



Praktische Informatik für Wirtschaftsmathematiker,
Ingenieure und Naturwissenschaftler I
(PIWIN I, 3 V + 1 Ü)
WS 2002/03

12. Vorlesungswoche

Softwaretechnik: Vorgehensmodelle und
Beschreibungsnotationen im Softwareentwurf

Unterlagen: Gumm/Sommer, Kapitel 11

Helmut Balzert: Lehrbuch der Software-Technik 1 / 2,
Spektrum Akademischer Verlag, Heidelberg u.a. 2001

UB: Lehrbuchsammlung L Sr 389-1

Folien nach Heinecke (FH Gelsenkirchen), Heyer (Uni Leipzig)



weitere Literatur im Umfeld Software-Engineering

Einige Beispiele:

- Petra Kroha: Softwaretechnologie
Prentice Hall, München u.a. 1997
(DM 54,95) UB: SN 20921
- Peter Forbrig
Objektorientierte Softwareentwicklung mit UML
Fachbuchverlag Leipzig, Leipzig 2001
(DM 29,80; mit CD-ROM) UB: SN 21713
- M. Hitz, G. Kappel
UML@Work, Von der Analyse zur Realisierung, 2. Aufl.
dpunkt.verlag, Heidelberg 2003



Übersicht

- Begriffe
- Herausforderungen in der Software-Entwicklung
- Vorgehensmodelle
- Projektverwaltung
- Planungsphase und Lastenheft
- Definitionsphase und Pflichtenheft
- Entity-Relationship-Modell
- Objektorientierte Analyse

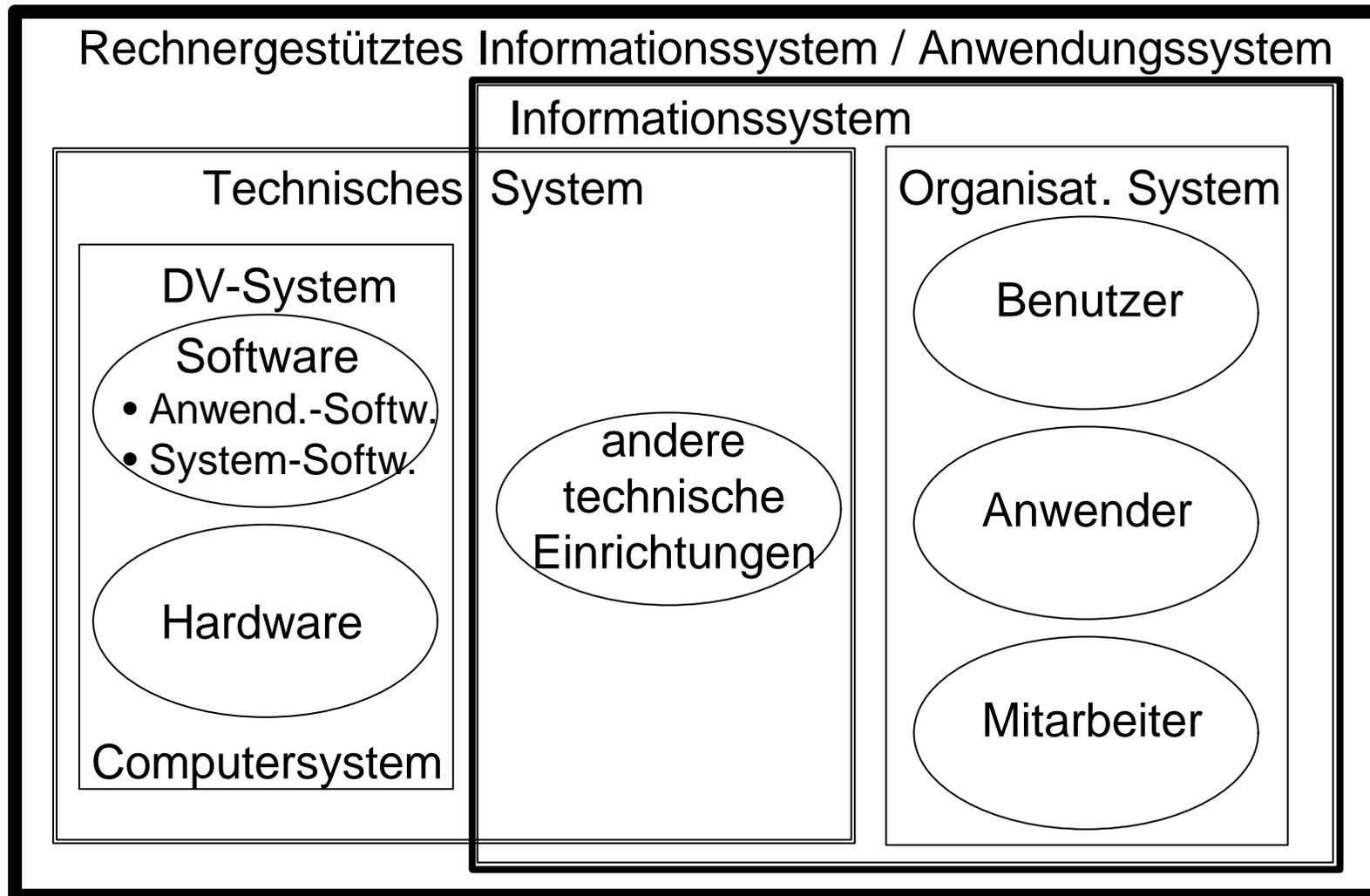


Wiederholung einiger Begriffe

- **Software**
Computer programs, procedures, rules, and possibly associated documentation and data pertaining to the operation of a computer system
- **Software-Produkt \Rightarrow Produkt**
...in sich abgeschlossenes Ergebnis eines erfolgreich durchgeführten Projekts oder Herstellungsprozesses (i.a. für einen Auftraggeber)
- **Software-System \Rightarrow System**
Ausschnitt aus der realen oder gedanklichen Welt, bestehend aus Gegenständen und darauf vorhandenen Strukturen
- **Systemsoftware (Basissoftware)**
Software, die für eine spezielle Hardware(familie) entwickelt wurde, um den Betrieb und die Wartung dieser Hardware zu ermöglichen bzw. zu erleichtern
- **Anwendungssoftware (application software)**
Software, die Aufgaben des Anwenders mit Hilfe eines Computersystems löst
- **Anwender**
(Angehörige einer) Institution, die ein Computersystem zur Erfüllung ihrer fachlichen Aufgaben einsetzt
- **Benutzer**
Personen, die ein Computersystem unmittelbar einsetzen und bedienen



Wiederholung einiger Begriffe





Software-Technik (Software-Engineering)

Die Bereitstellung und systematische Verwendung von Prinzipien, Methoden, Verfahren, Konzepten, Notationen und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von Software-Systemen unter Berücksichtigung von Kriterien wie Kosten, Zeit, Qualität .

- **Prinzipien**
Grundsätze, die dem Handeln zugrunde gelegt werden
- **Methoden**
planmäßig angewandte, begründete Vorgehensweisen
- **Verfahren**
ausführbare Vorschriften oder Anweisungen zum gezielten Einsatz von Methoden
- **Konzepte**
Modellierung eines Sachverhalts unter bestimmten Gesichtspunkten
- **Notationen**
stellt Informationen durch Symbole dar. Ein Konzept kann mit mehreren Notationen dargestellt werden.
- **Werkzeuge (tools)**
dienen der automatisierten Unterstützung von Methoden, Verfahren, Konzepten und Notationen



Warum ist marktreife Software so schwer zu entwickeln?

Will man Produkte für den Markt entwickeln, dann muß jede Produktentwicklung versuchen, folgende **Anforderungen** einzuhalten:

- a) **Funktionstreue**, d.h. die Übereinstimmung der definierten Produkthanforderungen mit dem fertiggestellten Produkt.
- b) **Qualitätstreue**, d.h. die Übereinstimmung der definierten Qualitätsanforderungen mit dem fertiggestellten Produkt.
- c) **Termintreue**, d.h. die Einhaltung der im Entwicklungsplan festgelegten und dem Kunden bzw. dem Marketing zugesagten Fertigstellungstermine.
- d) **Kostentreue**, d.h. die Einhaltung des in der Wirtschaftlichkeits- Rechnung geplanten Personal- und Sachaufwandes für die Produkt-Erstellung und -Pflege.



Software: Unterschiede zu anderen Produkten

- ❏ Software ist ein immaterielles Produkt
- ❏ Software unterliegt keinem Verschleiß
- ❏ Software ist frei von physikalischen Restriktionen
- ❏ Software ist leicht und schnell zu verändern
- ❏ Software kennt keine Ersatzteile
- ❏ Software veraltet
- ❏ Software ist schwer zu vermessen

Veränderungen der Software-Systeme, Herausforderungen

- ☞ zunehmende Bedeutung der Software
- ☞ wachsende Komplexität
- ☞ zunehmende Qualitätsanforderungen
- ☞ Nachfragestau und Produktionsengpässe
- ☞ Zunahme von Standard-Software
- ☞ Zunahme von „Altlasten“ (Stichwort: Legacy-Systems)
- ☞ Zunahme des „Outsourcing“



Herausforderung: Qualitätsanforderungen

1977:	7 - 20	Fehler auf 1000 Zeilen Quellcode
1994:	0,05 - 0,2	Fehler auf 1000 Zeilen Quellcode

ABER:

- zunehmende Komplexität
- 0,1%-Fehlerquote übertragen auf andere Bereiche:
 - 18 Flugzeugabstürze pro Tag
 - 16.000 verlorene Briefe pro Tag
 - 22.000 fehlgebuchte Schecks pro Stunde

Anforderungen an ein Softwareprodukt

- Korrektheit, Verifizierbarkeit
- Zuverlässigkeit, Robustheit
- Effizienz
- Ergonomie / Benutzungsfreundlichkeit
- Wartbarkeit: Korrigierbarkeit, Erweiterbarkeit
- Wiederverwendbarkeit, Portabilität
- spezielle Anwendungsforderungen, z.B. Echtzeitfähigkeit



Schwierigkeit: Änderungen während der Entwicklung

Während der Produktentwicklung ändern sich

- ✂ Produktanforderungen
- ✂ Hardware-Komponenten
- ✂ System-Software
- ✂ Entwicklungswerkzeuge und -methoden

Generationswechsel

- ⌚ der Anwendungssoftware alle 10 bis 15 Jahre
- ⌚ der Systemsoftware alle ca. 6 Jahre
- ⌚ der Hardware alle ca. 3 Jahre

