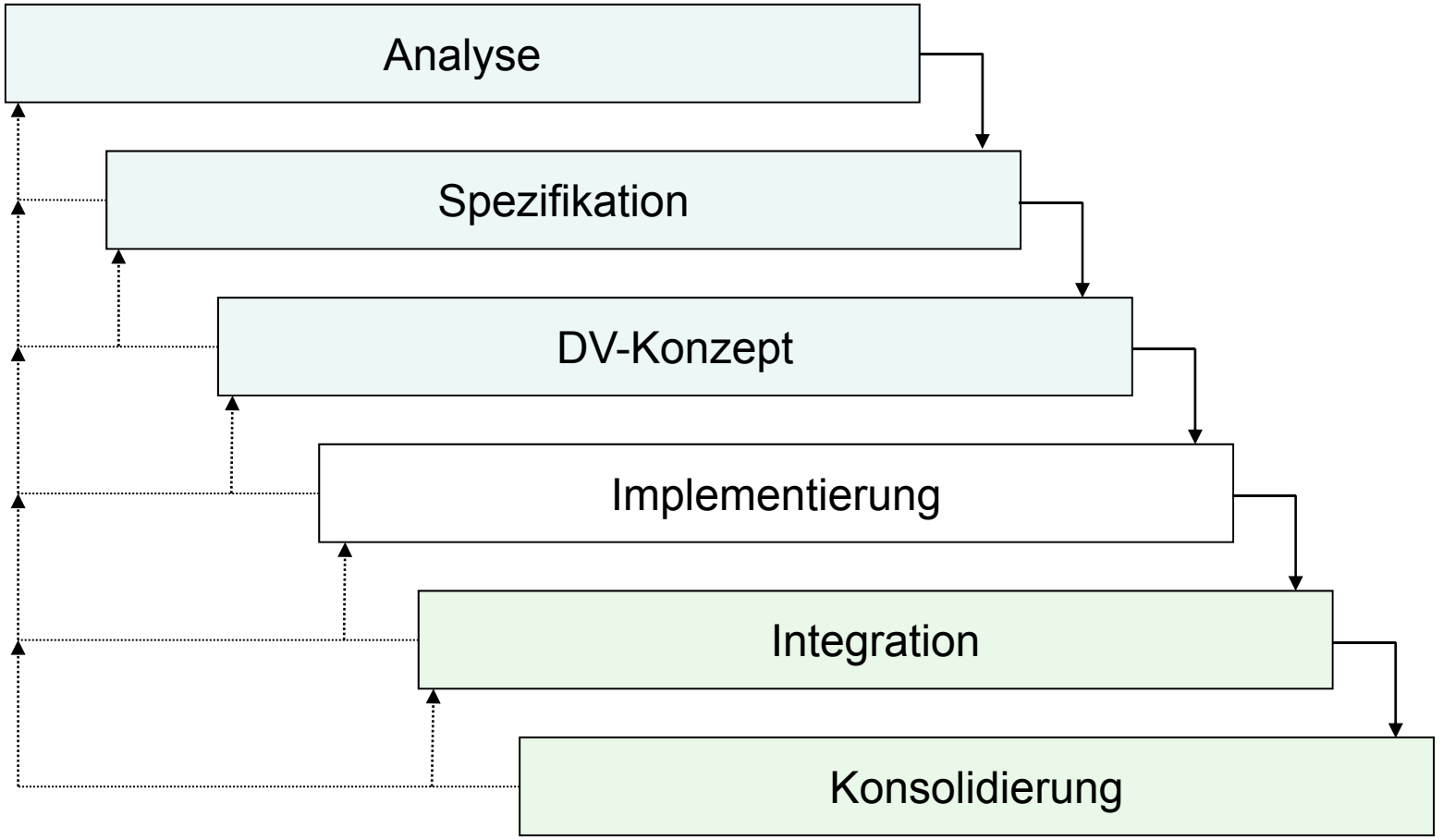
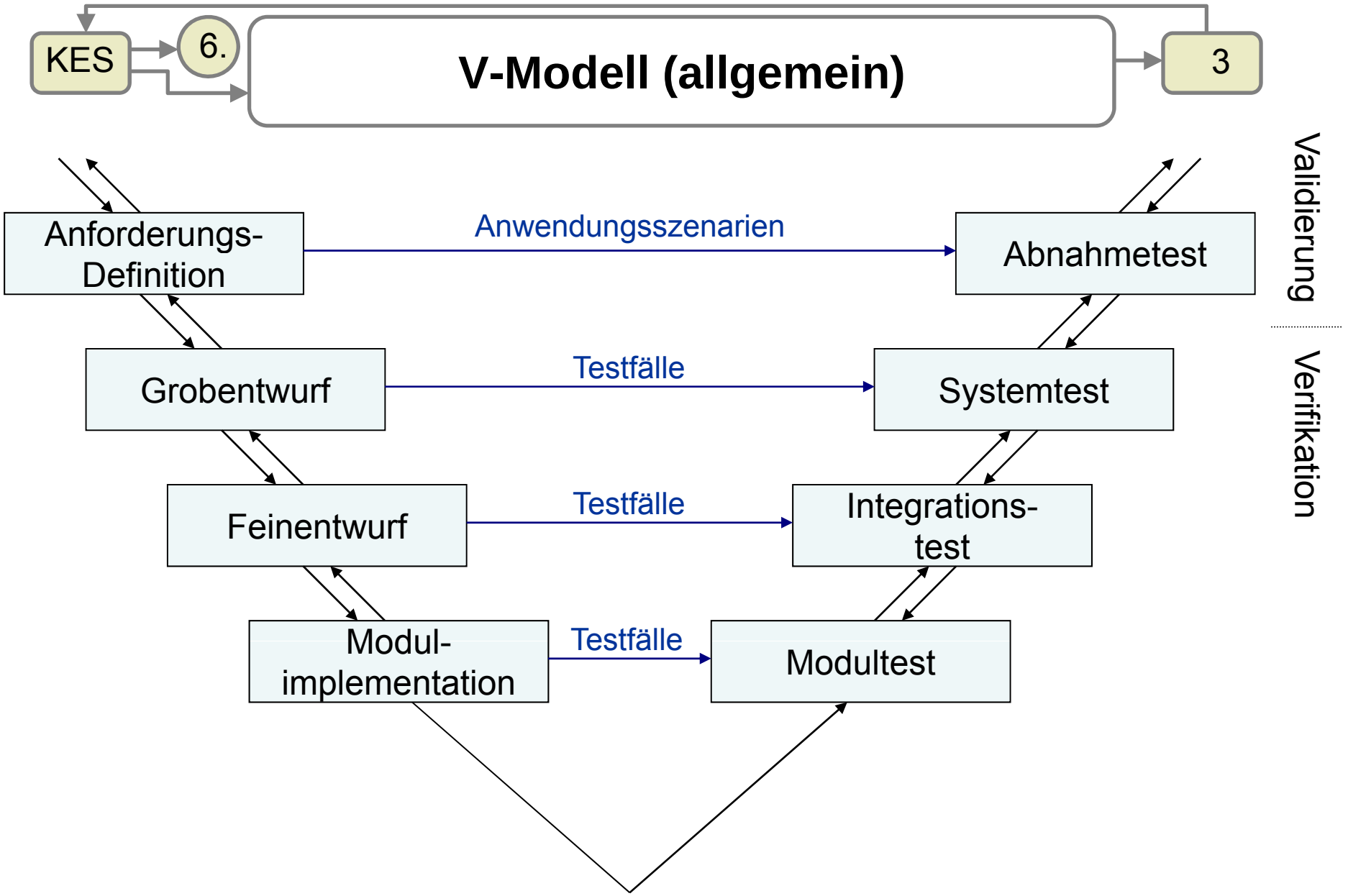
1. Vorgehensmodelle
2. Echtzeit Programmierung
3. Testen
    1. Allgemeines
    2. Remote Debugging
    3. In Circuit Emulation
    4. Hardware in the loop / Software in the loop
    5. JTAG

