



Management großer Softwareprojekte

Prof. Dr. Holger Schlingloff

Humboldt-Universität zu Berlin,
Institut für Informatik

Fraunhofer Institut für Rechnerarchitektur
und Softwaretechnik FIRST

Hausaufgabe von letzter Woche

- Lesen Sie Kapitel 1 („The tar pit“) aus dem Buch von F. Brooks und fassen Sie drei Hauptthesen des Kapitels in je einem Satz zusammen. Stimmen Sie nach Ihrer Erfahrung mit diesen Thesen überein?
- Erstellen Sie einen Ausschreibungstext für ein System zur Verwaltung der Internetseiten des Instituts (Design, Aktualisierungsmöglichkeit, Informationsdienste, Vorlesung, Privates, ...)



Hausaufgabe für nächste Woche

- Lesen Sie Kapitel 5 („The second system effect“) und Kapitel 11 („Plan to throw one away“) aus dem Buch von F. Brooks. Widersprechen sich die Aussagen der beiden Kapitel gegenseitig?
- Erstellen Sie das Konzept eines Angebots für ihre Ausschreibung des CMS von letzter Woche (oder, ersatzweise, das im Dokument angegebene). Annahmen: KMU (~80 MA) mit einschlägigen Erfahrungen, 90% Auslastung.



Wo stehen wir?

2. Projektphasen

2.1: Produktzyklus

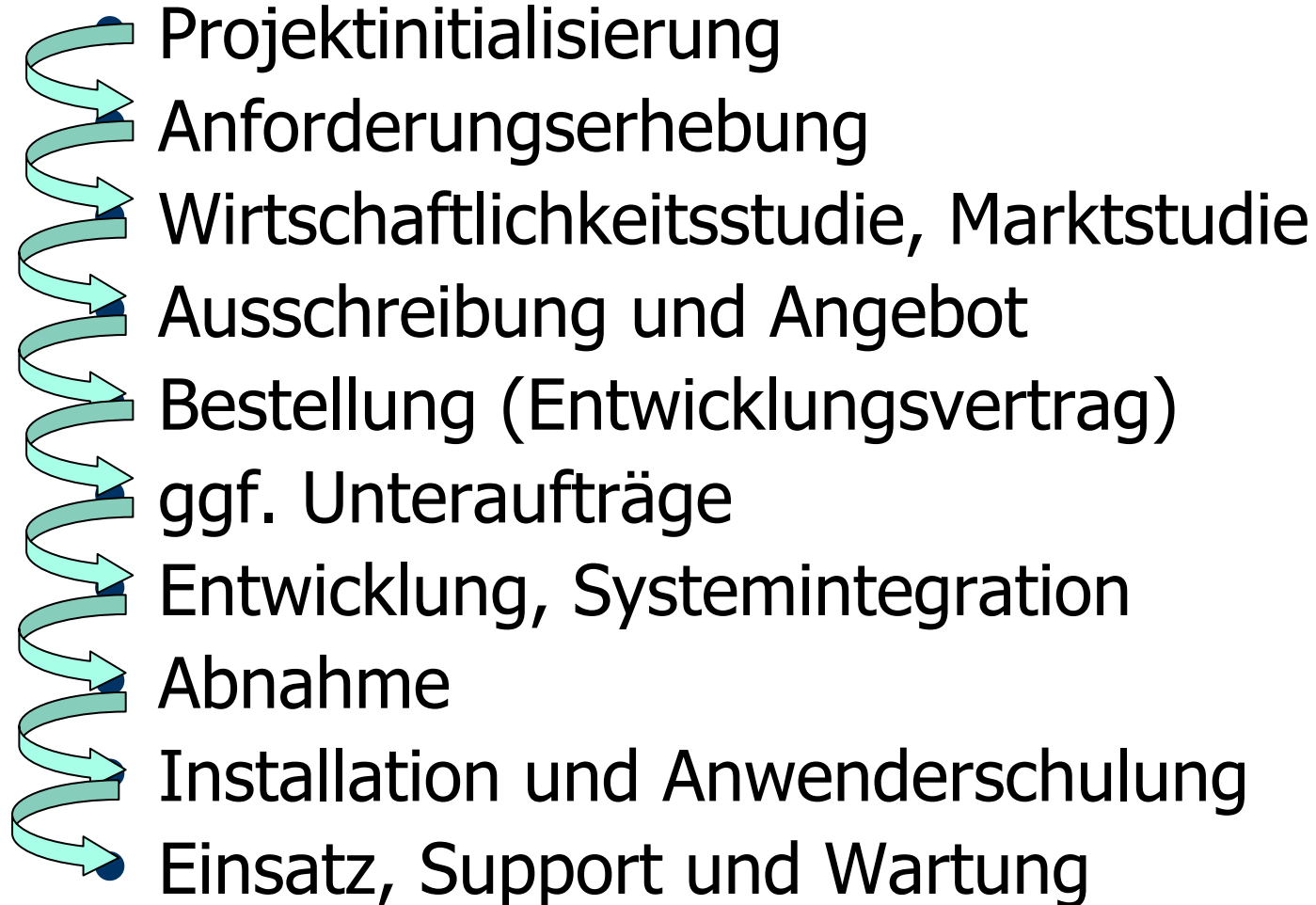
von der Projektidee bis zum Entwicklungsbeginn