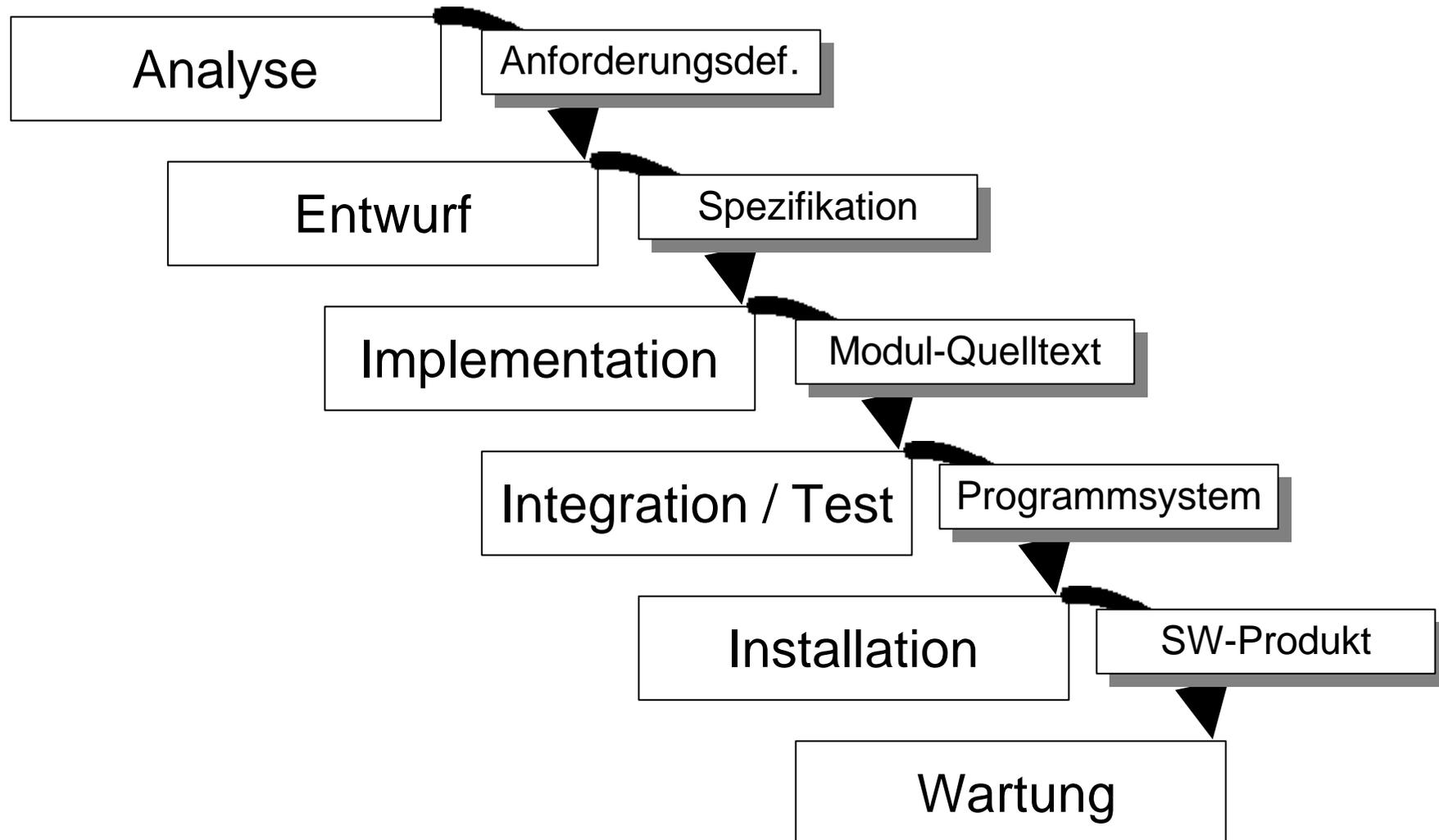


Vorgehensmodelle, eine Auswahl

- Wasserfall-Modell
- Modifizierte Phasenmodelle
- Spiralmodelle
- Transformationsmodelle
- Prototyping und evolutionäre Entwicklung



Wasserfall-Modell





Wasserfall-Modell

in der Praxis häufig eingesetzt

Vorteile

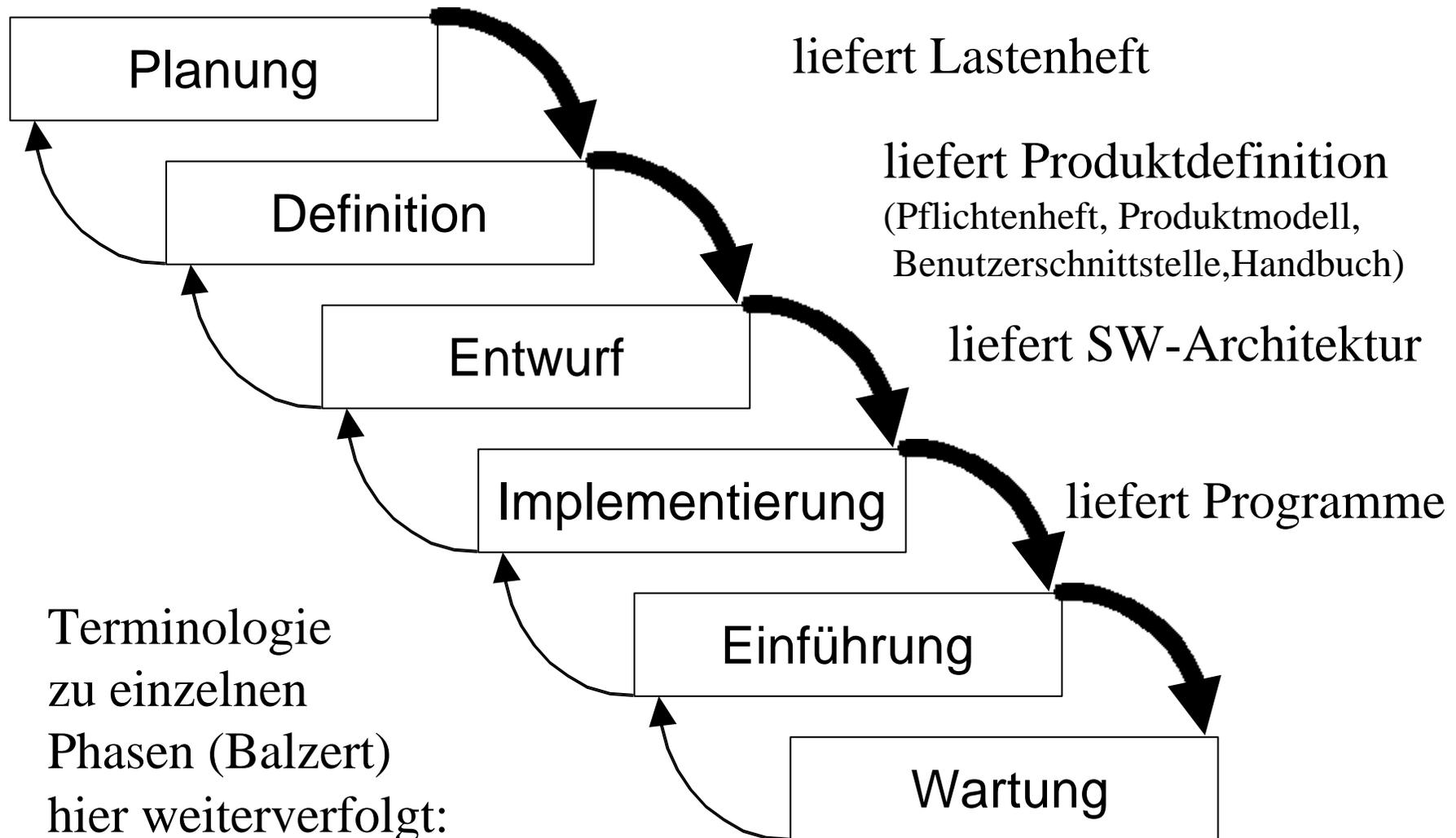
- ✎ klare Trennung mit Meilensteinen vereinfacht Projektplanung, Kosten- und Personalzuordnung (-> Hoffnung)

Nachteile:

- ✎ Abgrenzung der Phasen ist schwierig.
 - ✎ Phasen sind nicht streng sequentiell.
 - ✎ Ergebnis einer Phase ist selten fehlerfrei.
 - ✎ Fehler fallen nicht schon in der nächsten Phase auf.
 - ✎ erfordern in späteren Phasen erheblichen Aufwand
 - ✎ erfordern in frühere Phasen zurückzukehren (obwohl eigentlich nicht möglich)
 - ✎ Wartung ist eigentlich ein neuer Durchlauf des Modells.
-
- ☞ Rückkopplung ist nötig.
 - ☞ Varianten:
 - ☞ Wasserfall-Modell mit Rückkopplung
 - ☞ V-Modell
 - ☞ W-Modell