Wasserfallmodell

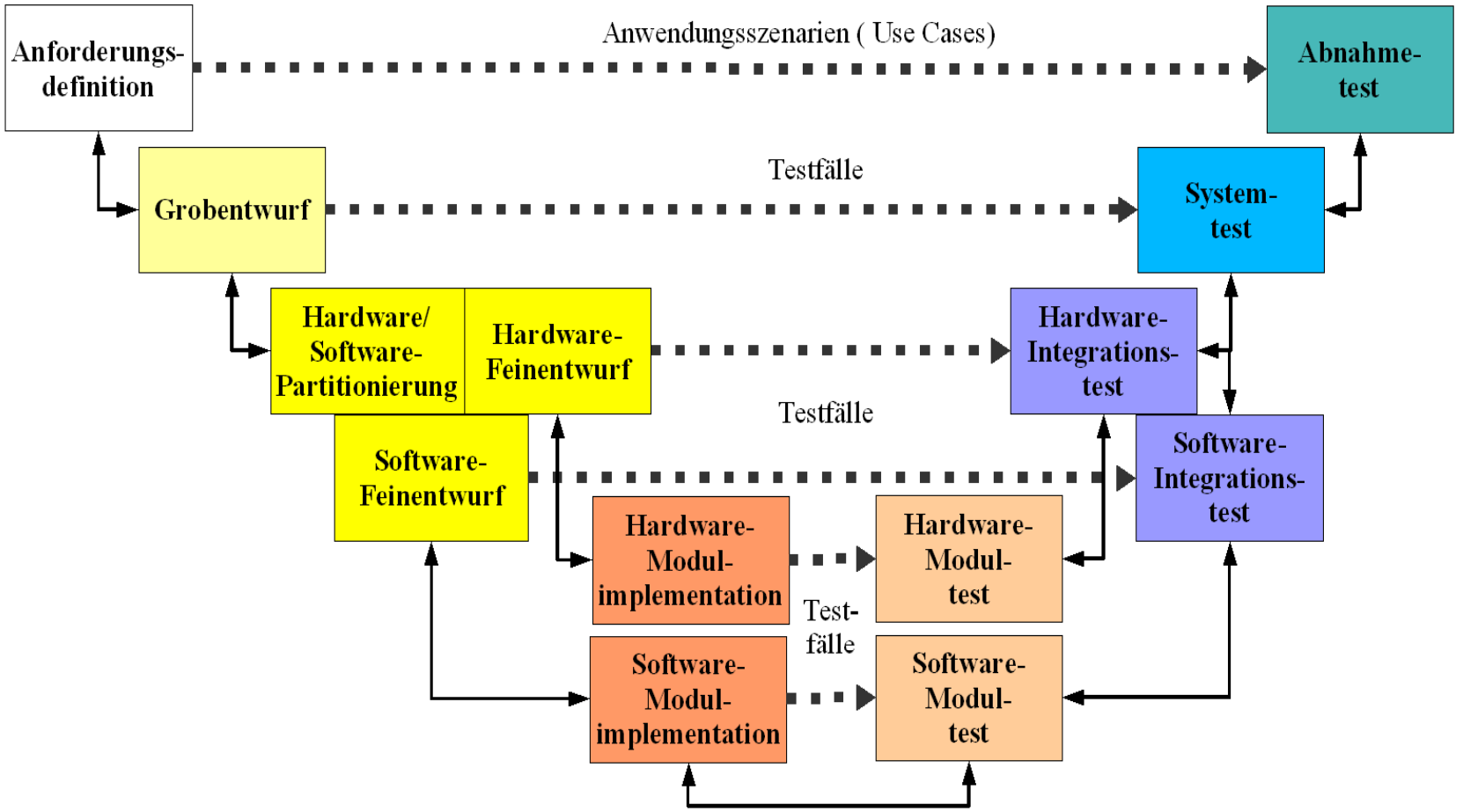
2





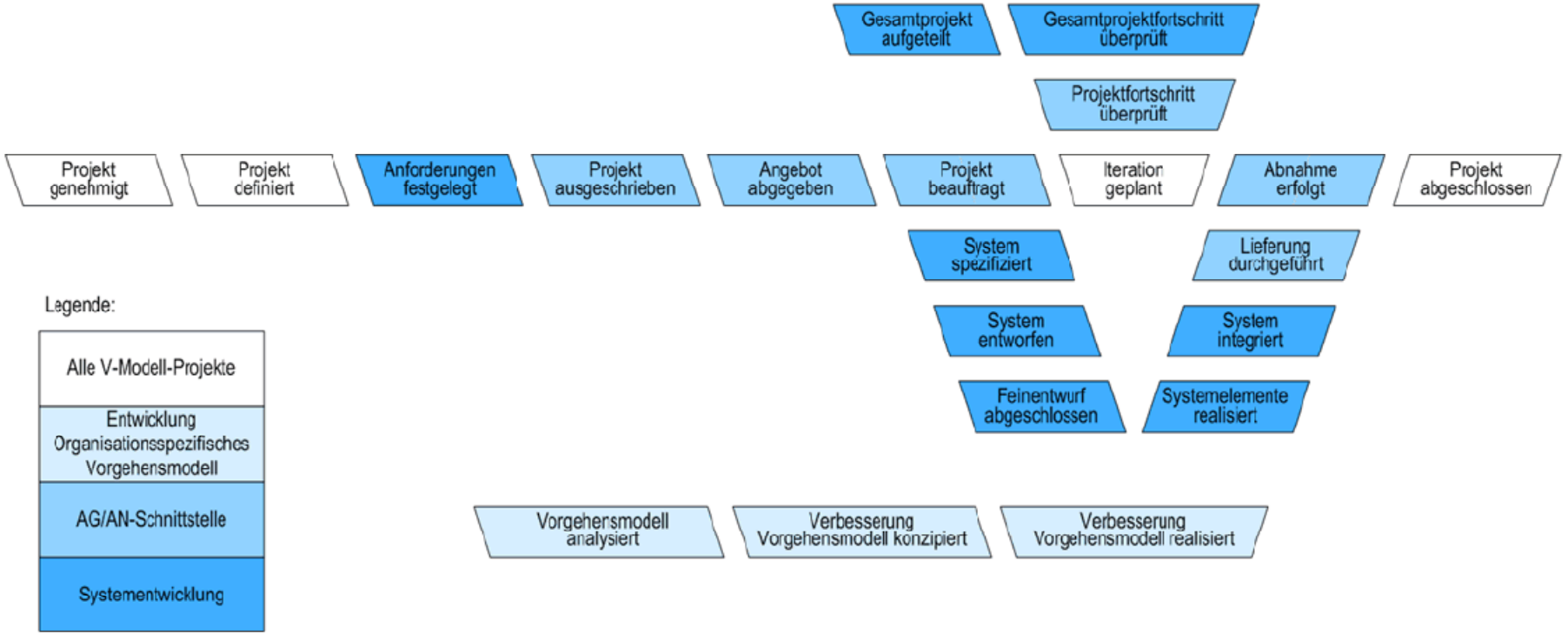
Quelle: Balzert, H: Lehrbuch der Softwaretechnik

