Systembau (Konstruktion)

- **Ziel**
  - Modell eines automatischen Teilsystems optimal in einen Softwareentwurf umsetzen
- **Input (aus Detailstudie)**
  - Technologisches Soll-Modell (Mensch-Maschine-Abgrenzung) => ERM
  - Vorgaben aus Rahmenorganisation
- **Vorgehen, Realisierungsphasen**
  - Relationale Datenbank entwickeln
  - Software-Architektur entwickeln (Modul-, Programm- und Schnittstellen-struktur erarbeiten)
  - Testplan erstellen
- **Resultate**
  - Betriebsbereite DB für Anwendungsfertigung
  - Software-Architektur
  - Testplan
  - Pläne: Fertigung, Benutzerausbildung
- **Methodische Grundlagen**
  - Systems Engineering
  - Strukturierte Designtechnik
  - Datenbank-Benchmarking und -Tuning
- **Programmiersprachen Einsatzbeispiele**
  - Kaufmännische Anwendungen
    - Cobol, Basic, Visual...
  - Technische Anwendungen
    - Java, C, Smalltalk
  - Kaufmännisch-Technische Anwendungen
    - Java, C, Smalltalk
  - Auswahlkriterien
    - Logische Kriterien
      - Anwendungsgebiet
      - Projektgrösse
      - Datenstrukturen (z.B. neue Datentypen definieren)
      - Effizienz des Compilers
    - Pragmatische Kriterien
      - Anforderungen durch Auftraggeber
      - Standardisierung der Sprache
      - Ausbildungsstand der Programmierer
      - Verfügbarkeit von Werkzeugen
- **Software-Architektur entwickeln**
  - Anwendungen aus Geschäftsvorgängen ableiten
  - Anwendungen transaktionsorientiert zerlegen
  - Dialoganwendungen entwicklen
  - Batchanwendungen entwicklen
  - Reportanwendungen entwicklen
- **Relationale Datenbank entwickeln**
  - Konzeptionelles Schema
    - Globales Datenmodell in Relationenmodell transformieren
    - Ansi-Sparc
    - Benutzersichten (externes Schema)
    - Physische Speicherung (internes Schema)
  - Speicherorganisation der DB spezifizieren (Tuning)
    - Leistungsverhalten der DB verbessern
    - Zugriffshäufigkeit analysieren
    - Datenvolumen berücksichtigen
    - Zugriffspfade berücksichtigen
    - Berücksichtigen der Eigenschaften des gewählten RDBMS
  - Konzeptionelles Schema denormalisieren (Tuning)
    - Kontrollierte Datenredundanz um Leistung zu steigern
    - Attribute mehrfach speichern (kont. Redundanz)
    - Relationen zusammenfassen (Performancesteigerung)