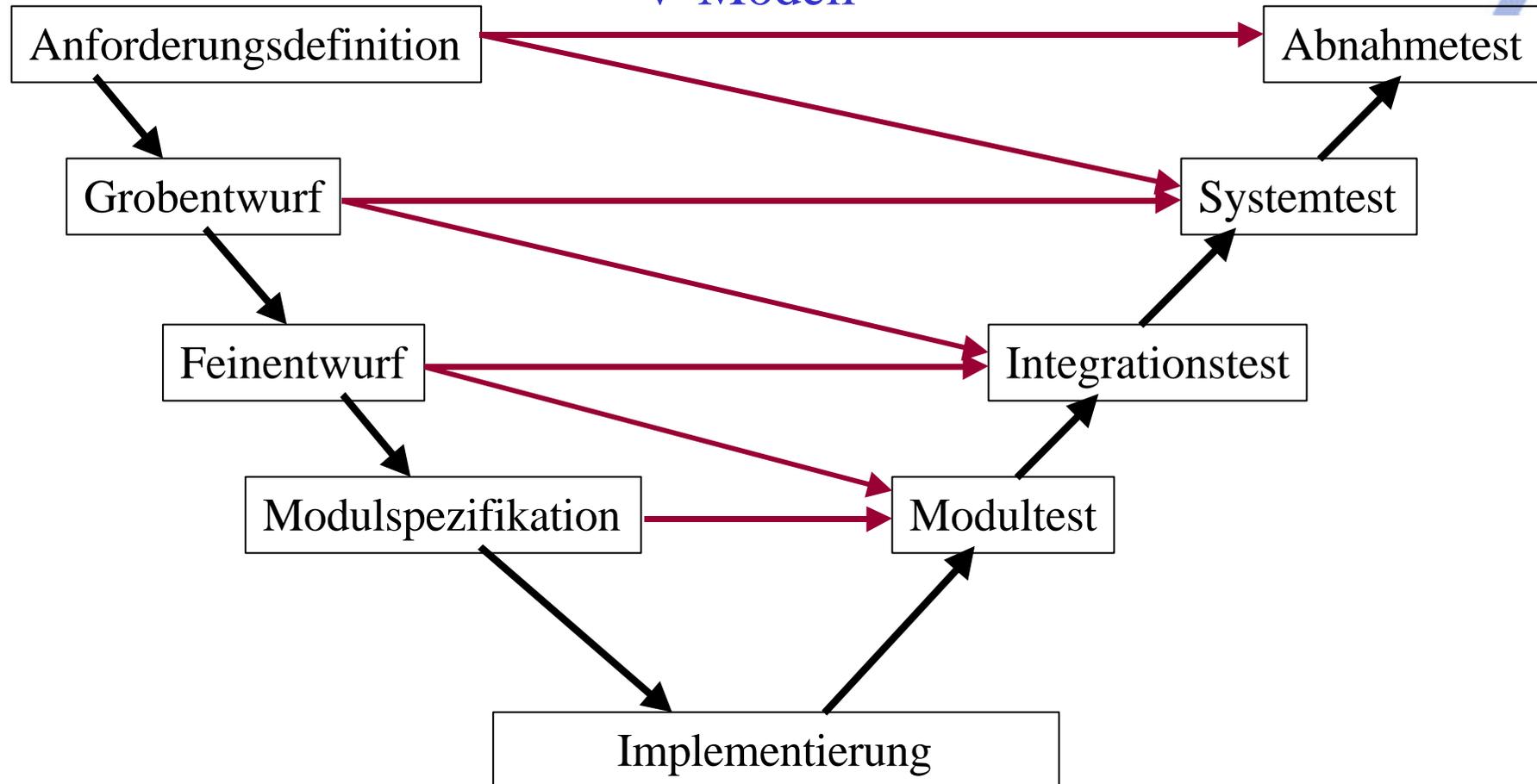
Modifizierte Phasenmodelle



Terminologie
zu einzelnen
Phasen (Balzert)
hier weiterverfolgt:
insb. Planungsphase,
Definitionsphase



V-Modell



W-Modell:

Parallel zum linken Ast: Vorbereitung der Testaktivitäten dieser Ebene

Parallel zum rechten Ast: Debuggen & Ändern



Spiralmodelle

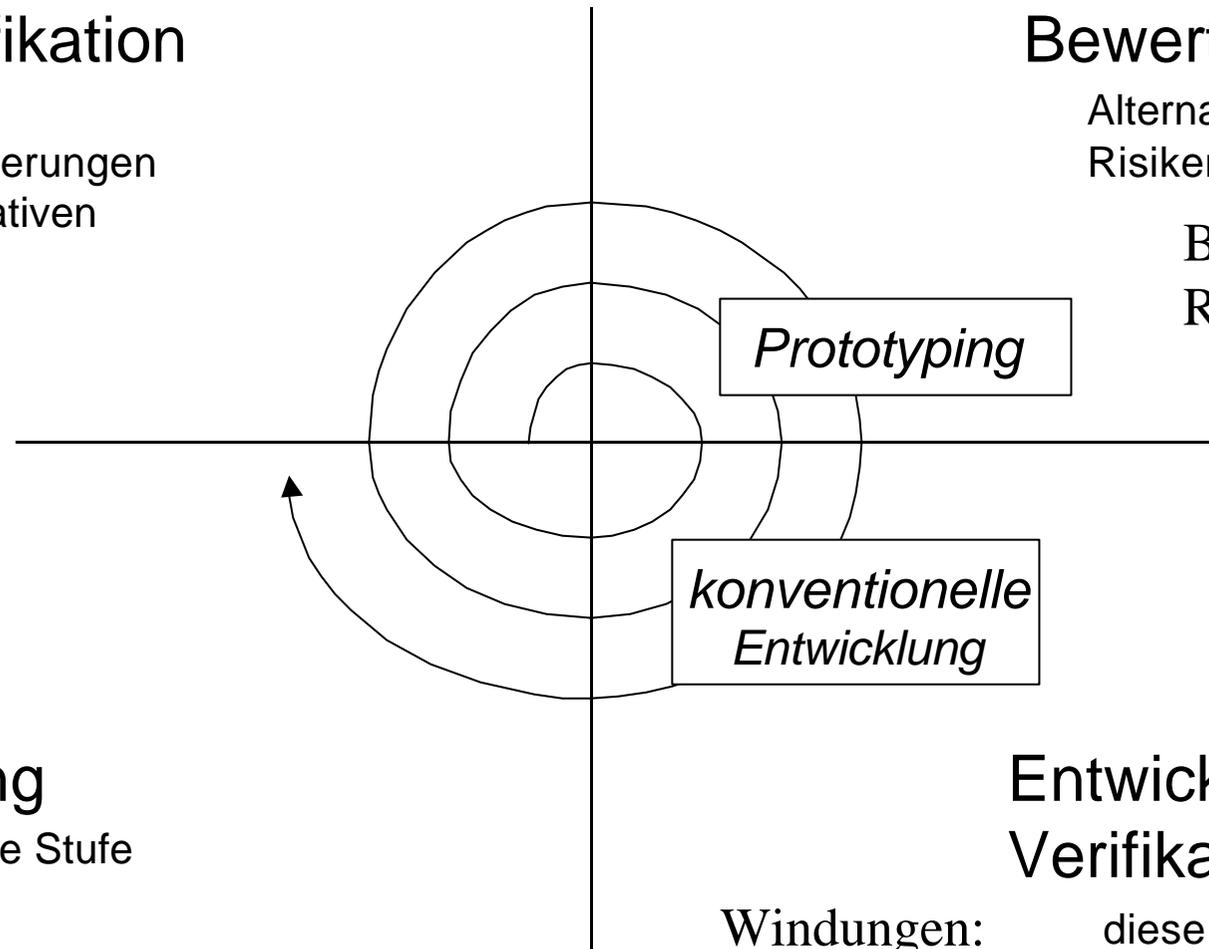
Identifikation

Ziele
Anforderungen
Alternativen

Bewertung

Alternativen
Risiken

Besonderheit:
Risikoanalyse



Prototyping

*konventionelle
Entwicklung*

Planung

nächste Stufe

Entwicklung

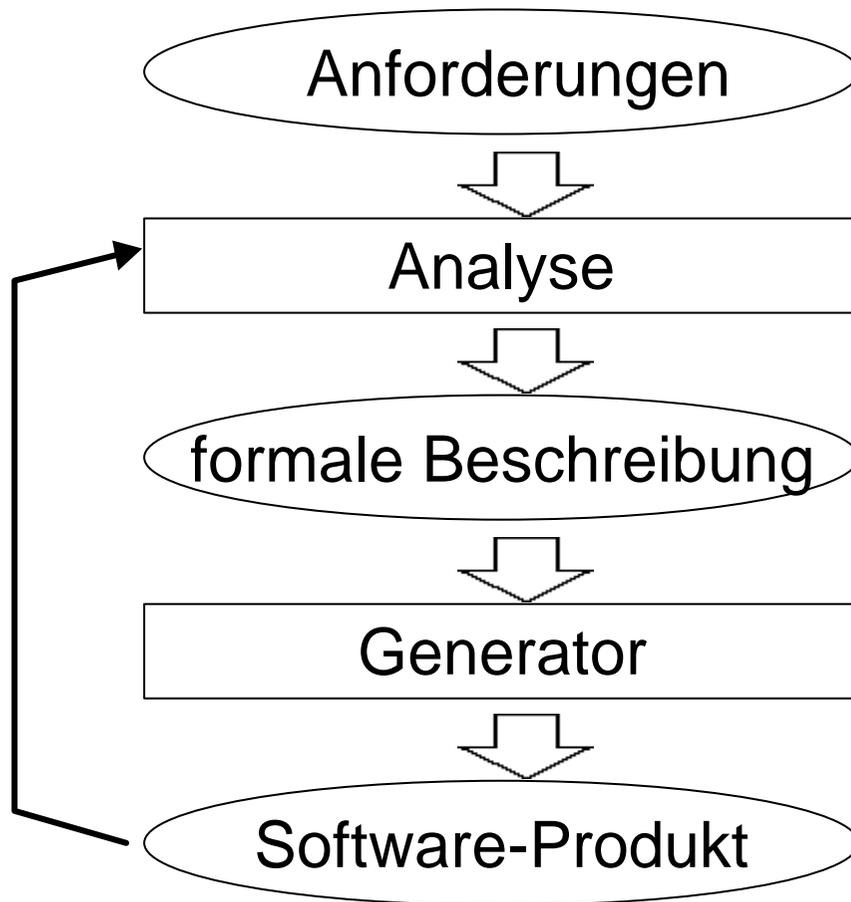
Verifikation

Windungen: diese Stufe

- Anforderungen
- Entwurf
- Implementierung



Transformationsmodelle



Vorteile:

- 👍 Kein Verständigungsproblem
- 👍 Kein Wartungsproblem
- 👍 Konsistenz zwischen Dokumentation und Produkt
- 👍 Korrektheit gesichert

Nachteile:

- 👎 Formale Beschreibung aufwendig und kompliziert
- 👎 Automatische Generierung schwierig
- 👎 Wenig praktische Erfahrungen



Prototyping

Einsatz von Prototypen:

- ☞ zur Anforderungsanalyse:
exploratives Prototyping
- ☞ zur Verbesserung des Entwurfs:
experimentelles Prototyping
- ☞ zur Implementation:
evolutionäres Prototyping

Arten von Prototypen:

- ☞ vertikaler Prototyp: einzelne Funktionen im Detail
- ☞ horizontaler Prototyp: alle Funktionen im Überblick

Vorgehensmodelle mit Prototyping:

1. Evolutionäre Entwicklung:
Spezifikation, Entwicklung, Validierung simultan
2. Inkrementelle Entwicklung: kombiniert Wasserfall mit evolutionär:
Grundfunktionalität wird iterativ erweitert

Vorteile:

- ☑ besseres Verständnis zwischen Kunde und Entwickler
- ☑ frühzeitige Einbeziehung der Benutzer möglich
- ☑ einheitlichere Benutzungsoberfläche

Nachteile:

- ☹ Gefahr einer inhomogenen Software-Architektur
- ☹ Doppelentwicklung von Modulen möglich
- ☹ häufiges Überarbeiten bestehender Module



Projektverwaltung

- Produktorganisation
 - ◆ Versionsmanagement
 - ◆ Konfigurationsmanagement
- Projektorganisation
 - ◆ Projektplanung
 - ◆ Teammanagement
 - ◆ Ablauforganisation



Versionsmanagement

Versionen

verschiedene Ausprägungen derselben Software (Modul, Programm, Programmsystem)

entstehen durch

- Fehlerkorrektur
- Erweiterung des Leistungsumfangs
- Anpassung an verschiedene Systemsoftware
- Anpassung an verschiedene Hardware

Versionen können

- einander ablösen (Revisionen)
- nebeneinander bestehen (Varianten)

Dokumentation zu jeder Version nötig

- Unterschiede zu Vorversionen
- Unterschiede zu anderen Varianten
- Gültigkeit

Werkzeuge: z.B. rcs, ccs



Konfigurationsmanagement

Konfigurationen

verschiedene Ausprägungen desselben Software-Produkts (lauffähiges Gesamtsystem)

aus Versionen von Komponenten zusammengestellt für

- bestimmte Systemsoftware
- bestimmte Hardware
- bestimmte Arbeitsumgebung
- bestimmten Leistungsumfang

Probleme:

- Mehrfachverwendung von Modulen
- Abhängigkeiten zwischen Komponenten
- Zugriffsrechte

bei größeren Projekten ist Unterstützung durch Software nötig

- Standard-Hilfsmittel wie make
- CASE-Tools



Projektplanung

Projektphasen (nicht nur bei Software-Projekten) aus Sicht des Managements

- ✎ Projektidee
- ✎ Zieldefinition
- ✎ Machbarkeitsstudie
- ✎ Grobgliederung
- ✎ Feinstrukturierung
- ✎ Arbeitsplanung
- ✎ Ablaufüberwachung

⇒ die gleichen Probleme wie bei Phasenmodellen der Softwareentwicklung!

