

6. Entwurf / Architekturmodellierung

Ziele:

Architekturen, Architektursprachen und ihre Bedeutung

Übersicht über den Entwurfsprozeß

Details bzgl. Entwurfsprozeßwissen, Wiederverwendbarkeit

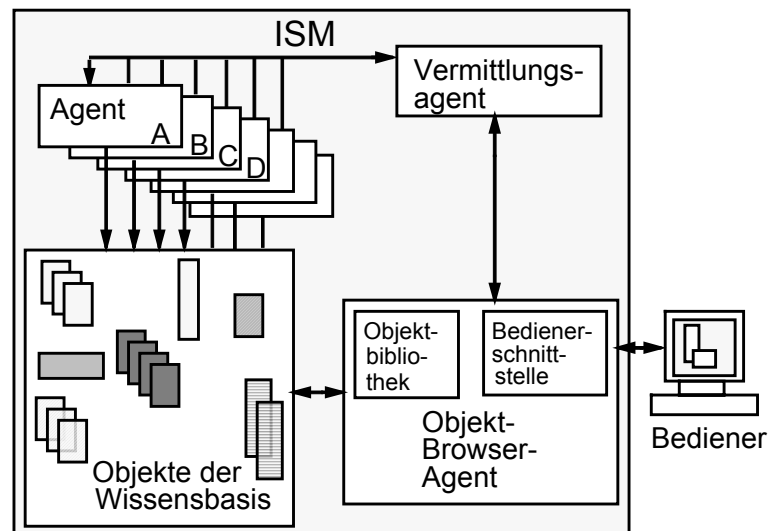
→ Architekturmodellierungsvorlesung

Architekturen: Bedeutung und Festlegung

Die Festlegung der Entwurfsspezifikation

- Wiederholung: Aufgabe d. Entwurfs = Beschreibung des Innenlebens eines Softwaresystems auf einer groben Ebene
statisch: Modulgeflecht, Softwarearchitektur, Bauplan
- Festlegung in einer Sprache informell, formal; Graphik, Text
Aspekte: Syntax, Semantik, Pragmatik
Graphik: Module: Art, Name als Knoten
Beziehungen: Art als Kanten
zur Übersicht
Text: Module Export und Import
zur detaillierten Festlegung
- Was verstehen wir **nicht** unter Architektur

Kästen:
Komplexe
Verbindungen:
haben etwas
miteinander zu
tun



Die ISM Architektur

Architektursprachen

- Syntax der Entwurfsspezifikation
 - Kontextfreie Syntax: Aufbau eines PiG-Dok. an einer Stelle
 - Welche Textelemente (Arten), wie aufgebaut?
 - Welche Graphikelemente (Arten)?
 - Kontextsensitive Syntax: Zusammenwirken verschiedener Teile
 - Übereinstimmung verschiedener Textteile
 - Aussondern verbotener Strukturen
- Semantik der Entwurfsspezifikation
 - Verschieden v. Semantik eines fertigen Programms:
 - keine dyn. Semantik, da Rümpfe noch offen
 - Zusammenwirken der Module als statische Angaben
 - drücken wir innerhalb der Syntax aus
 - Begriff Softwarearchitektur bevorzugt
 - Sprache und Architektur haben viel mit Semantik des Anwendungsbereichs und Semantik des beschriebenen Programmsystems zu tun
- Semantik der Module
 - kann formal definiert werden
 - alg. Spezifikationen
 - Vor-/Nachbedingungen
 - hier nicht betrachtet

- Pragmatik
 - Implementierbarkeit einer Architektur
 - Übertragbarkeit in eine PS
 - Lesbarkeit einer Architektur
 - Arbeitsteilung, Überprüfbarkeit, Kostenermittlung
 - Auftragbarkeit eines Architekturdiagramms

- Hatten bereits versch. Bedeutungen von Spezifikation (RE, PiG, math. Spez.)
 - Auf Architekturmodellierungsniveau
 - Entwurfsspezifikation (Architektur (Syntax),
Funktion (Sem.), z.B. Machbarkeit (Pragmatik))
 - Architektur allein
 - Festlegung eines Moduls der Architektur (Syntax,
Sem., Pragmatik)
 - Syntaktischer Anteil der Spezifikation eines Moduls