- **heute:** von der Produktentwicklung bis zur Außerbetriebsetzung

2.2: Produktentwicklungszyklus



2.1 Produktzyklus



Abnahme

- Validierung, ob das Produkt den im Pflichtenheft geforderten Anforderungen genügt
- formaler Prozess, mit festgelegtem Termin, Beteiligten und Agenda
- bei großen Projekten meist in mehreren Teilen
(auch: Entscheidung über vorzeitigen Abbruch!)
- Nachbesserungen oft strittig (Abnahmekriterien!)



Installation, Schulung

- Migration oft ein wesentliches Arbeitspaket
- Parallelbetrieb, stufenweise Übernahme oder „big bang Verfahren“
- „Plan B“ für Katastrophen machen!
- Schulung oft nicht im Projektvolumen, aber für den Gesamterfolg kritisch
- auch in Forschungsprojekten wird oftmals „Nachhaltigkeit“ gefordert

➔ stille Ressourcen einplanen! (Helpfiles etc.)



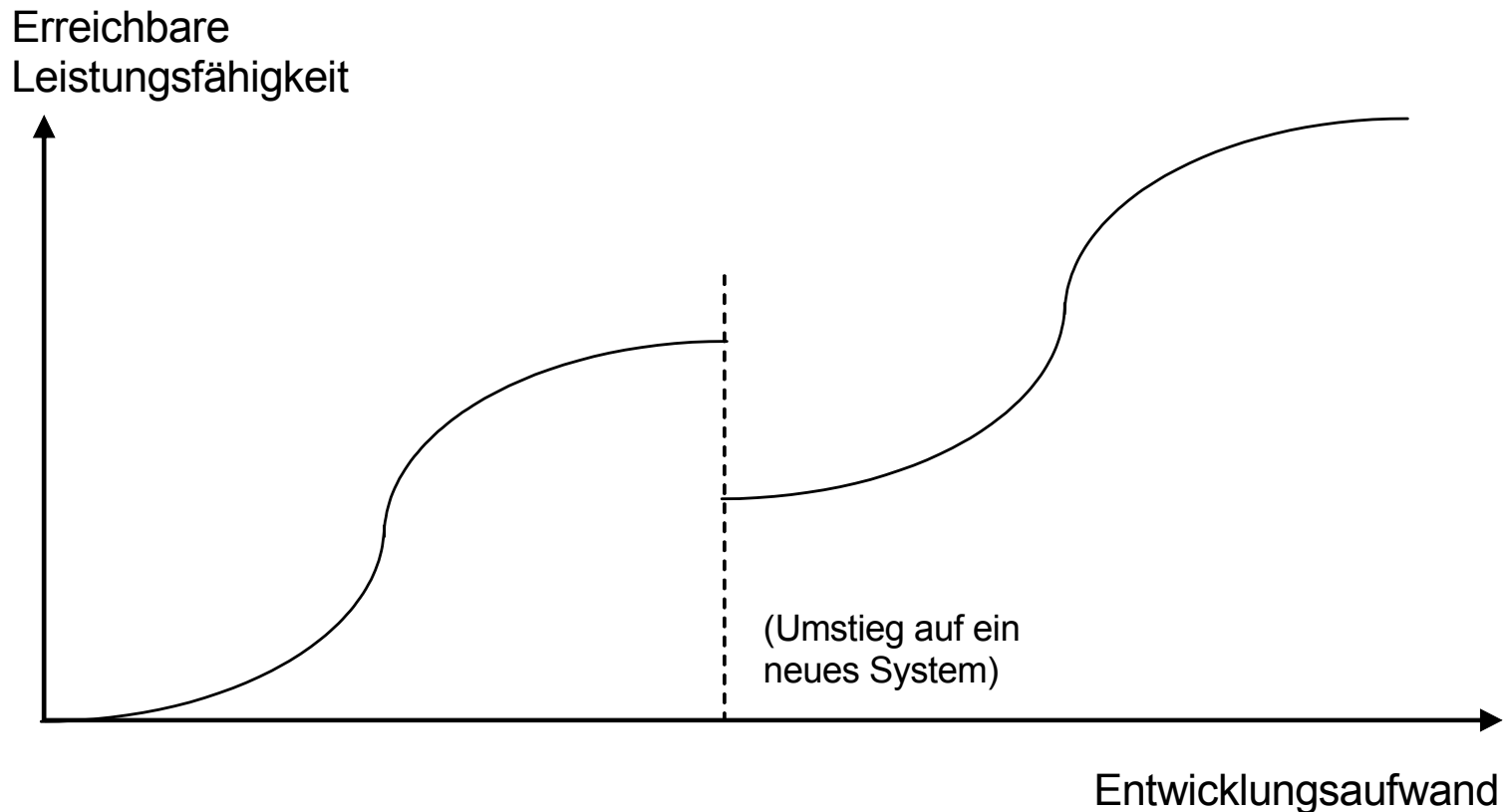
- kritische Beurteilung des Geschaffenen
 - aus Sicht des Auftraggebers
 - bringt das System den erwarteten Mehrwert?
 - welche Umstellungsschwierigkeiten traten auf?
 - aus Sicht des Auftragnehmers
 - war die ursprüngliche Kalkulation stimmig?
Falls nein, worin lagen die Ursachen?
 - welche Probleme traten im Projektverlauf ein?
 - was waren Gründe für Abweichungen vom Plan?
 - welche Teile liefen besonders gut?

Support und Wartung

- Support wird meist separat abgerechnet
eigentlich nicht mehr Teil des Projektes
- „einen Schritte vorwärts, zwei zurück“
 - jeder beseitigte Fehler zieht weitere nach sich
 - ursprüngliche Entwickler nicht mehr verfügbar
- wichtigstes Instrument:
Änderungsmanagement
 - Genehmigungsverfahren mit Kosten/Nutzen-Rechnung und festen Zuständigkeiten
 - Versionskontrolle, „service packs“



- Software-Lebensdauer z.T. > 20 Jahre!