Projektziele

- Sachziel: Was soll das Projekt leisten?
- Kostenziel: Was darf das Projekt kosten?
- Terminziel: Wann soll das Projekt fertig sein?

Probleme:

- 💣 häufig keine geeignete Datenbasis (Pilotprojekt?)
- 💣 Zielkonflikte
- 💣 Vorgaben politischer Natur



Aspekte der Projektplanung: Teammanagement

Beteiligte im Softwarehaus:

- Projektleiter, Entwickler,
- ggf. weitere (Projektkoordinator, Controlling)

Beteiligte beim Auftraggeber:

- Projektauftraggeber, ggf. Projektmitarbeiter,
- ggf. weitere (Betriebsrat, Datenschutz, Revision)

externe Beteiligte

- Berater, Gutachter
- Prüf- und Genehmigungsinstitutionen

Ablauforganisation

Planung des Projektablaufs

- Gliederung in Vorgänge
- Ermittlung der Beziehungen zwischen Vorgängen
- Zuordnung der Ressourcen zu Vorgängen
- Festlegung von Meilensteinen

Hilfsmittel

- Zeitbalken (Gantt-Diagramme)
- Netze (PERT / CPM)
- Projektplanungssoftware



Planungsphase

Bei der Planungsphase geht es darum,

- wie das Produkt gestaltet sein soll (Lastenheft)
- wie und mit welchem Aufwand das Produkt erstellt wird (Projektplan & Projektkalkulation)

Übersicht

- Vorgehen bei der Produktplanung
- Durchführbarkeitsstudie
- Erstellung eines Lastenheftes
- Gliederung für ein Lastenheft
- Aufwand für die Produkterstellung
- Produktumfang und Produktivität
- Einflußfaktoren auf den Aufwand
- Methoden zur Aufwandsschätzung



Vorgehen bei der Produktplanung

- ◆ **Auswählen des Produkts**
 - Trendstudien, Marktanalysen, Forschungsergebnisse
 - Kundenanfragen
 - Vorentwicklungen
- ◆ **Voruntersuchung des Produkts**
 - Ist-Aufnahme
 - Festlegen der Hauptanforderungen
- ◆ **Durchführbarkeitsuntersuchung**
 - Fachliche Durchführbarkeit
 - Personelle Durchführbarkeit
 - Risiken und Alternativen
- ◆ **Ökonomische Betrachtung**
 - Aufwands- und Termschätzung
 - Rentabilität



Ergebnis der Planungsphase

- Lastenheft (grobes Pflichtenheft)
- Projektkalkulation
- Projektplan

Planungsphase stark unterschiedlich nach

- ◆ Firmensituation
- ◆ Personal
- ◆ Produktart



Erstellung eines Lastenhefts

- ◆ **Aufgabe:**
 - Darstellung der fachlichen Grundanforderungen des Produkts
- ◆ **Adressaten:**
 - Auftraggeber, Auftragnehmer
- ◆ **Inhalt:**
 - Nur Grundanforderungen - was, nicht wie.
- ◆ **Sprache:**
 - verbale Beschreibung, angemessene Abstraktion, Nummerierung
- ◆ **Aufbau:**
 - übersichtliches Gliederungsschema
- ◆ **Zeitpunkt:**
 - erstes Dokument der Anforderungen an das neue Produkt



Gliederung für ein Lastenheft

- ◆ Zielbestimmung
- ◆ Produkteinsatz
 - Anwendungsbereich
 - Zielgruppen
- ◆ Produktfunktionen
 - Nur Hauptfunktionen, numeriert mit /LF10/ etc. (Zehnerschritte)
- ◆ Produktdaten
 - Hauptdaten des Produkts, numeriert mit /LD10/ etc.
- ◆ Produktleistungen
 - Leistungsanforderungen bzgl. Zeit, Genauigkeit etc., /LL10/ ...
- ◆ Qualitätsanforderungen
- ◆ Ergänzungen



Lastenheft-Beispiel: Szenario

- Die Daten der an einem Lehrstuhl beschäftigten HiWis sollen in einer Datei verwaltet werden.
- Über jeden HiWi sind folgende Informationen aufzubewahren:
 - Name, Vorname, Matrikelnummer, Geburtsdatum,
 - Heimatadresse mit TelefonNr.,
 - Studienadresse mit TelefonNr.,
 - Studienfach (ggfs mehrere), Semesterzahl,
 - Vordiplom (Ja / Nein, wenn ja, dann Note und Datum),
 - Diplom (Ja / Nein, wenn ja, dann welche Note und Datum),
 - Beschäftigungszeiten (von, bis, Stunden-zahl pro Woche, Tätigkeitsgebiete, benutzte SoftwareSysteme),
 - Arbeitszeugnis ausgehändigt (Ja/Nein, wenn ja, dann Datum).
- Aus den Daten ist folgende Liste zu erstellen:
 - Titel: »Alphabetische Liste aller HiWis«
je HiWi: Name, Vorname, MatrikelNr., Studienadresse, Geburtsdatum.
- je HiWi müssen sämtliche Daten erzeugt, geändert, ausgegeben werden können.



Lastenheft-Beispiel: Lastenheft (Teil 1)

1 Zielbestimmung

Das Programm HIWI-Verwaltung soll einen Lehrstuhl in die Lage versetzen, die beschäftigten HiWis zu verwalten.

2 Produkteinsatz

Das Produkt wird im Sekretariat eines Lehrstuhls eingesetzt und von der Sekretärin bedient.

3 Produktfunktionen

- /LF10/ Ersterfassung, Änderung und Löschung von Hilfsassistenten
- /LF20/ Ausgabe einer alphabetischen Liste aller Hilfsassistenten mit folgenden Daten: Name, Vorname, MatrikelNr., Studienadresse, Geburtsdatum
- /LF30/ Ausgabe sämtlicher Daten eines Hilfsassistenten
- /LF40/ Gezielte Abfragen mit einer Endbenutzersprache



Lastenheft-Beispiel: Lastenheft (Teil 2)

4 Produktdaten

- /LD10/ Folgende Daten sind über jeden Hilfsassistenten zu speichern:
 - Name, Vorname, Matrikelnummer, Geburtsdatum,
 - Heimatadresse mit TelefonNr.,
 - Studienadresse mit TelefonNr.,
 - Studienfach (ggfs mehrere), Semesterzahl,
 - Vordiplom (Ja / Nein, wenn ja, dann Note und Datum),
 - Diplom (Ja / Nein, wenn ja, dann welche Note und Datum),
 - Beschäftigungszeiten (von, bis, Stunden-zahl pro Woche, Tätigkeitsgebiete, benutzte SoftwareSysteme),
 - Arbeitszeugnis ausgehändigt (Ja/Nein, wenn ja, dann Datum).

5 Produktleistungen

- /LL10/ Die Funktionen /LF10 / und /LF40/ dürfen nicht mehr als 2 Sekunden Antwortzeit benötigen.
- /LL20/ Es müssen max. 100 HiWis verwaltet werden können.



Lastenheft-Beispiel: Lastenheft (Teil 3)

6 Qualitätsanforderungen

	sehr gut	gut	normal	nicht relevant
Produktqualität				
Funktionalität		x		
Zuverlässigkeit			x	
Benutzbarkeit		x		
Effizienz			x	
Änderbarkeit			x	
Übertragbarkeit				x



Aufwand für die Produkterstellung

Gewinn (bzw. Verlust!) = Verkaufserlös - **Kosten**

- ◆ bei Standardsoftware Schätzung:
Erlös = (VK-Preis - laufende Stückkosten) x Menge
- ◆ bei Individualsoftware meist nur Kosten beeinflussbar

Kosten

- ◆ **Personalkosten** <- wesentlicher Kostentreiber
- ◆ Kosten der Entwicklungsumgebung
(Hardware, Systemsoftware, Tools)
- ◆ ggf. Lizenzkosten
- ◆ sonstige Kosten
(Verbrauchsmaterial, Reisekosten etc.)



Produktumfang und Produktivität

Umfang des Produkts

- ◆ LOC: Lines of Code, geschätzt (Maß ist problematisch!)
- ◆ Function Points: ermittelt aus Produkthanforderungen

Produktivität eines Mitarbeiters/Mitarbeiterin

- ◆ LOC / Monat oder LOC/Jahr

Faustregel: 350 LOC / Ingenieurmonat über alle Phasen

Aufwand eines Produkts

- ◆ MJ: Mitarbeiterjahre (Mannjahre, Menschjahre)
- ◆ MM: Mitarbeitermonate (auch Personenmonate PM)
- ◆ Vorsicht: 1 MJ = maximal 10 MM

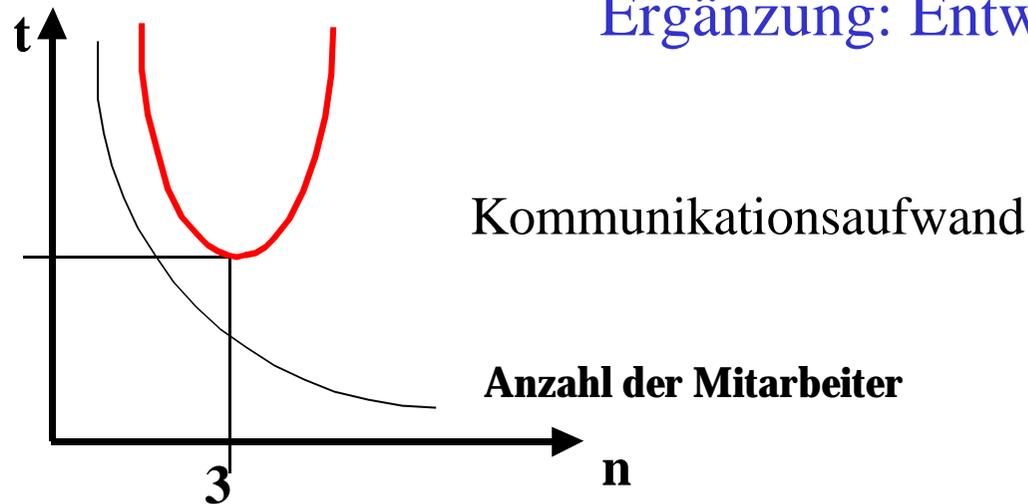


Einflußfaktoren auf den Aufwand

- ◆ Quantität
 - Größe in LOC
 - Funktions- und Datenumfang
 - Komplexität
- ◆ Qualität
 - verschiedene Qualitätsmerkmale
 - schwer meßbar
- ◆ Entwicklungsdauer
 - abhängig von der Anzahl der Mitarbeiter bzw. umgekehrt
 - sinnvoller Erfahrungswert (in der Literatur):
 $2,5 \times (\text{Aufwand in MM})^{0,35}$ für Online-Systeme
- ◆ Produktivität
 - Motivation, Erfahrung, Lernfähigkeit