- - Zielsetzung d. Verant.: Einführung einer formalen Sprache f. Softwarearchitekturen (Text + graph. Form)

Bedeutung		
RE,	PiG,	PiK
z.B. SA	?	z.B. übl. PS
o.ä.	Modulkonzept von PS	

- Architekturbeschreibung hier
 - Detailbeschreibung aller Module (Export, Import)
 - enthält alle Informationen
 - Überblicksdarstellung: Architekturdiagramm
 - (zusätzl. Design Rationale (DOK))
 - (Skizze der bet. Rumpfe (PiK))

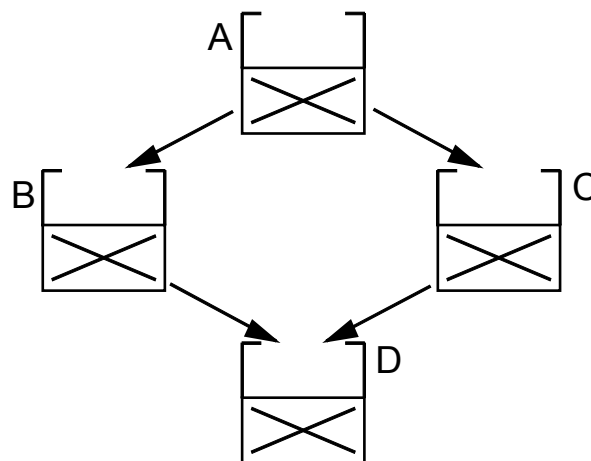
- Bedeutung einer solchen Notation
 - Quintessenz eines Programmsystems:
 - Dokument zur Beurteilung der Struktureigensch.
 - Qualitätseigenschaften damit bewertbar
 - Wiederverwendbarkeit

- Titel der Veranstaltung PiG
 - gibt keine Methode (automatisch) zur Erstellung
 - von Softwarearchitekturen: geistig schwierige Aufg.
 - neben geeign. Sprache: viele Hinweise, Teilarch.,
 - Standardstrukturen

- In PiG-Vorlesung Beschreibung von Architekturen und Teilarch.:
 - Momentaufnahmen (Ergebnisse) des Entwurfs/
 - der Wartung
 - Weniger mit Prozeß selbst
 - d. h. Entstehen einer Architektur,
 - RE ->PiG - Übergang,
 - PiG -> PiK - Übergang