2.2 Software-Entwicklungszyklus

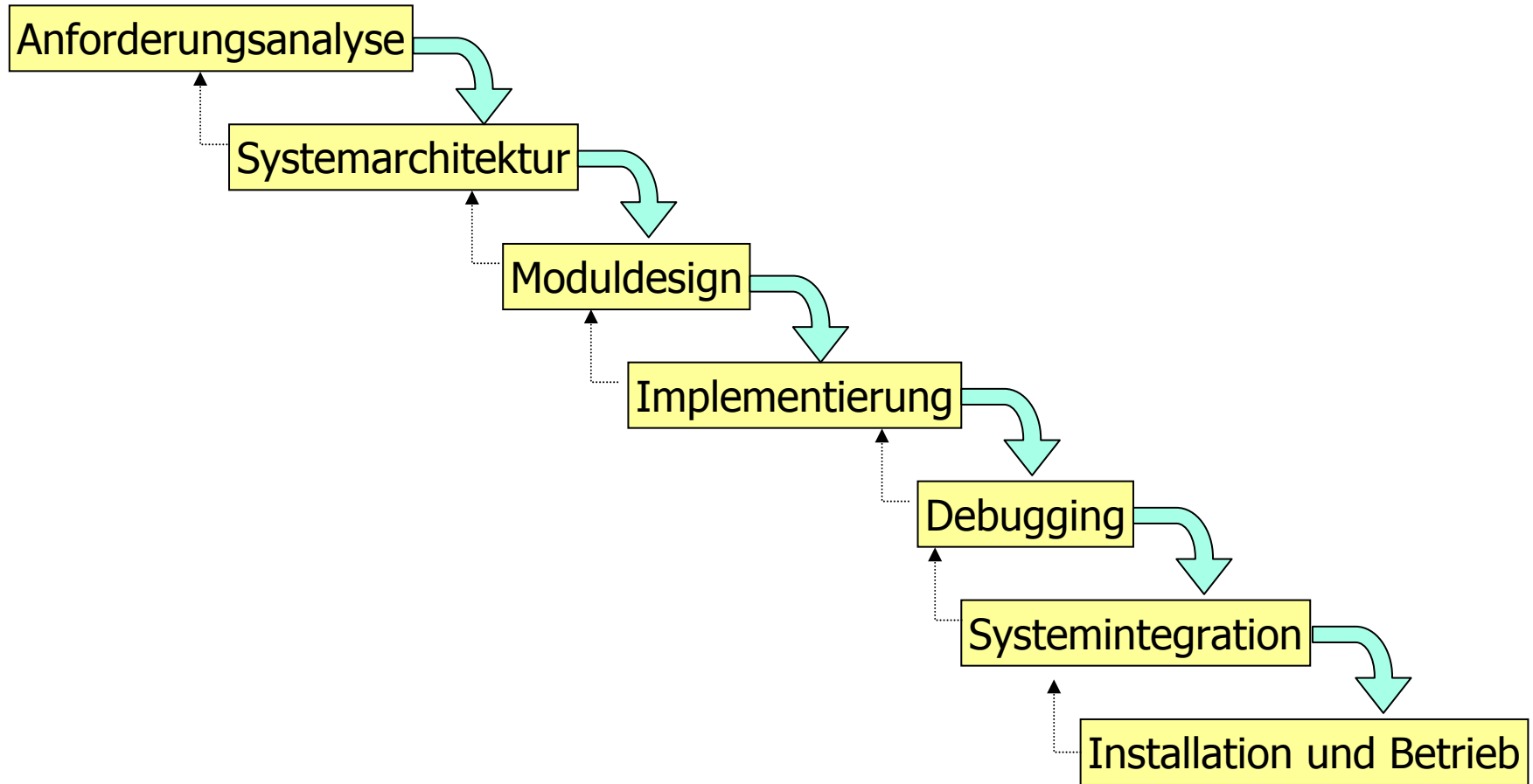
verschiedene Phasenmodelle (vgl. SW-Engineering), kein „einzig wahres“

Idee der Modelle: Vorgehensweisen
standardisieren („Checklisten“)