V-Modell (Erweitert für eingebettete Systeme)



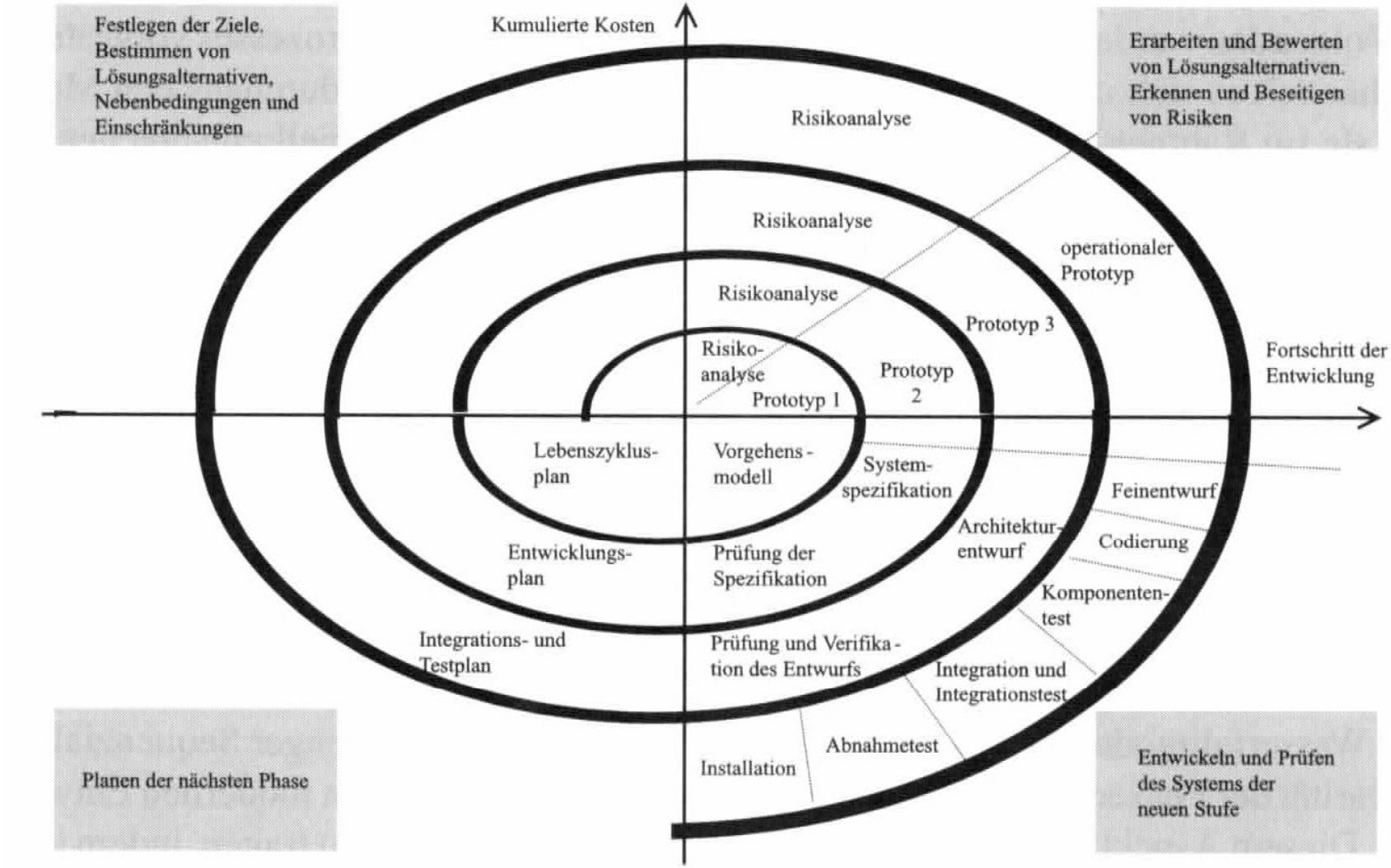
Quelle: Müller, M.; Implementation eines UML Entwurfs; Studienarbeit; TU Ilmenau, 2005