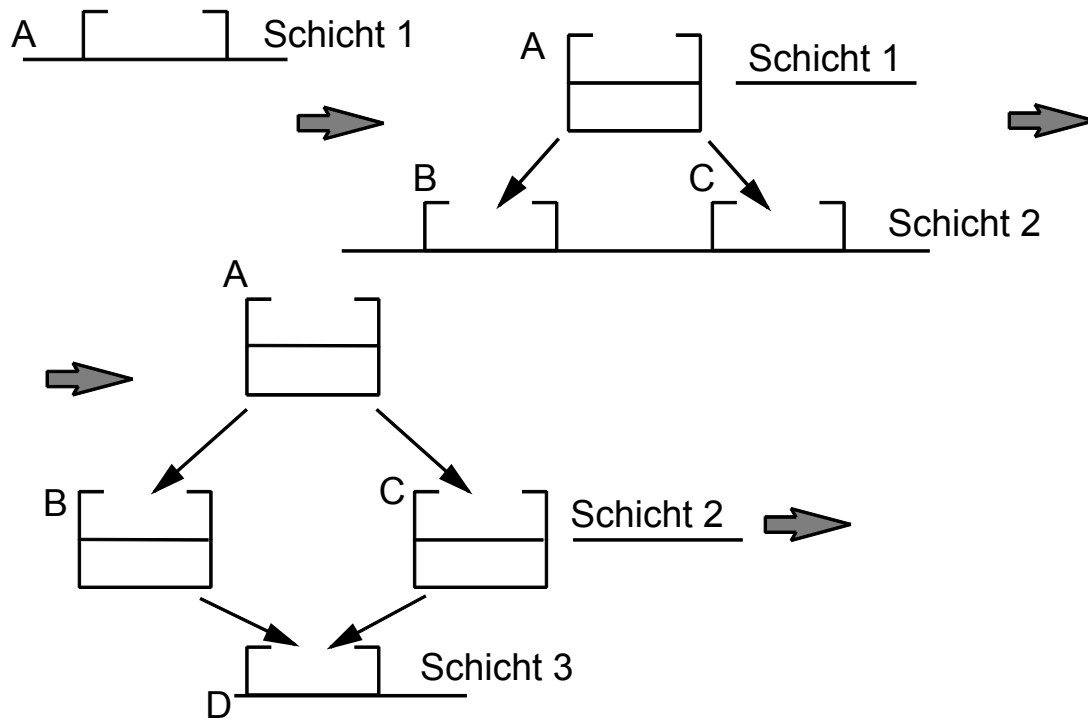
Das Architekturparadigma

- Softwarearchitektur ist idealisierte Vorstellung
(Architekturparadigma, diskretes Paradigma)
 - alle Niveaus einer Architektur bedacht
 - alle Module aufgeführt
 - ohne Modulrümpfe zu betrachten



- Probleme
 - schwer, über alle Niveaus gleichzeitig nachzudenken
 - tiefe Niveaus vage
 - Erkennen der für die Realisierung eines Moduls
 nötigen Ressourcen ohne daß dieser real. wird
 - Module sind schwer zu realisieren: Notwendigkeit
 - Herausziehen von Teilen, (Modulen)
 - Module zu trivial: eingebettet, Verschmelzen auf
 Architekturebene
 - => PiK müßte bereits durchgeführt sein
 - => führt zu vielen Rückgriffen

- kontinuierliches Paradigma: PiG und PiK verzahnt
Spezifikation und Implementation in einem
schichtenweise
Gesamtprogramm entsteht durch Schichten



- Architekturparadigma unvermeidlich, denn bei PiG-PiK-Verzahnung:
 - gäbe es keine Arbeitsteilung (Entw., Impl.,...)
 - entsteht vielleicht gar keine Architektur
 - PiG-PiK-Ebenen gehen durcheinander
 - Erkennen von Gemeinsamkeiten schwierig
 - Überprüfung der Architektur vor Realisierung nicht möglich
- Name "Programmieren im Großen" eigentlich falsch da,
 - nicht wird programmiert
 - Programmieren ist aber mit zu bedenken
- Architektur ist Abstraktion
 - Herausfaktorisieren aller Modulrümpfe
 - versch. von der Abstraktion durch bestimmte Schicht eines Softwaresystems
- große Systeme: Architekturerstellung arbeitsteilig
 - Gesamtarchitektur bis auf Module/Teilsysteme
 - Teilarchitekturen für Teilsysteme
 - obige Idealisierung zwei- oder mehrfach
 - Gesamtarchitektur ohne Modulrealisierungen
 - Gesamtarchitektur ohne Teilsystemarchitekt.

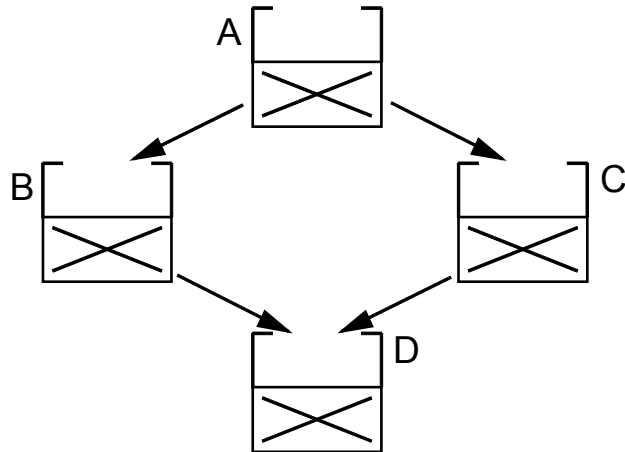
— Entwicklungsstrategien (nach Architekturparadigma
Entwurfsstrategien):