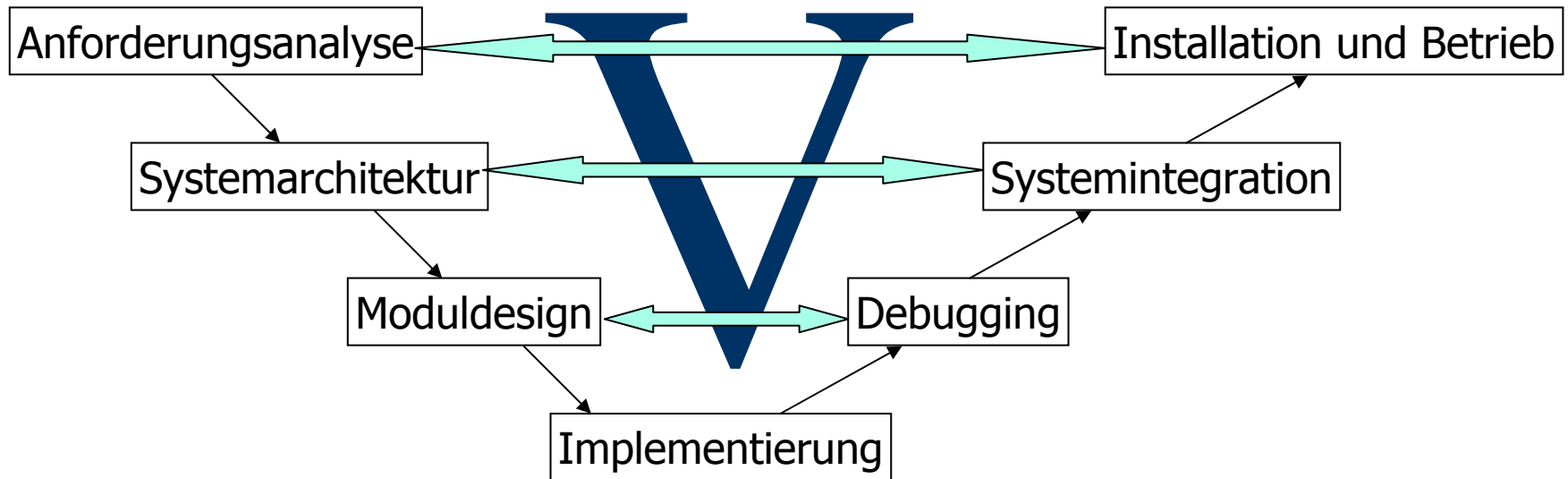
projektspezifische Anpassungen („Tayloring“)