Ergänzung: Entwicklungsdauer



Optimale Entwicklungsdauer = $2,5 * (\text{Aufwand in MM})^s$

mit $s = 0,38$ für Stapel-Systeme
 $s = 0,35$ für Online-Systeme
 $s = 0,32$ für Echtzeit-Systeme

Es wird geschätzt, dass der Entwicklungsaufwand für ein neues Online-System neun Mitarbeiter-Monate beträgt. Als optimale (im Sinne des angenommenen math. Zusammenhangs) Entwicklungsdauer ergibt sich:

Dauer = $2,5 * 9^{0,35} = 5,3$ Monate, die durchschnittliche Größe des Entwicklungsteams beträgt:

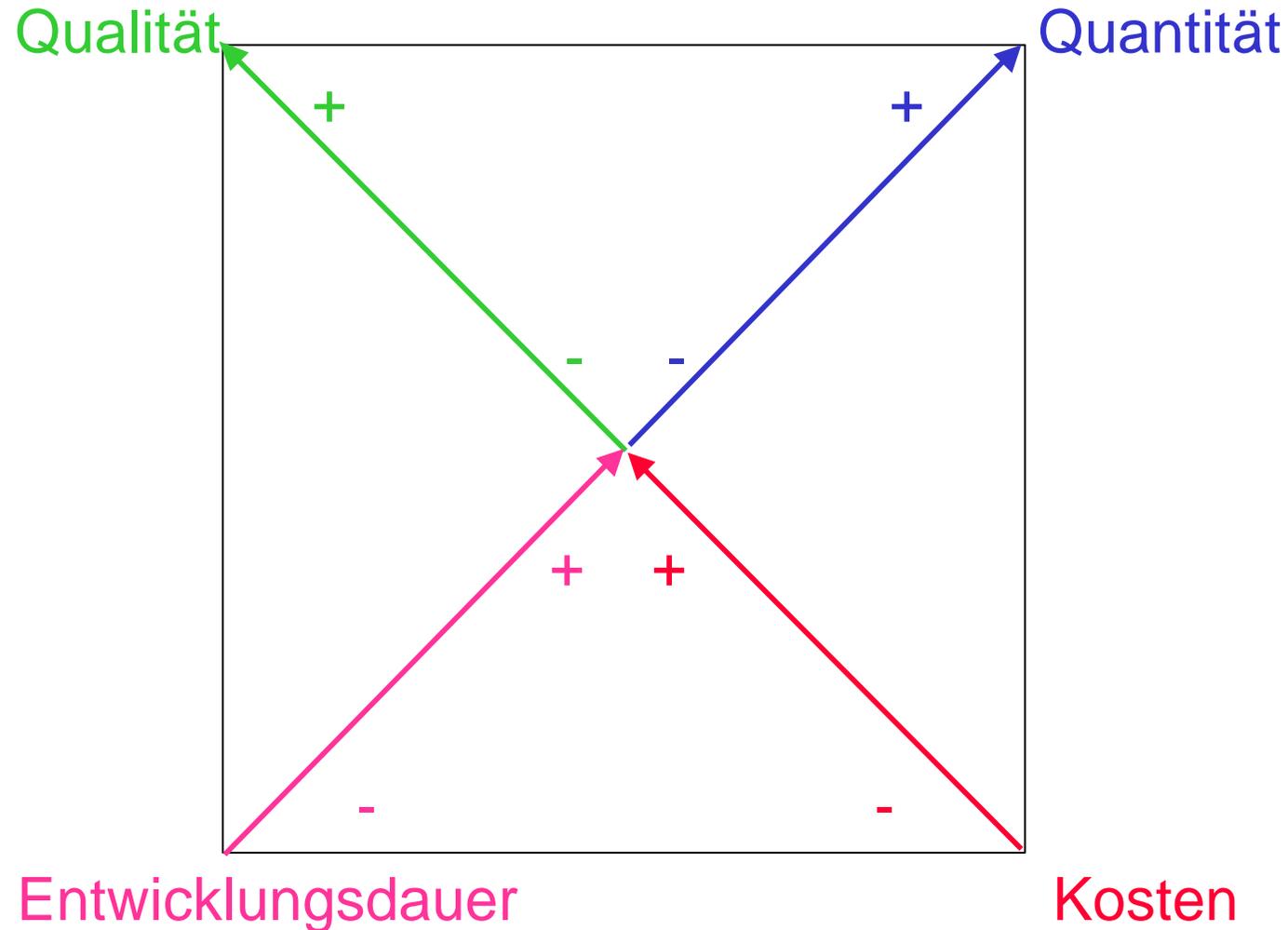
Anzahl der Mitarbeiter = $9 \text{ MM} / 5,3 \text{ Monate} = 1,7 \approx 2$