top-down

bottom-up

jo-jo

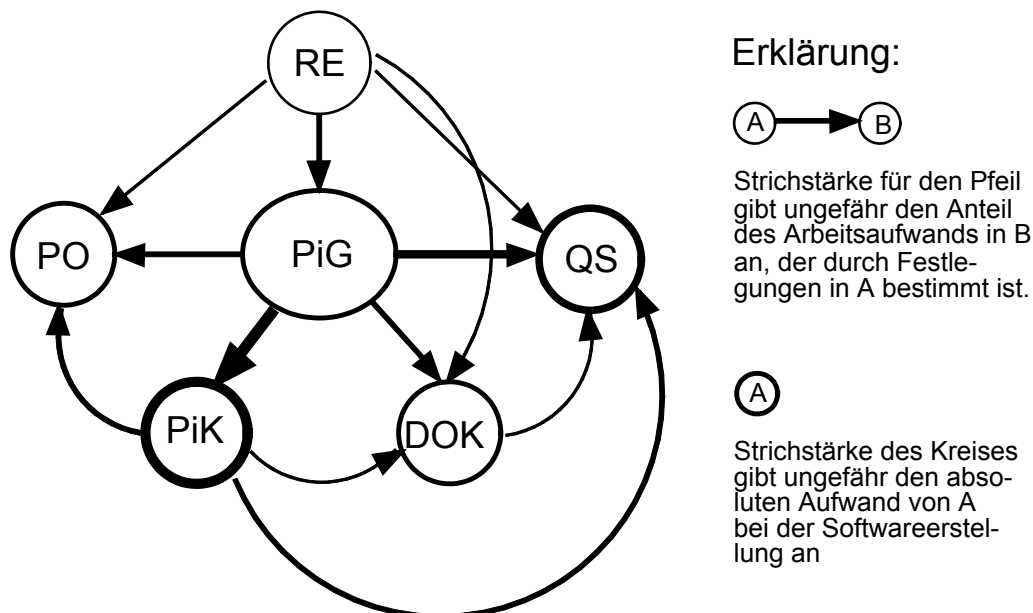
Top-down



Bottom-up

Zur Bedeutung der Architekturmodellierung

- zur Bedeutung von RE und Entwurf (PiG) s. o.
- Verzahnung der Arbeitsbereiche
Beisp. Änderung eines Softwaresystems
- PiG im Zentrum



=> Softwarearchitektur ist die wichtigste
Struktur eines Softwaresystems

- Änderungen übersehen
- Struktureigenschaften erkennen
- Qualitätseigenschaften beurteilen
- Standardlösungen identifizieren
- Wiederverwendbarkeit von Komponenten

— Abgrenzung PiG - Projektorg. insb. Projektmanagement
(Programming in the Many)

Versions- und Konfigurationskontrolle

Verantwortlichkeits- und Zugriffskontrolle

Freigabekontrolle

Nachrichten- und Dokumentverteilungskontrolle

Erfolgskontrolle

bezieht sich auf Architektureinheiten

gehört nicht zum PiG

Der Entwurfsprozeß

- Aufgabe

Zerlegung in Module / Teilsysteme

Def. der Beziehung zwischen Bausteinen

Bauplan (statische Festl. des SW-Systems)

besondere Abstraktion (\Rightarrow Architekturparadigma)

Forderungen an Architekturen und entspr. Sprachen

- allgemeine Forderungen an die Entwurfsspezifikation:
 - Vollständigkeit
 - alle "Funktionen" der Anforderungsdefinition
 - alle internen Bausteine und Festl. ihrer Bez. für die Realisierung
 - nicht vollständig: unterspezifiziert
 - Widerspruchsfreiheit
 - nicht widerspr. frei: z. B. Verletzung kontext sensitiver Regeln
 - Minimalität
 - nur nötige Entwurfsentscheidungen, keine überfl.
 - nicht minimal: überspezifiziert
 - Verständlichkeit
 - Notwendigkeit für die Implementierung
 - Änderbarkeit
 - geringe Veränderungen der Anf. def. sollten geringe Änderungen der Spez. verursachen
- spezifische Forderungen:
 - Adaptabilität
 - Portabilität
 - Wiederverwendbarkeit

