weitere „Durchläufe“:
Erarbeitung von –
Architektur Sichten,
Verfeinerungen, ...



Architektur-Vorgehen - WIE-10

SAD
Vorgehen

Entwerfen der Architektur – Details:

■ Kontext:

- Schnittstellen (inkl. Technologien) und Systembausteine
- Dienste und Akteure

■ Schlüsselabstraktionen:

- Abstraktionen – fachliche Ebene -> fachliche Architektur

■ architekturelevante Anforderungen:

- Anforderungen mit hoher Priorität früh einfließen lassen

■ Architekturmittel:

- rein der Zweckmässigkeit folgen (nicht Hypes folgen)
- auf korrekte und sinnvolle Verwendung von Mitteln achten
- Mittel für fachliche Architektur einsetzen
- Mittel für technische Architektur einsetzen (Fundus ist grösser)
- Fachlich -> Übergang zu techn. Arch. (Datenbanken, Application Server, Middleware, ... auswählen bzw. entscheiden)
- Auswahlhilfen: Referenzarchitekturen suchen (geben aber z.B. auch Technologien vor)-> Basisarchitekturen -> Muster-Taktiken ...

Architektur-Vorgehen - WIE-11

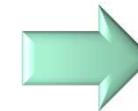
SAD
Vorgehen

Entwerfen der Architektur – Details:

■ Bausteine (bzw. Subsysteme) und Verantwortlichkeiten:

- Identifikation der Bausteine erfolgt unter Verwendung der ausgewählten Architektur-Mittel
- Bausteine und ihre Identifikation sind eng mit ihrer Verantwortlichkeit verknüpft
- es gibt fachliche Bausteine: befassen sich mit den Schlüsselabstraktionen und den damit verbundenen funktionalen Anforderungen)
- es gibt technische Bausteine: fachliche Bausteine alleine sind nicht lebensfähig – sie müssen sich auf Plattformen stützen – auch dazu empfiehlt sich zuerst eine Suche nach Referenzarchitekturen:

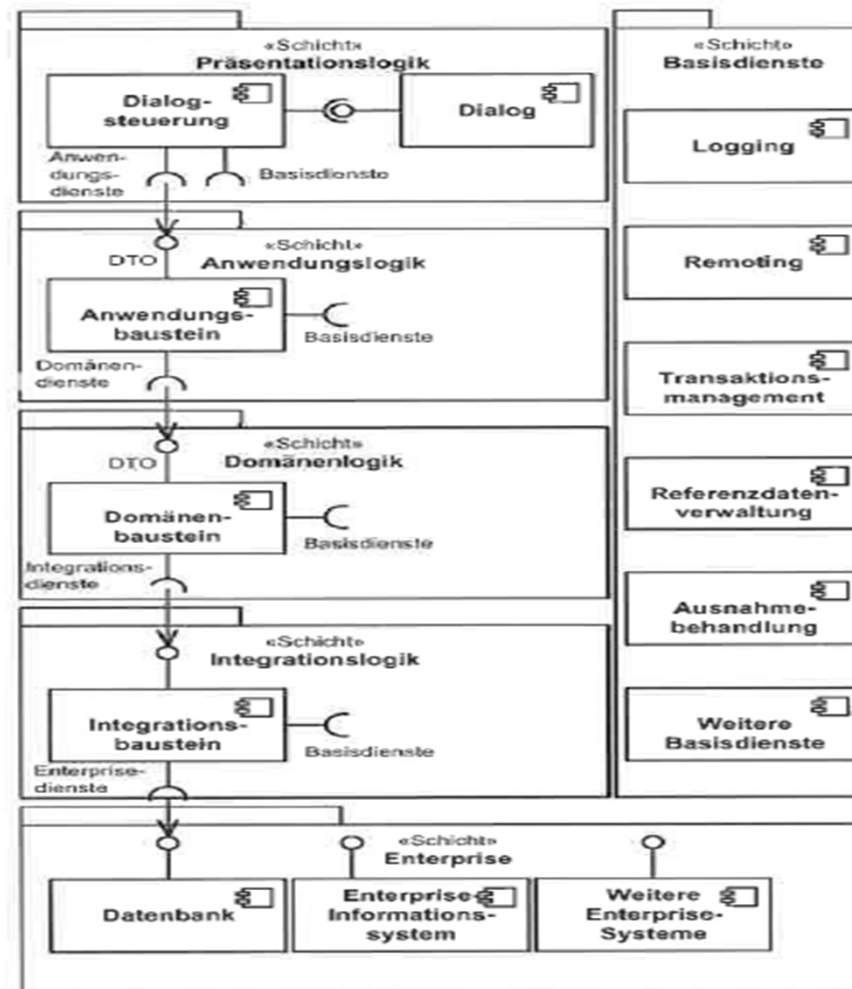
Beispiel und Legende siehe folgende Folien