Einflußfaktoren auf den Aufwand

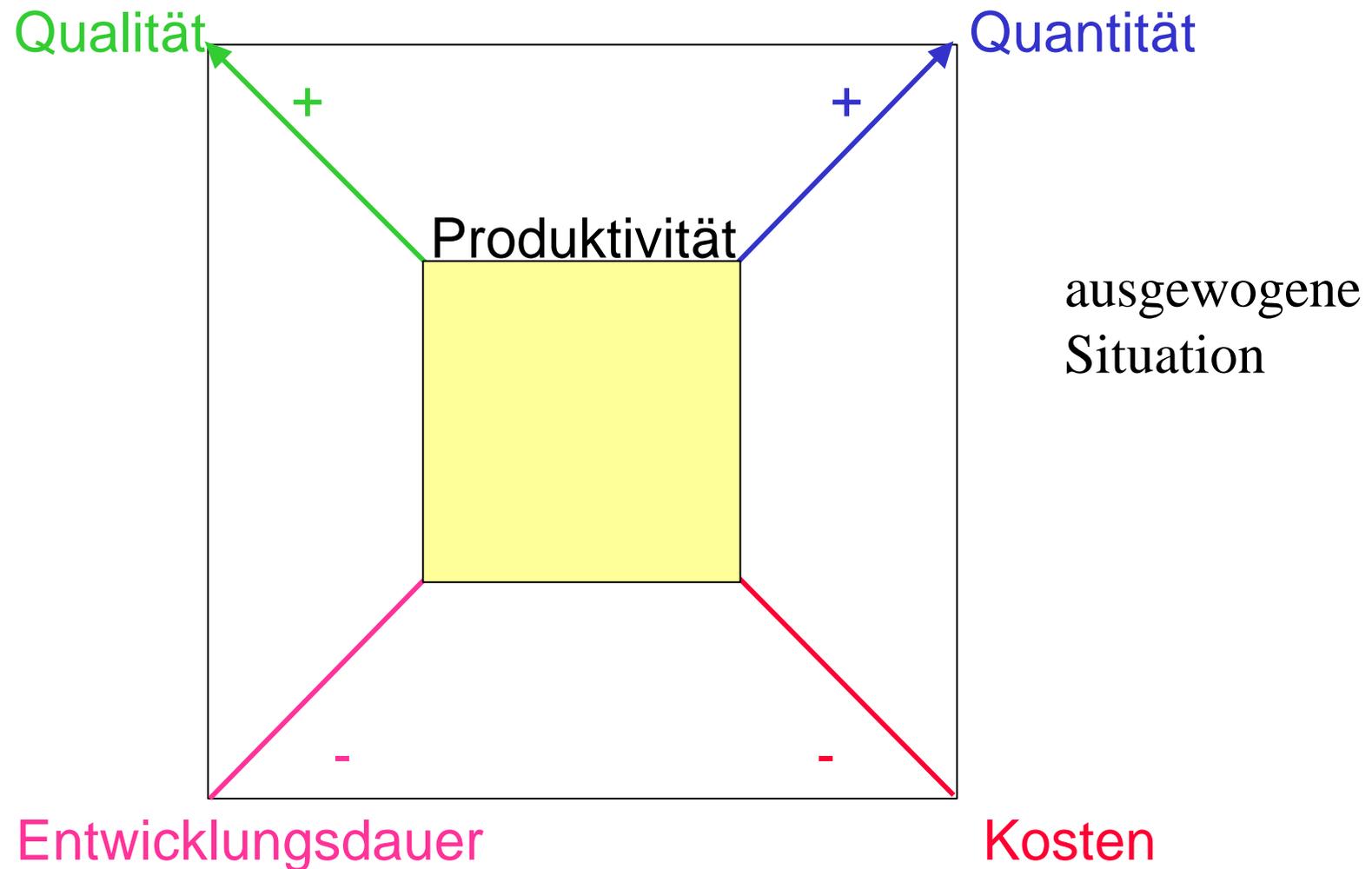
„Teufelsquadrat“ nach Sneed

Produktivität einer Arbeitsgruppe entspricht konstanter Fläche



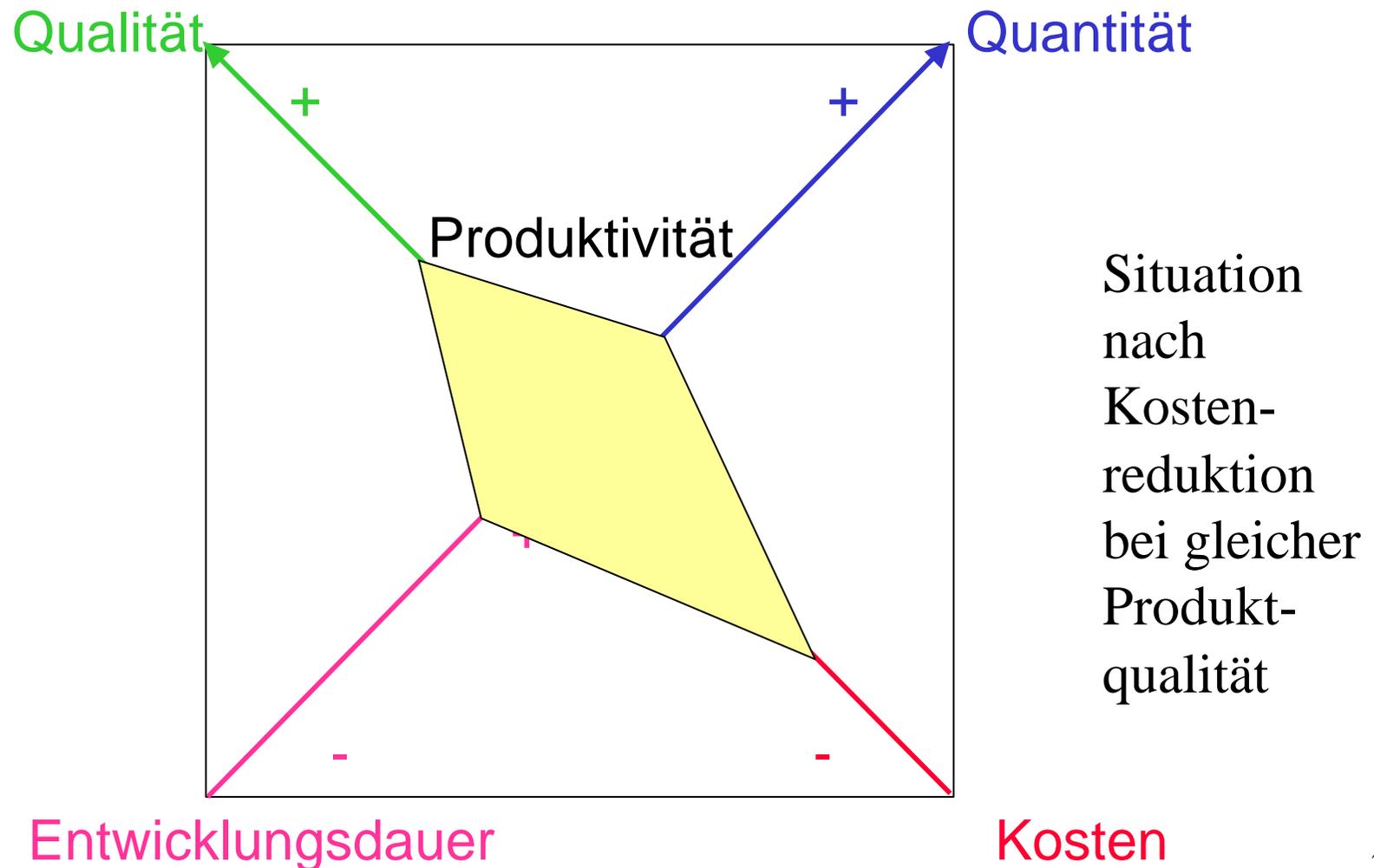


Einflußfaktoren auf den Aufwand





Einflußfaktoren auf den Aufwand





Methoden zur Aufwandsschätzung

Grundproblem:

Wie lässt sich der erforderliche Zeitaufwand vorab sinnvoll schätzen ?

Ganze Reihe von Methoden bekannt (Achtung: kein Stein der Weisen bekannt!)

- ◆ Analogiemethode
- ◆ Relationsmethode
- ◆ Multiplikatormethode
- ◆ Gewichtungsmethode
- ◆ Prozentsatzmethode
- ◆ Methode der parametrischen Gleichungen

davon abgeleitet zahlreiche Kombinationsmethoden,

z.B. Function-Point-Methode



Analogiemethode

in Analogie zu abgeschlossenen „ähnlichen“ Projekten

- ähnliches Anwendungsgebiet, ähnlicher Produktumfang
- ähnlicher Komplexitätsgrad, gleiche Programmiersprache etc.

Erfahrung und Faustregeln, z.B.

- Software-Entwicklungen, die hauptsächlich vorhandene Software wiederverwenden, benötigen nur etwa $\frac{1}{4}$ der Zeit und anderer Ressourcen von Neuentwicklungen

Resultate

- basieren auf individueller Erfahrung und Intuition
- nicht verallgemeinerbar, nicht allgemein nachvollziehbar

etwas formalisierter: **Relationsmethode**

- Faktorenlisten, z.B. Programmiersprache, Programmiererfahrung
- Richtlinien zur Bewertung
 - Punktwerte für einzelne Faktoren
 - Ermittlung des Mehr-/Minderaufwands in Punkten
- Richtlinien zur Umrechnung, Punkte -> Prozentaufschläge



Multiplikatormethode

- Zerlegung in Teilprodukte, z.B.
 - Steuerprogramm
 - Datenverwaltung
 - E/A-Programm
- Ermittlung des Aufwands pro Teilprodukt (meist LOC)
 - Steuerprogramm 1 x 500 LOC
 - Datenverwaltung 1 x 800 + 2 x 250 = 1300 LOC
 - E/A-Programm 1 x 700 + 2 x 500 = 1700 LOC
- Bewertungsfaktor, z.B.
 - Steuerprogramm 1,8 x 500 LOC = 900 LOC bew.
 - Datenverwaltung 1,0 x 1300 LOC = 1300 LOC bew.
 - E/A-Programm 1,5 x 1700 LOC = 2550 LOC bew.
- Umrechnung in MM



Prozentsatzmethode

- Ermittlung des prozentualen Aufwands pro Phase aus möglichst vielen Projekten
 - Definition 18% ± 10% (a.A. 30%)
 - Entwurf 19% ± 10% (a.A. 30%)
 - Codierung 34% ± 10% (a.A. 15-20%)
 - Test 29% ± 10% (a.A. 20-25%)
- Ermittlung des Aufwands einer bestimmten Phase des neuen Projekts
 - durch vollständige Bearbeitung der (ersten) Phase
 - durch sorgfältige Schätzung der (ersten) Phase
- Errechnung des Gesamtaufwands des Projekts

Achtung:
Codierung i.d.R.
nicht dominant!!!

Parametrische Gleichungen

- Ermittlung des wertmäßigen Einflusses von Faktoren auf den Gesamtaufwand
 - Statistische Analyse (Korrelationsanalyse) zahlreicher abgeschlossener Projekte
- Zusammenfassung der Faktoren mit stärkstem Einfluß in einer Gleichung
 - ähnlich der Multiplikatormethode
- Ständige Überprüfung und Anpassung der Koeffizienten



Function Points

- **Kategorisierung jeder Anforderung**
Eingabedaten, Abfragen, Ausgabedaten, Datenbestände, Referenzdaten
- **Klassifizierung jeder Anforderung**
einfach, mittel, komplex
- **Ermittlung der Function Points**
Summe von Anzahl pro Kategorie und Klasse mal Gewichtungsfaktor
- **Bewertung von Einflußfaktoren**
Verflechtung, Dezentralität, Transaktionsrate, Verarbeitungslogik, Wiederverwendbarkeit, Konvertierungen, Anpaßbarkeit
- **Berechnung der bewerteten Function Points**
- **Ermittlung des Personalaufwands hierfür aus Kurve (mit Erfahrungswerten)**
- **Aktualisierung der Kurve nach Projektabschluß**

Fazit: Aufwandsabschätzung stets grob,
Erfolg ? von vielen Faktoren abhängig ...



Definitionsphase

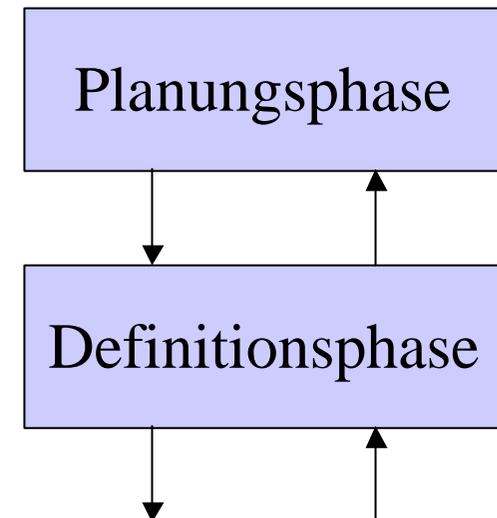
Mit dem Lastenheft (grobes Pflichtenheft) ist die Aufgabenstellung für eine Auftragsvergabe nicht genau genug umrissen.

Definitionsphase liefert Produktdefinition als genaue Beschreibung, Aufgabenbeschreibung („was“ aber nicht „wie“),

Produktdefinition beinhaltet (nach Balzert): Pflichtenheft, Produktmodell, Benutzeroberfläche, Erstfassung eines Benutzerhandbuchs.

Übersicht

- Systemanalyse
- Definitionsbereiche
- Komplexitätsarten
- Konzepte zur Produktbeschreibung
- Beschreibungsarten
- Erstellung eines Pflichtenhefts
- Gliederung für ein Pflichtenheft
- Vorgehen zur Erstellung der Produktdefinition





Systemanalyse

- ◆ Anforderungen (requirements)
 - Festlegung der Produkteigenschaften
 - qualitativ und quantitativ
 - aus Sicht des Auftraggebers
- ◆ Systemanalyse (requirements engineering)
 - Systematisches Vorgehen zur Ermittlung der Anforderungen
 - iterativer Prozeß
- ◆ Definition des Produkts
 - Anforderungen ermitteln
 - Anforderungen festlegen
 - Anforderungen analysieren
 - Anforderungen darstellen
 - Anforderungen verabschieden



Produktanforderungen

- Anforderungen ermitteln
 - Auftraggeber (extern / Marketing / Fachabteilung)
 - Benutzer (bzw. deren Repräsentanten)
 - ggf. Ist-Analyse eines vorhandenen Systems
- Anforderungen festlegen
 - Gruppieren (Funktionen / Daten / Leistungen / Benutzungsschnittstelle / Qualitätsanforderungen)
 - Klassifizieren (z.B. muß / soll / wünschenswert)
 - Nutzung von Beschreibungsmethoden (graphisch / textuell)
- Anforderungen analysieren
 - Inhaltliche Vollständigkeit
 - Konsistenz
 - Eindeutigkeit
 - Durchführbarkeit



Produktanforderungen (2)

- Anforderungen darstellen
 - Animation
 - Simulation
 - Ausführung
 - ⇒ Prototyping
- Anforderungen verabschieden
 - Vertragliche Festlegung
 - Produkt-Definition als Dokument(e)



Produktdefinition

- ggfs aus Teilproduktdefinitionen zusammengesetzt
- meist auch Vertragsbestandteil zwischen Auftraggeber & -nehmer
- Grundlage für den Software-Entwurf (in der Folgephase).

Was muß definiert werden?