V-Modell XT



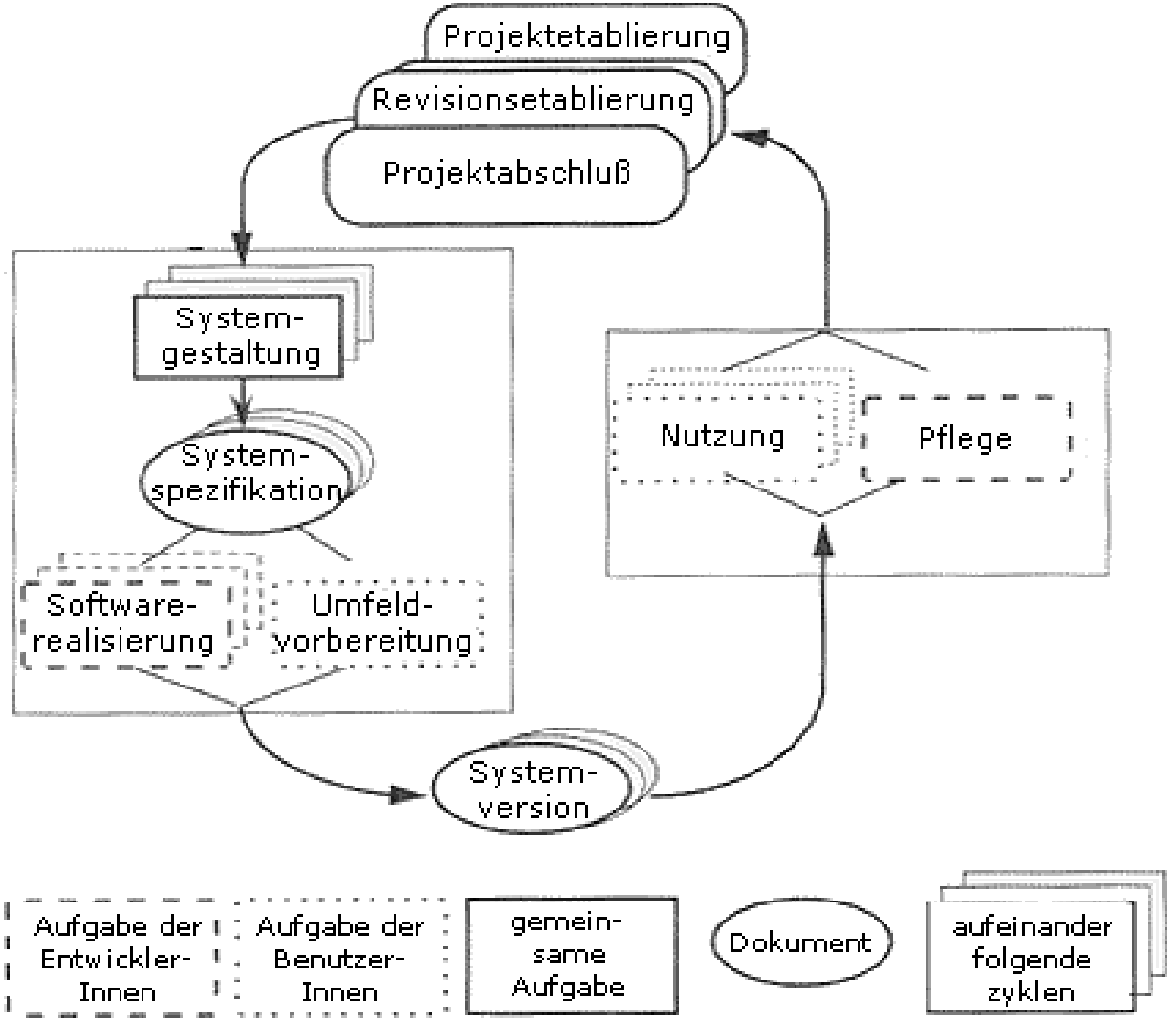
Quelle: Dokumentation V-Modell XT, 2004, http://www.v-modell-xt.de/