Architektur-Vorgehen - WIE-12

SAD
Vorgehen

Beispiel - technische Referenzarchitektur:



Architektur-Vorgehen - WIE-13

SAD
Vorgehen

Beispiel - technische Referenzarch. - Legende:

Schicht	Verantwortlichkeit
Basisdienste	Technische Bausteine, die Basisfunktionalität verkörpern, sind dieser Schicht zugeordnet. Sie können von Bausteinen der Präsentationslogik-, Anwendungslogik-, Domänenlogik- und Integrationslogik-Schicht verwendet werden. Sollten die Schichten über Prozess und Netzwerkgrenzen hinweg verteilt sein, müssen entsprechende Aufrufmechanismen (Stichwort: Entfernte Aufrufe (englisch: <i>remoting</i>)) vorgesehen werden.
Präsentationslogik	Die Präsentationslogik-Schicht enthält Bausteine, die zur Kommunikation mit dem Benutzer dienen.
Anwendungslogik	Die Anwendungslogik-Schicht enthält Bausteine, die Anwendungslogik realisieren. Anwendungslogik ist Logik, die eng mit der zu realisierenden Anwendung in Beziehung steht und nicht respektive nur schwer über Anwendungsgrenzen hinweg wiederverwendet werden kann. Die Darstellung eines Leistungsindikators in grafischer oder textueller Form ist beispielsweise vom konkreten Anwendungsfall abhängig und schwer wiederverwendbar.
Domänenlogik	Die Domänenlogik-Schicht enthält Bausteine, die Domänenlogik realisieren. Domänenlogik repräsentiert Fachlichkeit, die über Anwendungsgrenzen hinweg wiederverwendet werden kann. Sie ist somit anwendungsneutral und operiert allein auf fachlichen Abstraktionen. Die Berechnung eines Leistungsindikators ist beispielsweise unabhängig davon, ob der Indikator grafisch oder textuell dargestellt wird.
Integrationslogik	Die Integrationslogik-Schicht beherbergt Bausteine, welche die Anbindung von Enterprise-Systemen kapseln. Ein einfaches Beispiel sind Data-Access-Objekte, die den Datenbankzugriff abstrahieren [Fowler 2003].
Enterprise	Die Enterprise-Schicht, oftmals auch Backend-Schicht genannt, beinhaltet Systeme, mit denen das zu entwickelnde System interagieren muss. Dies können Datenbanken, Enterprise-Informationssysteme, System-Management-Systeme etc. sein.

Schichten und Verantwortlichkeiten

Bausteine und Verantwortlichkeiten

Baustein	Verantwortlichkeit
Dialog	Ein Dialog entspricht dem View-Baustein innerhalb des Model-View-Controller-Architektur-Musters [Buschmann et al. 1996].
Dialogsteuerung	Ein Dialogsteuerungsbaukasten entspricht dem Controller-Baustein innerhalb des Model-View-Controller-Architektur-Musters [Buschmann et al. 1996]. Er nutzt Dienste von Anwendungsbausteinen.
Anwendungsbaustein	Ein Anwendungsbaustein kapselt Anwendungslogik und stellt diese zur Verfügung. Der Datenaustausch erfolgt mittels Datentransferobjekten (DTO) [Fowler 2003].
Domänenbaustein	Ein Domänenbaustein kapselt Domänenlogik und stellt diese zur Verfügung. Der Datenaustausch erfolgt mittels Datentransferobjekten (DTO).
Integrationsbaustein	Ein Integrationsbaustein kapselt Integrationslogik (z. B. Datenbank, SAP, LDAP) und stellt diese zur Verfügung.
Basisdienstbaustein	Ein Basisdienstbaustein stellt Basisdienste zur Verfügung (z. B. Logging, Referenzdatenverwaltung).