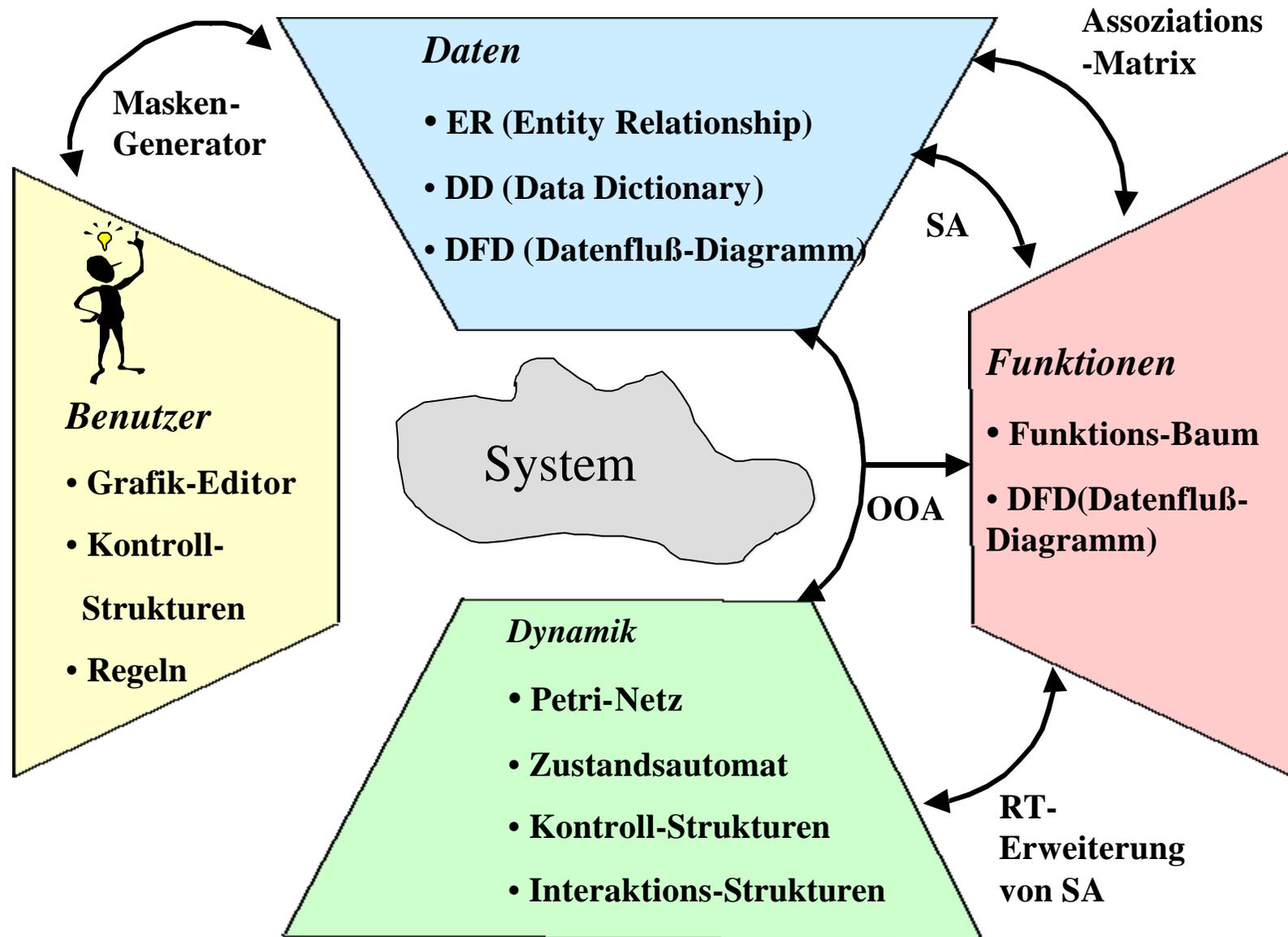
Vier verschiedene Sichten und ihre Zusammenhänge:

- Daten
- Funktionen
- Dynamik
- Benutzungsoberfläche

Es existieren hierzu eine Vielzahl an Formalismen & Notationen.



Sichten und ihre Konzepte





Begriffe zur vorherigen Folie



Benutzer-Oberfläche

ER = Entity Relationship

DD = Data Dictionary

DFD = Datenfluß-Diagramm

RT = Real Time Analysis

OOA = Object Oriented Analysis

SA = Structured Analysis

Bedeutung der einzelnen Sichten anwendungsabhängig,
gleichgewichtig oder ungleichgewichtig.

Structured Analysis / Real Time Analysis: traditioneller Ansatz

Objektorientierte Analyse: moderner Ansatz

(Stichwort: Unified Modeling Language UML)



Woher stammt die Komplexität in Software-Systemen ?

- ◆ Komplexität der Funktionen
 - z.B. Integrierte Büroprogramme mit > 1500 Funktionen
- ◆ Komplexität der Daten
 - z.B. Datenbanken: viele und komplexe Datenstrukturen
- ◆ Komplexität der Algorithmen
 - z.B. Planungssysteme: viele und komplexe numerische Verfahren
- ◆ Komplexität der Zeitabhängigkeit
 - z.B. Echtzeitsysteme: Nebenläufigkeit, Synchronisation, Antwortzeit
- ◆ Komplexität der Systemumgebung
 - z.B. Embedded Systems: Teilkomponenten, Schnittstellen
- ◆ Komplexität der Benutzungsoberfläche
 - z.B. CAD: komplexe graphische Interaktion

dadurch erklärt sich Vielzahl an Notationen, in denen sich jeweils einzelne Aspekte besonders gut darstellen lassen,

z.B. Datenmodellierung mit Entity Relationship Diagrammen



Erstellung eines Pflichtenhefts

- ◆ Aufgabe:
 - Zusammenfassung aller fachlichen Anforderungen des Produkts
- ◆ Adressaten:
 - Auftraggeber, Auftragnehmer, ggf. Benutzer
- ◆ Inhalt:
 - Fachlicher Funktions-, Daten-, Leistungs- und Qualitätsumfang
 - Vertragliche Beschreibung des “ Was” (nicht wie).
- ◆ Sprache:
 - detaillierte verbale Beschreibung, Nummerierung einzelner Anforderungen
- ◆ Aufbau:
 - übersichtliches formalisiertes Gliederungsschema
- ◆ Zeitpunkt:
 - 1. Dokument nach Abschluß der Planungsphase, iterativ



Gliederung für ein Pflichtenheft

Internationale Standardisierung (IEEE, 1984)

Standard *Guide for Software Requirements* von IEEE 1984 verabschiedet, von ANSI (American National Standard Institute) 1985 übernommen.

Hier vereinfachte Fassung nach Balzert

- ◆ Zielbestimmung
 - Mußkriterien, Wunschkriterien, Abgrenzungskriterien
- ◆ Produkteinsatz
 - Anwendungsbereich, Zielgruppen, Betriebsbedingungen
- ◆ Produktumgebung
 - Software, Hardware, Orgware, Schnittstellen



- Gliederung für ein Pflichtenheft (2)

- ◆ Produktfunktionen
 - Einzelne Abschnitte für Funktionen / Funktionsgruppen
 - Was, nicht wie - aus Benutzersicht
 - unabhängig von der Benutzungsoberfläche
 - Einzelanforderungen verbal, numeriert mit /F10/ etc.
 - Kennzeichnung von Wunschkriterien /F20W/
- ◆ Produktdaten
 - Beschreibung der langfristig zu speichernden Daten
 - aus Benutzersicht
 - nummeriert mit /D10/ etc.
- ◆ Produktleistungen
 - Alle zeitbezogenen oder umfangsbezogenen Leistungsanforderungen
 - nummeriert mit /L10/ etc.



- Gliederung für ein Pflichtenheft (3)

- ◆ Benutzungsoberfläche
 - Bildschirmlayout, Drucklayout, Tastaturbelegung, Dialogstruktur etc.
 - nur soweit produktspezifisch
 - numeriert mit /B10/ etc.
- ◆ Qualitätszielbestimmung
 - operationalisierte Qualitätsmerkmale
 - soweit nicht in allgemeinen Standards vorgegeben
- ◆ Testszenarien / Testfälle
- ◆ Entwicklungsumgebung
 - Software, Hardware, Orgware, Schnittstellen
 - insbesondere benötigte Werkzeuge
- ◆ Ergänzungen
 - z.B. Installationsbedingungen



Pflichtenheft-Beispiel (Teil 1)

1 Zielbestimmung

Das Programm HIWI-Verwaltung soll einen Lehrstuhl in die Lage versetzen, die beschäftigten Hilfsassistenten rechnergestützt zu verwalten.

1.1 Mußkriterien

- Verwalten von HIWI-Daten
- Erstellen einer Liste aller HiWis
- Ausgabe aller Daten eines HiWis

1.2 Wunschkriterien

- Ausdrucken von Arbeitszeugnissen
- Automatische Generierung einer Verlängerung der HIWI-Verträge um drei Monate zum Quartalsende
- Berechnung der benötigten drei Gehälter für alle HIWIs bis zum Jahresende
- Serienanschreiben zwecks Einladung zur Weihnachtsfeier
- Gezielte Abfragen mit einer Endbenutzersprache

1.3 Abgrenzungskriterien

- Software ist nicht netzwerkfähig
- keine Überprüfung der Arbeits- / Urlaubszeiten der HIWIs



Pflichtenheft-Beispiel (Teil 2)

2 Produkteinsatz

Das Produkt wird im Sekretariat eines Lehrstuhls zur Verwaltung der HiWis eingesetzt

2.1 Anwendungsbereich

- HiWi-Verwaltung; Abfragen, (kommerzieller Anwendungsbereich)

2.2 Zielgruppen

- Mitarbeiter/-in eines Lehrstuhls (Sekretär/in)

2.3 Betriebsbedingungen

- Büroumgebung



Pflichtenheft-Beispiel (Teil 3)

3 Produktumgebung

Das Produkt läuft auf einem Arbeitsplatzrechner mit graphischer Benutzungsoberfläche

3.1 Software

- Betriebssystem: Windows XP

3.2 Hardware

- PC

3.3 Orgware

- Netzwerkverbindung zum Drucker

3.4 Produktschnittstellen

- Für die Verlängerungen der Arbeitsverträge wird eine Word-Serienbrief-Datei erzeugt.