Spiralmodell



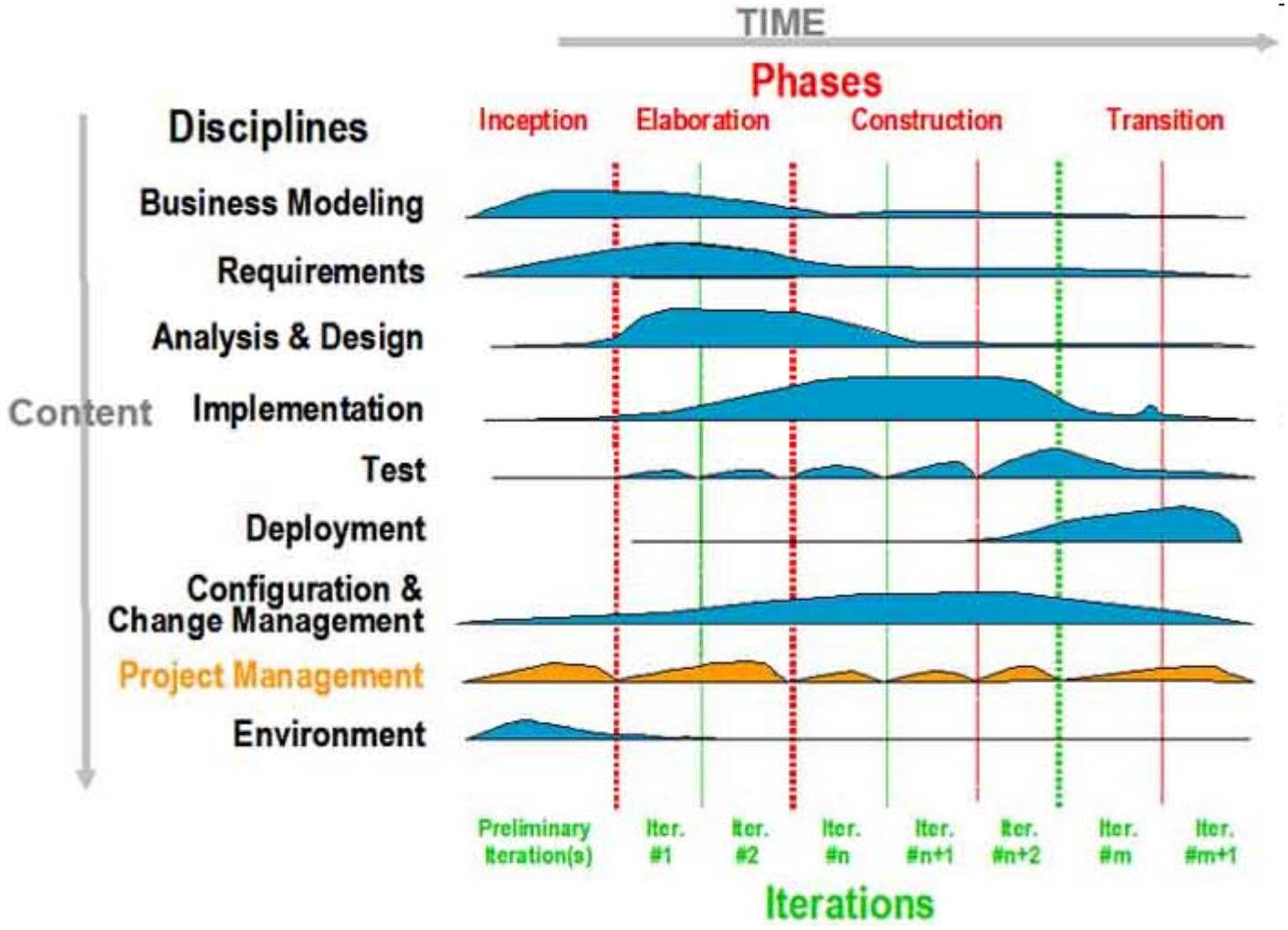
Quelle: A. Silkora, R. Drescher: Software-Engineering und Hardware-Design, Hanser, 2002

Rapid Prototyping



Quelle: C. Floyd u.a., in: Informatik-Spektrum Heft 1, Feb. 1997.

Rational Unified Process (RUP)