O. Vogel et.al., Software-Architektur, Grundlagen-Konzepte-Praxis, Spektrum, 2009

Architektur-Vorgehen - WIE-14

SAD
Vorgehen

Entwerfen der Architektur – Details:

■ Schnittstellen:

- Bausteine kommunizieren über Schnittstellen
- Stellen selbst Funktionalität (Dienste) zur Verfügung -> Schnittstellen
- Nutzen Funktionalität (Dienste) anderer Bausteine -> Schnittstellen
- Schnittstellen bestimmen den Grad der Abhängigkeit von Bausteinen
- Schnittstellen sollten ihren Zweck aber nicht ihre Implementierung offenbaren
- Dienste: Zweck sollte bereits aus Namensgebung ersichtlich sein
- Grundsätzliche Arten von Diensten:
 - ändern Zustand eines Bausteins nicht = Funktionen, Abfragen
 - ändern Zustand des Bausteins = Kommandos*möglichst nicht vermischen in einem Baustein ...*
- Dienste haben Vor- und Nachbedingungen
- Dienste haben Eingabe- und Ausgabeparameter
- Dienste sollten möglichst kein Details offenbaren (z.B. Datenaustausch über Datentransferobjekte und nicht über Entitäten)
- Beschreibungen von Schnittstellen: UML, IDL, WSDL, DSL, ...