Pflichtenheft-Beispiel (Teil 4)

4 Produktfunktionen

- /F10/ Ersterfassung, Änderung und Löschung von HiWis /LF10/
- /F20/ Aufnahme eines HiWis mit Überprüfung, ob er/sie mehr als 19 h/Woche an der Universität als HIWI beschäftigt ist.
(HIWIs dürfen höchstens 19 h/Woche an der Hochschule arbeiten)
- /FW30/ Ausdrucken eines Arbeitszeugnisses mit Überprüfung, ob
 1. die Beschäftigungszeit beendet ist
 2. noch kein Zeugnis gedruckt worden ist
- /F40/ Ausgabe (Drucker + Bildschirm) einer alphabetischen Liste aller HiWis mit folgenden Daten: Name, Vorname, Matrikel-Nr., Studienadresse, Geburtsdatum /LF20/
- /F50/ Je HIWI: Ausgabe sämtlicher Daten /LF30/
- /FW60/ Individuelle, gezielte Abfragen mittels Menüs und Maus /LF40/
- /FW70/ Suchoperationen auf allen Datenfeldern /LF40/
- /FW80/ Zum Quartalsende werden alle auslaufenden HIWI-Verträge automatisch um 3 Monate verlängert und ausgedruckt
- /FW90/ Zum aktuellen Zeitpunkt wird die Summe der bis zum Jahresende benötigten Gehälter für alle HIWIs berechnet
- /FW100/ Das Programm generiert Steuerdateien mit den Adressen aller HiWis, um mit Word Serienbriefe an alle HIWIs zu schicken



Pflichtenheft-Beispiel (Teil 5)

5 Produktdaten

/D10/ Personendaten:

Name, Vorname, Matrikel-Nummer, Geburtsdatum, Heimatadresse, Heimat-Telefonnummer, Studienadresse, Studien-Telefonnummer

/D20/ Studiendaten (können pro Student mehrere sein):

Studienfach, Semesteranzahl, Vordiplomnote + Datum, Diplomnote + Datum

/D30/ Beschäftigungsdaten (können pro Student mehrere sein):

Beginndatum, Endedatum, Stundenzahl pro Woche, Tätigkeitsgebiete, benutzte Software-Systeme, Arbeitszeugnisdatum

6 Produktleistungen

/L10/ Die Funktionen /F10/, /F20/, /F30/, /F40/, /F50/, /F60/, /F70/ und /F90/ dürfen nicht mehr als 2 Sekunden Antwortzeit benötigen /LL10/

/L20/ Die Funktionen /F80/ und /F100/ dürfen nicht mehr als 5 Minuten benötigen (incl. Drucken). Der Arbeitsfortschritt wird auf dem Bildschirm angezeigt.

/L30/ Es müssen maximal 1.000 Hilfsassistenten verwaltet werden können. /LL20/.



Pflichtenheft-Beispiel (Teil 7)

- 7 Benutzungsschnittstelle
- /B10/ Standardmäßig ist eine menüorientierte Bedienung vorzusehen.
- /B20/ Die Bedienungs Oberfläche ist auf Mausbedienung auszulegen; eine Bedienung ohne Maus muß aber auch möglich sein.
- /B30/ ISO 9241 ist zu beachten.
- /B40/ Sämtliche Daten sind paßwortgeschützt und dürfen nur von autorisierten Mitarbeitern des Lehrstuhls bearbeitet werden.



Pflichtenheft-Beispiel (Teil 8)

6 Qualitätszielbestimmung

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität				
Angemessenheit		X		
Richtigkeit		X		
Interoperabilität				X
Ordnungsmäßigkeit		X		
Sicherheit		X		
Zuverlässigkeit				
Reife			X	
Fehlertoleranz			X	
Wiederherstellbarkeit		X		
Benutzbarkeit				
Verständlichkeit		X		
Erlernbarkeit		X		
Bedienbarkeit	X			
Effizienz				
Zeitverhalten			X	
Verbrauchsverhalten			X	
Änderbarkeit				
Analysierbarkeit			X	
Modifizierbarkeit			X	
Stabilität		X		
Prüfbarkeit			X	
Übertragbarkeit				
Anpaßbarkeit				X
Installierbarkeit				X
Konformität				X
Austauschbarkeit				X



Pflichtenheft-Beispiel (Teil 9)

9 Globale Testfälle

Folgende Funktionssequenzen sind zu überprüfen:

/T10/ HIWI-Ersterfassung, Änderung, Weiterbeschäftigung, Vertragsende, Arbeitszeugnis drucken und Löschen

/T20/ Serienanschreiben drucken

/T30/ Summe der Gehälter bis zum Jahresende bestimmen, HIWI-Vertrag unterbrechen, neue Summe der Gehälter bis zum Jahresende bestimmen

Folgende Datenkonsistenzen sind einzuhalten:

/T40/ Ein Beschäftigungszeitraum und die Studiendaten können nur eingegeben werden, wenn ein entsprechender HIWI-Eintrag (Personendaten) vorhanden ist.

10 Entwicklungsumgebung

Offen



Erstellung der Produktdefinition (nach Balzert)

- ◆ (iterative) Erstellung des Pflichtenhefts
- ◆ (iterative) Erstellung eines Produktmodells
 - ◆ formalisiert und verfeinert Beschreibung im Pflichtenheft
 - ◆ setzt Auswahl der Beschreibungskonzepte voraus
 - ◆ Konzepte sind reichhaltig, bisher noch nicht betrachtet,
 - ◆ im folgenden Entity-Relationship Diagramme als Beispiel herausgegriffen
 - ◆ Unified Modeling Language (UML) als Menge von festgelegten Notationen insb. für die Objektorientierte Analyse und den Objektorientierten Entwurf
- ◆ (iterative) Konzeption der Benutzungsoberfläche
 - ◆ als Prototyp oder Beschreibung
- ◆ (iterative) Erstellung eines ersten Benutzungshandbuchs
 - ◆ Motivation: Sichtweise des Anwenders sollte frühzeitig mitbedacht werden

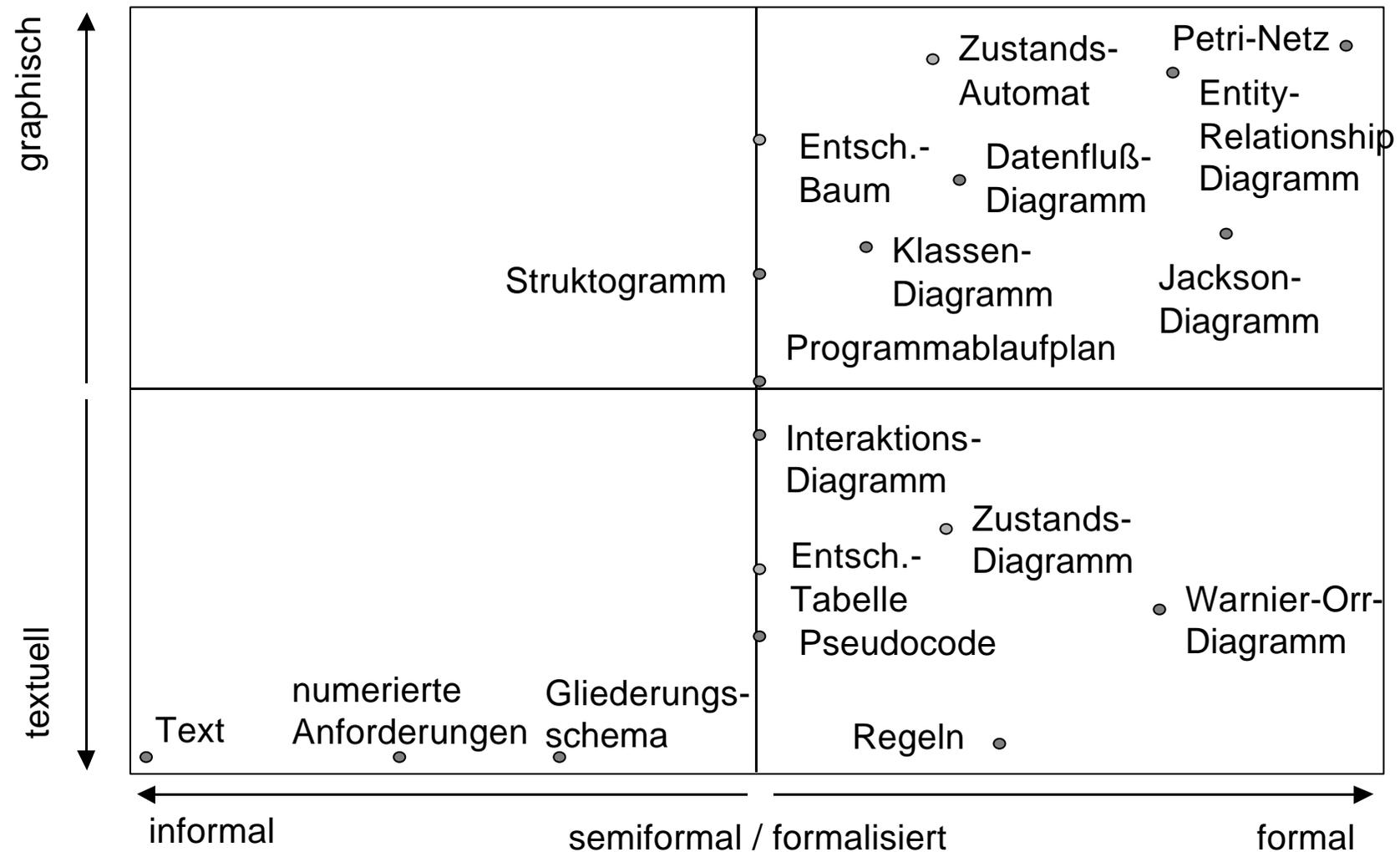


Abschlußübersicht

- Planungsphase:
 - Lastenheft, Projektplan und Projektkalkulation
- Definitionsphase:
 - Produktdefinition: Pflichtenheft, Produktmodell, Benutzungsoberfläche und Benutzungshandbuch
 - legt fest, was erstellt werden soll
- Entwurfsphase:
 - Software-Architektur, Spezifikation von Systemkomponenten
 - legt fest, wie System gestaltet wird
- Implementierungsphase
 - Programmcode
 - füllt realisiert Systemkomponenten
 - schließt Testphase mit ein
- Installation & Wartungsphase



Übersicht zu Beschreibungsarten





Konzepte zur Produktbeschreibung

F=Funktion, D=Daten, A=Algorithmen, Z=Zeitabhängig, S=Systemumgebung,
 B=Benutzungsoberfläche D=Definition, E=Entwurf, I=Implementierung
 SA=Structured Analysis, RT=RealtimeAnalysis, OOA= Objectoriented Analysis

Konzept	Komplexität						Phase			Methode		
	F.	D.	A.	Z.	S.	B.	D.	E.	I.	SA	RT	O A
Funktionsbaum	+						x	x		o	o	
Datenflußdiagr.	+				+		x			x	x	
Jackson-Design		+	+				x	x	x			
Syntax-Diagr.							x	x				
Data Dictionary		+					x	x	x	x	x	
Entity-Relationsh.		+					x				o	x
Klassen-Diagr.	+	+					x	x	x			x
Warnier-Orr-Dia.			+						x			
Struktogramme			+						x			
Progr.-Ablaufplan			+						x			
Pseudocode			+				x		x	x	x	o
Entscheidungs-T. Regeln			+			+	x		x	x	x	
Zustandsautomat		o		+		o	x	x			x	x
Petri-Netz				+		o	x					
Interaktionsdiagr.				+	+		x	x				x