Quelle: IBM

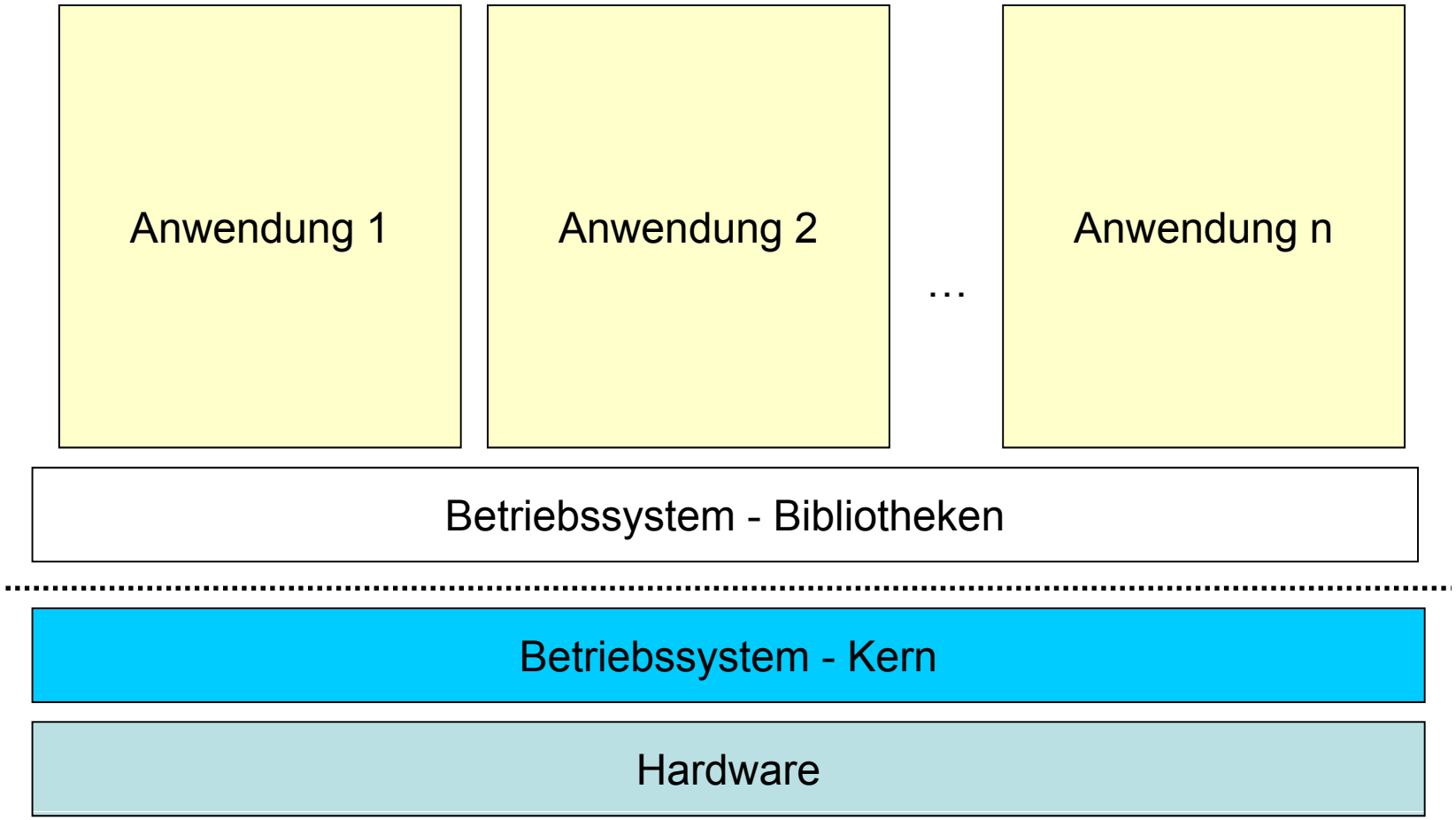
Softwaretechnische Aspekte

1. Vorgehensmodelle
2. Echtzeit Programmierung
3. Testen
    1. Allgemeines
    2. Remote Debugging
    3. In Circuit Emulation
    4. Hardware in the loop / Software in the loop
    5. JTAG

„Echtzeitbetrieb ist ein **Betrieb eines Rechnersystems**, bei dem Programme zur Verarbeitung anfallender Daten **ständig** derart betriebsbereit sind, dass die Verarbeitungs-ergebnisse **innerhalb einer vorgegebenen Zeitspanne** verfügbar sind.“ (DIN 44300)