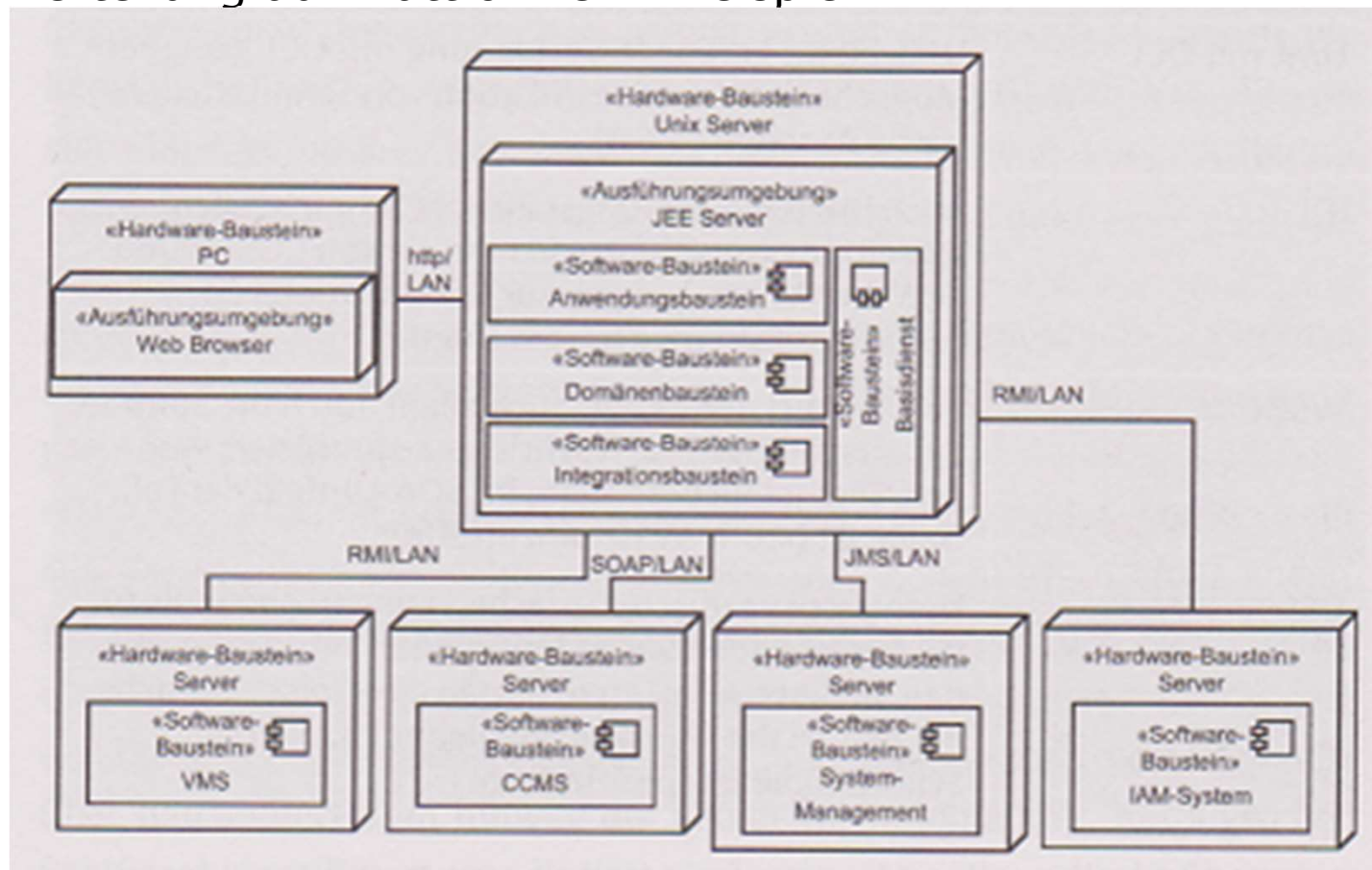
Architektur-Vorgehen - WIE-15

SAD
Vorgehen

Entwerfen der Architektur – Details:

■ Verteilung der Bausteine:

- Verteilung auf Plattformen - Beispiel:



O.Vogel et.al., Software-Architektur, Grundlagen-Konzepte-Praxis, Spektrum, 2009

Architektur-Vorgehen - WIE-16

SAD
Vorgehen

Entwerfen der Architektur – Checkliste:

(weitere Checklisten siehe auch im folgenden Foliensatz zu den Übungen:
MIT_SAD_Übungen_SS2014_UE_Informationen-3)

Werden alle Umsysteme (auch mittelbare wie z. B. System-Management) im Systemkontext aufgeführt?

Werden alle architekturrelevanten Anwendungsfälle durch Bausteine abgedeckt?

Ist jeder Baustein in mindestens einen Anwendungsfall involviert?

Wurde jeder Baustein hinsichtlich seiner Verantwortlichkeit, Schnittstelle und Interaktionen mit anderen Bausteinen dokumentiert?

Werden alle nicht-funktionale Anforderungen jeweils durch entsprechende Mittel abgedeckt?

Wurde jede Schlüsselabstraktion einem fachlichen Baustein zugewiesen?

Wurden fachliche Bausteine gemäß der eingesetzten Referenzarchitektur entworfen?

Wurden Bausteine für Querschnittsaufgaben (z. B. Logging) vorgesehen?

Besitzt jeder Baustein eine klare Verantwortlichkeit und keine Überlappungen mit anderen Bausteinen?

Wurde jeder Baustein einer passenden logischen Schicht zugeordnet?

Existieren Schnittstellen zwischen dem System und seinen Umsystemen?

Existieren Schnittstellen zwischen den Subsystemen?

Erfolgen Interaktionen zwischen Bausteinen ausschließlich über Schnittstellen?

Finden keine unnötigen Interaktionen zwischen Bausteinen statt?

Existieren keine zirkulären Abhängigkeiten zwischen Bausteinen?

Wurde jeder Baustein auf einen Knoten verteilt?

Wird der Belang Nebenläufigkeit ausreichend berücksichtigt?

Wurde die Architektur einer Beurteilung unterzogen?

Wurden die Ergebnisse der Beurteilung in die Architektur eingearbeitet?