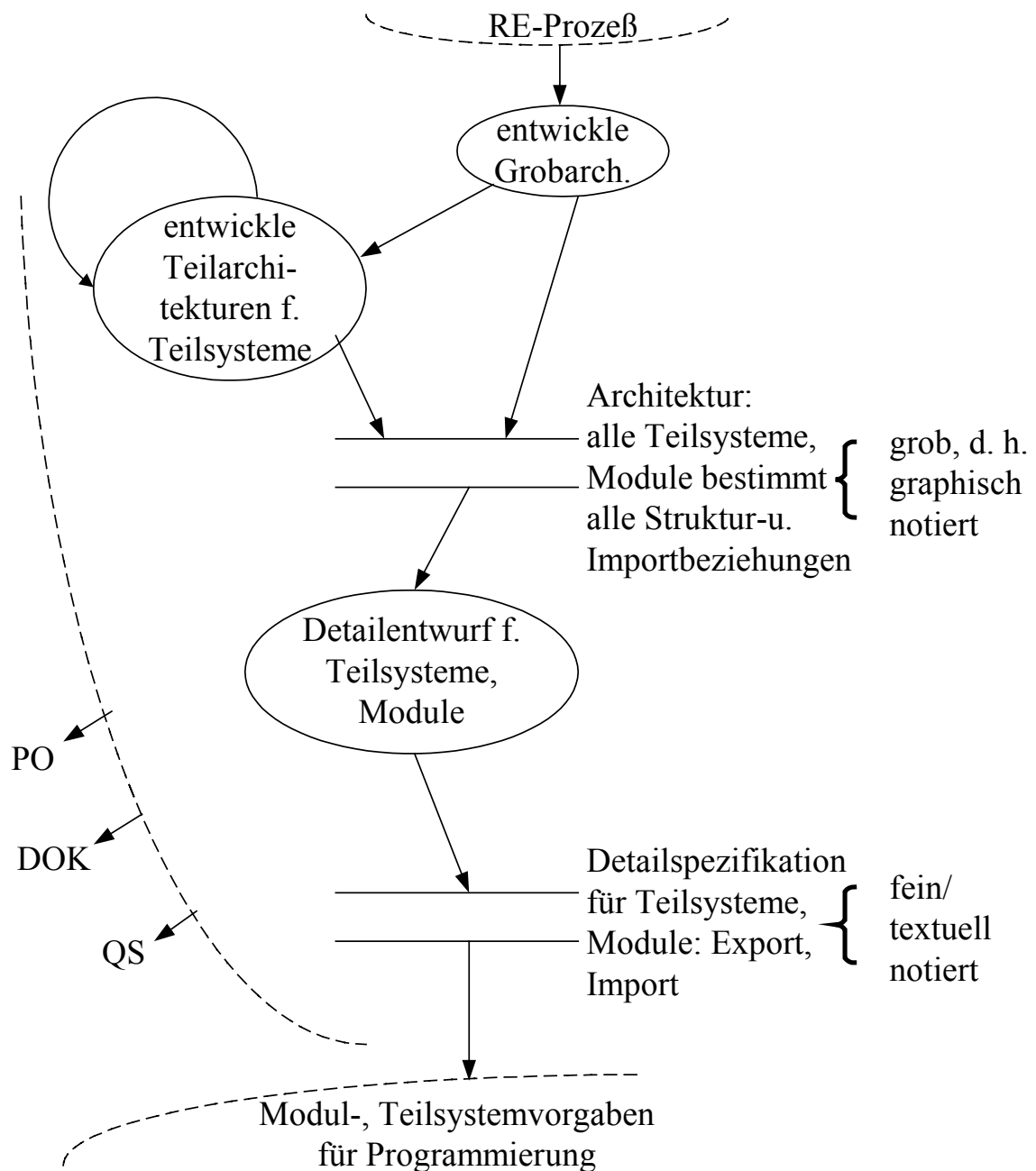
- Forderungen an die Spezifikationsnotation (-methode)
 - Formalisiertheit
 - für Feststellung der Vollständigkeit
 - für Feststellung der Widerspruchsfreiheit
 - für spätere Verifikation der Implementation
 - Implementierungsunabhängigkeit
 - Unterstützung der Vermeidung von Überspezifikation
 - Konstruierbarkeit
 - Spezif. in der Notation/Methode soll leicht erstellbar sein
 - Rezipierbarkeit
 - Notation muß leicht verständlich sein, damit Spezif. leicht verständlich ist
 - Änderbarkeit
 - Unterstützung der Änderungsmögl. einer Spezifikation
 - dies ist notw. jedoch nicht hinreichend für die Änderbarkeit der Spez. selbst
- versch. Rollen bei Erstellung/Prüfung der Entwurfsspezifikation: analog zu RE
 - Entwickler, Qualitätssicherer, Projektleiter, Projektmanager