Wasserfallmodell

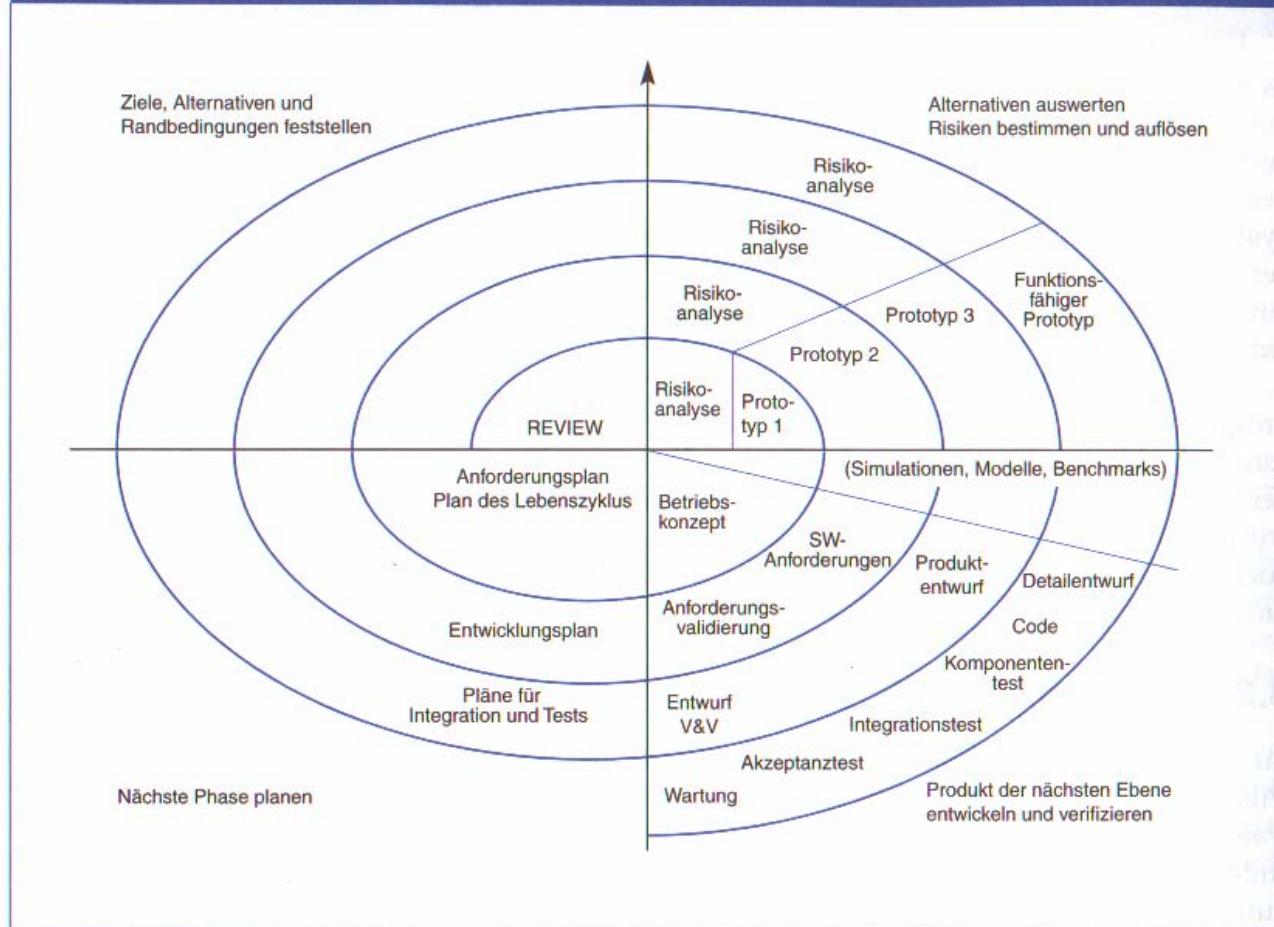


V-Modell



Spiralmodell