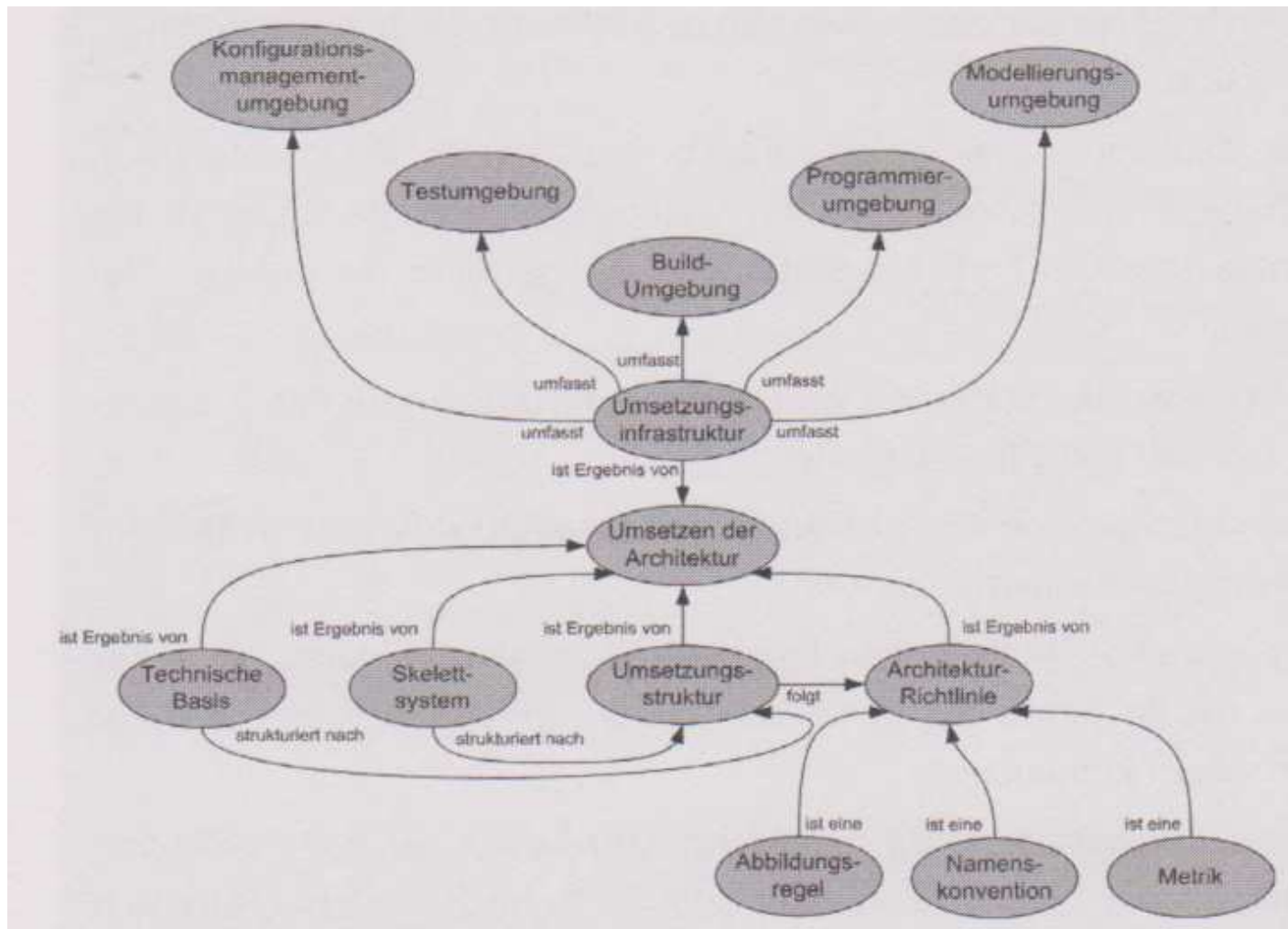
Wurde der Reifegrad der eingesetzten Technologien ausreichend beachtet und bewertet?

Wurden organisatorische Rahmenbedingungen berücksichtigt?

Architektur-Vorgehen - WIE-17

SAD
Vorgehen

Umsetzen der Architektur – Konzepte:



O.Vogel et.al., Software-Architektur, Grundlagen-Konzepte-Praxis, Spektrum, 2009

Architektur-Vorgehen - WIE-18

SAD
Vorgehen

Umsetzen der Architektur – Checkliste:

- Wird die Einhaltung von Architektur-Richtlinien automatisiert geprüft?
- Wird die korrekte Umsetzung der Architektur automatisiert sichergestellt?
- Existiert in der Dokumentation zu jeder Metrik je ein Beispiel für den Gut- und Schlecht-Fall?
- Existieren Regeln für die Abbildung von Bausteinen der logischen Sicht auf Bausteine der Umsetzungssicht?
- Findet sich jeder Baustein der logischen Sicht in der Umsetzungssicht wieder?
- Wurde die Umsetzungsinfrastruktur auf die organisatorischen Rahmenbedingungen abgestimmt?
- Wurde die Umsetzungsinfrastruktur gemäß Architektur-Richtlinien konfiguriert?
- Gibt es eine Anleitung (englisch: *cookbook*) für die Umsetzungsinfrastruktur?
- Existiert eine technische Basis?
- Genügt die technische Basis der vorgegebenen Architektur?
- Deckt die technische Basis alle architekturelevanten Anwendungsfälle ab?