- Entwurfsteilprozeß innerhalb des Entwicklungsprozessen (ein mögl. Vorgehen)

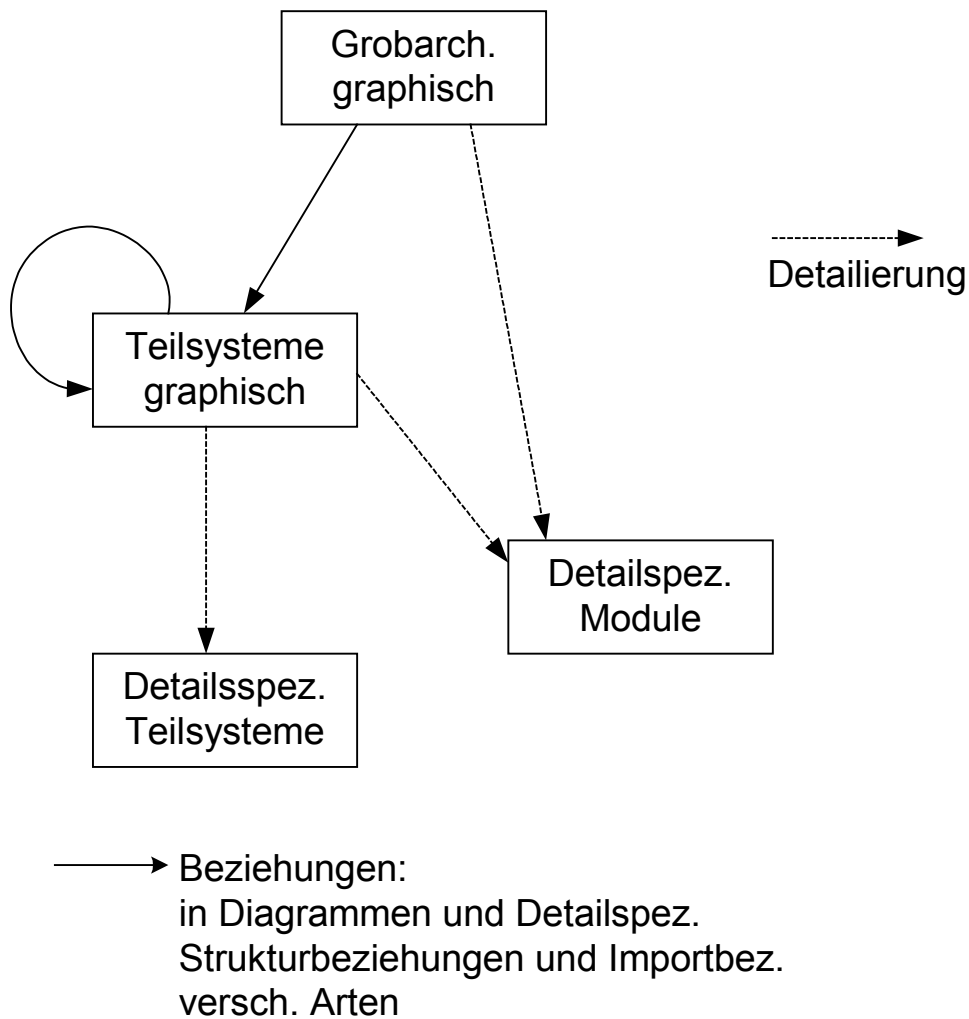
zweistufig: Grob-, Detaildesign (nicht nur zweischichtig)

Entwurfsprozeß ist rekursiv

Rückgriffe unvermeidlich (→ Architekturparadigma)



- die zugehörige Teilkonfiguration
grobgranular
Anzahl der Teilsysteme/Module vorab unbekannt



- vielfältige Beziehungen zu anderen Teilkonfigurationen RE, PIK, PM, DOK, QS