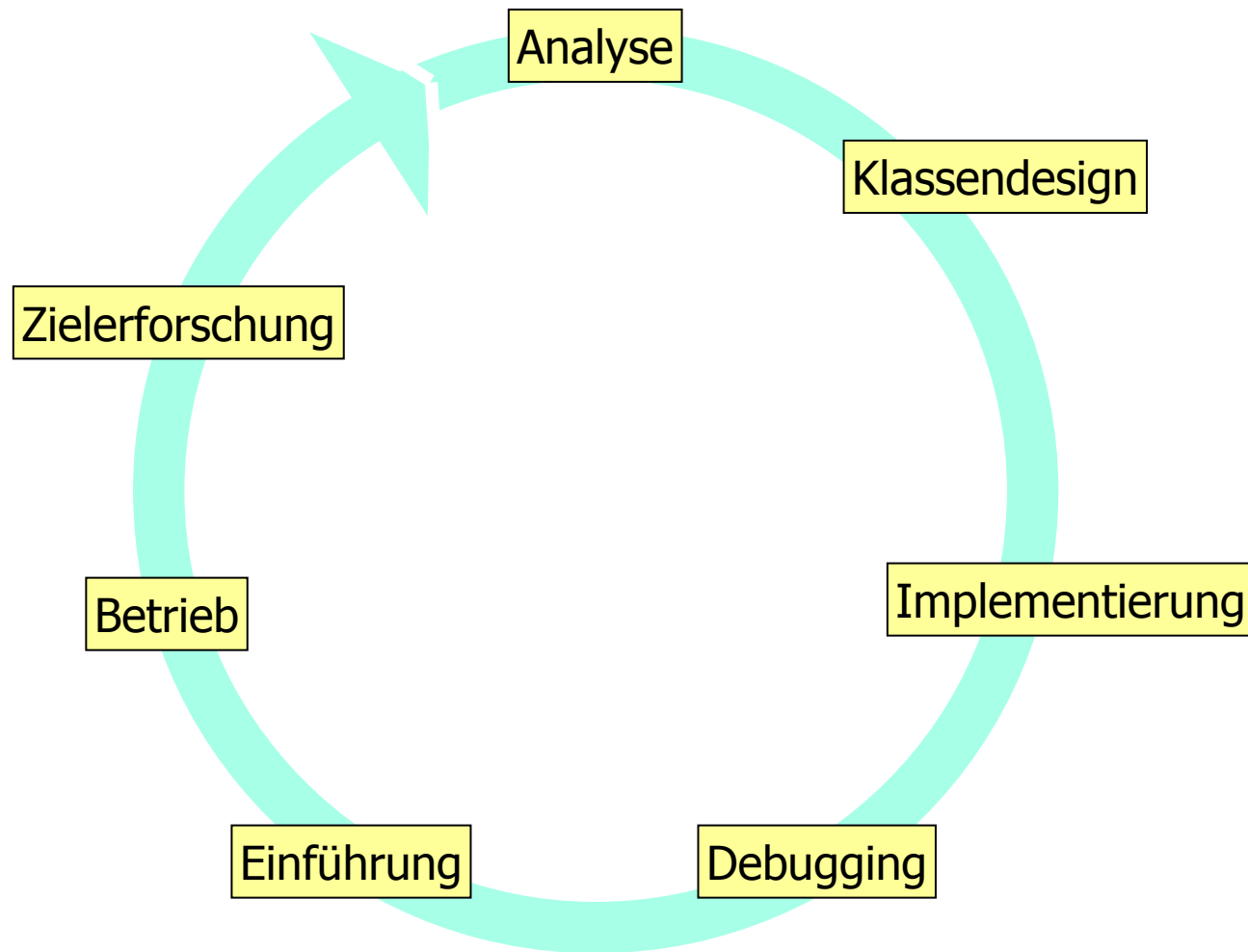
Abbildung 3.7 Das Boehm'sche Spiralmodell des Softwareprozesses
(© 1988 IEEE)