Archit.-Entscheidungen-**WIE**-19

SAD
Entscheidungen

Architektur und Entscheidungen:

- Entscheiden ist eine Schlüsselaufgabe des Architekten
- Was sind architektonisch notwendige und relevante Entscheidungen
- Wesentliche Merkmale von Entscheidungen:
 - Es gibt mehrere Möglichkeiten, Alternativen

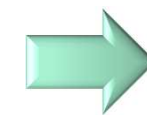


Problemlösungs- und Entscheidungsprozess ()*

- Entscheidung fundiert getroffen

(*) Architektur muss alle funktionalen und nichtfunktionalen Anforderungen ausreichend berücksichtigen -> ist daher eigentlich immer ein Kompromiss (Entscheidung = „Entscheidung für den besten Kompromiss“).

Entscheidungsfindungsprozess hat Schritte ...



Archit.-Entscheidungen-WIE-20

SAD
Entscheidungen

Schritte im Entscheidungsprozess:

(kann intuitiv oder rational erfolgen ...)

■ Entscheidung vorbereiten:

- Anforderungen, Restriktionen, IT-Umfeld, ...

■ Entscheidungen treffen:

- Varianten gegenüberstellen, Balance bezüglich der Erfüllung der Anforderungen unter gegebenen Restriktionen finden, desweiteren Kostenüberlegungen, organisatorische Überlegungen

■ Entscheidung kommunizieren:

- immer (pro)aktiv kommunizieren

■ Entscheidung realisieren:

- Ausgewählte Variante nach Entscheidung (erfolgreich) umsetzen

■ Entscheidung beurteilen:

- Beurteilung im Nachhinein (+/-) -> Lernen (für zukünftige Entscheidungen)

Wichtig:

Richtige Informationsmengen bereitstellen für die Entscheidungsfindung;
architektonische Entscheidungen sind in der Regel strategischer Natur

Archit.-Entscheidungen-**WIE**-21

SAD
Entscheidungen

Gegenstände/Themenpunkte/Aspekte von Architektur-Entscheidungen:

- Architekturelevante Anforderungen (z.B. Skalierbarkeit)
- Auswahl bestimmter, geeigneter Architektur-Mittel (z.B. Muster)
- Art und Weise der Anwendung der Architektur-Mittel (z.B. es fehlt ein Baustein, der zur Anwendung eines Musters erforderlich ist)
- Struktur der Bausteine (z.B. bestimmte Bausteine arbeiten direkt und nicht über eine Schnittstelle zusammen)

Bei Vorliegen notwendiger Architekturentscheidungen können unter anderem folgende Fragen beantwortet werden:

- Welches sind die architekturelevanten Anforderungen und warum ?
- Warum wurden bestimmte Architekturmittel ausgewählt und warum ?
- Warum wurden die Architektur-Mittel auf diese Weise angewendet ?
- Warum liegen die Bausteine in dieser Struktur vor ?

O.Vogel et.al., Software-Architektur, Grundlagen-Konzepte-Praxis, Spektrum, 2009

Archit.-Entscheidungen-WIE-22

SAD Entscheidungen

Beispiele architektonischer Entscheidungen bzw. zugehöriger Fragestellungen:

- Aus welchen Subsystemen besteht das System + welche Verantwortlichkeiten haben die jeweiligen Subsysteme
- Wie ist das System auf unterschiedliche Schichten organisiert + welche Subsysteme auf welchen Schichten angesiedelt
- Welche Beziehungen bestehen zwischen den Subsystemen und wie kommunizieren diese
- Welches sind die architekturtragenden Bausteine + wie sehen die Schnittstellen zu anderen Bausteinen aus
- Auf welcher Plattform wird das System betrieben + welche Dienste werden benötigt
- Wie werden SW-Bausteine auf der Plattform installiert + wie sind diese über HW-Bausteine verteilt
- ...