Bilduelle: Balzert,
Lehrbuch der
Softwaretechnik



Objektorientiertes Modell



Anforderungsanalyse

- Input: Angebots- oder Ausschreibungstext
- Output: Systemspezifikation
genaue Festlegung der Funktionalität
modular, wartbar, realisierbar, gekapselt

„use cases“, „message sequence charts“, „statemachines“, ...

Lastenheft: formale Definition der *Benutzeranforderungen*

Pflichtenheft: formale Definition der *Systemanforderungen*



Bsp. (Sommerville)

Bildquelle: Sommerville

Definition einer Benutzeranforderung (Im Lastenheft)

1. Die Software muss über Mittel zur Darstellung externer, von anderen Werkzeugen erzeugter Dateien verfügen und auf sie zugreifen können.

Spezifikation der Systemanforderungen (im Pflichtenheft)

- 1.1 Der Benutzer sollte über Möglichkeiten zur Definition externer Dateitypen verfügen.
- 1.2 Jeder externe Dateityp kann eine damit verknüpfte Anwendung besitzen, mit der die Datei bearbeitet wird.
- 1.3 Jeder externe Dateityp kann als bestimmtes Symbol auf dem Bildschirm des Anwenders dargestellt werden.
- 1.4 Es sollten Möglichkeiten zur Definition des Symbols für externe Dateitypen durch den Benutzer bereitgestellt werden.
- 1.5 Wenn ein Benutzer ein Symbol auswählt, das eine externe Datei repräsentiert, so soll die durch dieses Symbol dargestellte Datei mit der Anwendung geöffnet werden, die mit dem entsprechenden externen Dateityp verknüpft ist.



Probleme der Anforderungsanalyse

- fehlendes Domänenwissen
 - Spezialisten hinzuziehen!
- mangelnde Kommunikation
 - Auftraggeber, Projektleiter, Entwickler beteiligen!
- Änderungen
 - Modularisierung, Querverweise, Änderungsmanagement
- Mehrdeutigkeiten
 - strikte Formalisierung



Systementwurf

- Input: Anforderungsspezifikation
- Output: Systemarchitektur
 - strukturelle Sicht auf das System
 - Struktur + Interaktion, Schnittstellen,
Datenhaltung
 - System- und Komponentenmodell

Systemmodell: Zergliederung des Systems

Komponentenmodell: Funktionalität der Komponenten



Probleme in der Entwurfsphase

- Fehlentscheidungen haben späte Folgen
- Leistungsfähigkeit vs. Wartbarkeit
 - Subsystem- und Moduldekomposition
- Sicherheitskriterien mit berücksichtigen!
 - Schichtenmodell gegen Angriffe bevorzugt
 - Client-Server für Skalierbarkeit
- Integrität der Modelle
 - Objekt-, Datenfluss-, Steuerungsmodell