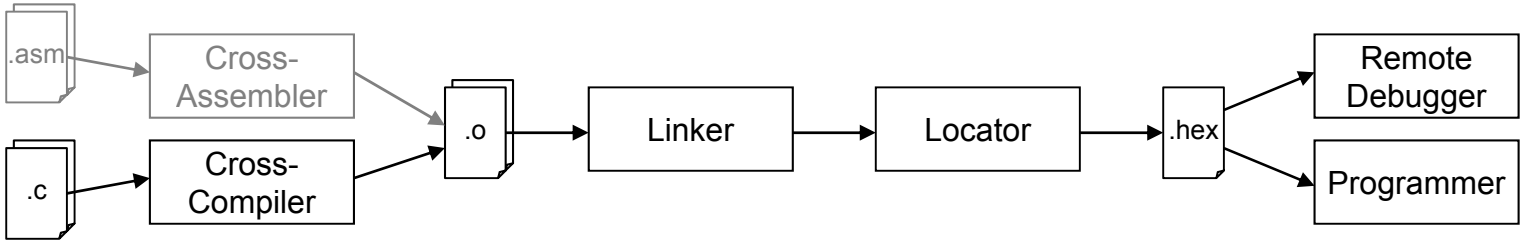
→ siehe 1. Teil der Vorlesung

Softwarearchitektur



Embedded C / C++

- Übersetzungsvorgang
  - Cross-Compiler übersetzt Quelltext in Binärcode für die Zielarchitektur
  - Locator passt Speicheradressen für Programm- und Datenbereiche entsprechend der vorhanden ROM / RAM Adressen an



```
void led_out(char ziffer, char dot, int stelle) {
/* Ziffer: 0-15 (A-F),'-','<Space>'
   Dot: >0:
   Stelle: 0-7 */

    /* 7 – Segment Repräsentation der Ziffern 0-9, a-f, -) */
    unsigned char ztab[18] = {0x3F,0x06,0x5b,0x4f,0x66,0x6d,0x7d,
            0x07,0x7F,0x6F,0x77,0x7c,0x39,0x5e,0x79,0x71,0x40,0};
    /* Adressen der 7-Segment Anzeigen */
    unsigned char digit[8]={0x20, 0x10, 0x40, 0x30, 0x60, 0x50,
                  0x80, 0x70};
    unsigned char out;
    if (ziffer=='-') ziffer=16;
    if (ziffer==' ') ziffer=17;
    if (ziffer>17) ziffer=17;
    out = ztab[ziffer];
    if (dot) out |= 0x80; // Punkt einschalten, falls erforderlich
    DP1L = 0xFF; // Richtung Datenbusport: Ausgabe
    P1H = digit[stelle]; // Adresse auf Adressbus ausgeben
    P1L = out; // Wert auf Datenbus ausgeben
    P1H = 0; // Adressbus zurücksetzen
}
```

Assembler

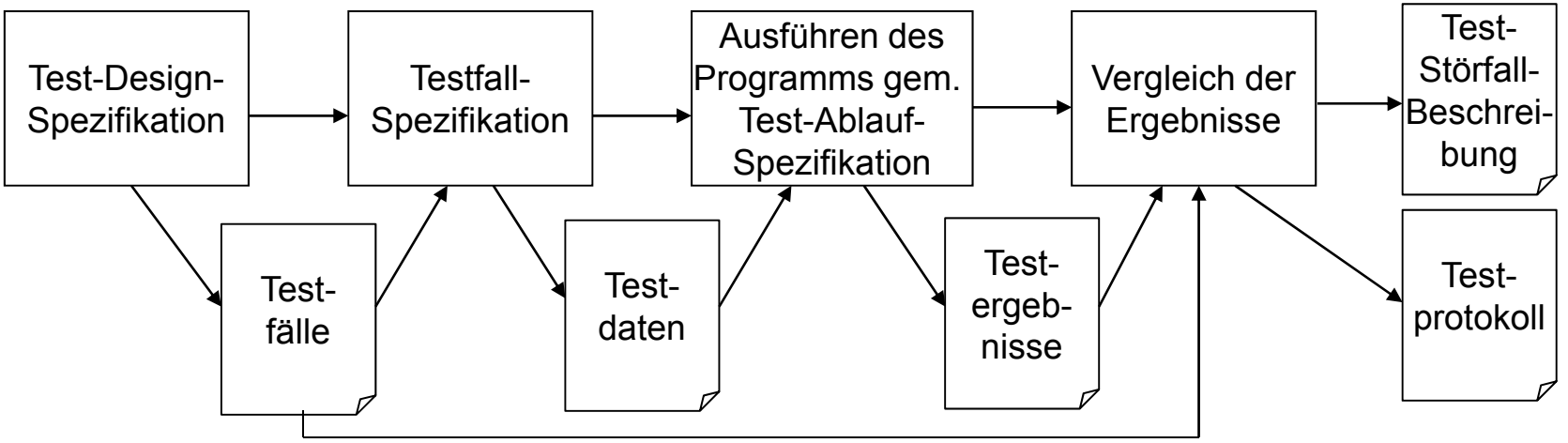

```
        MOV DX, 0B0H ; 1. Ausgabeadresse in DX
        MOV EDI,10
        MOV ESI,OFFSET ziff ;Adresse von ziff ins Register bringen
m2:     MOV AL,[ESI+EDI-1]
        OUT DX,AL
        INC DX ; nächste Displaystelle
        DEC EDI ; nächste Ziffer
        JNZ m2 ; Solange wie EDI nicht 0 zurück zu m2

ziff    DB 3FH,03H,6DH,67H,53H,76H,7EH,23H,7FH,77H
```

Softwaretechnische Aspekte

1. Vorgehensmodelle
2. Echtzeit Programmierung
3. Testen
    1. Allgemeines
    2. Remote Debugging
    3. In Circuit Emulation
    4. Hardware in the loop / Software in the loop
    5. JTAG

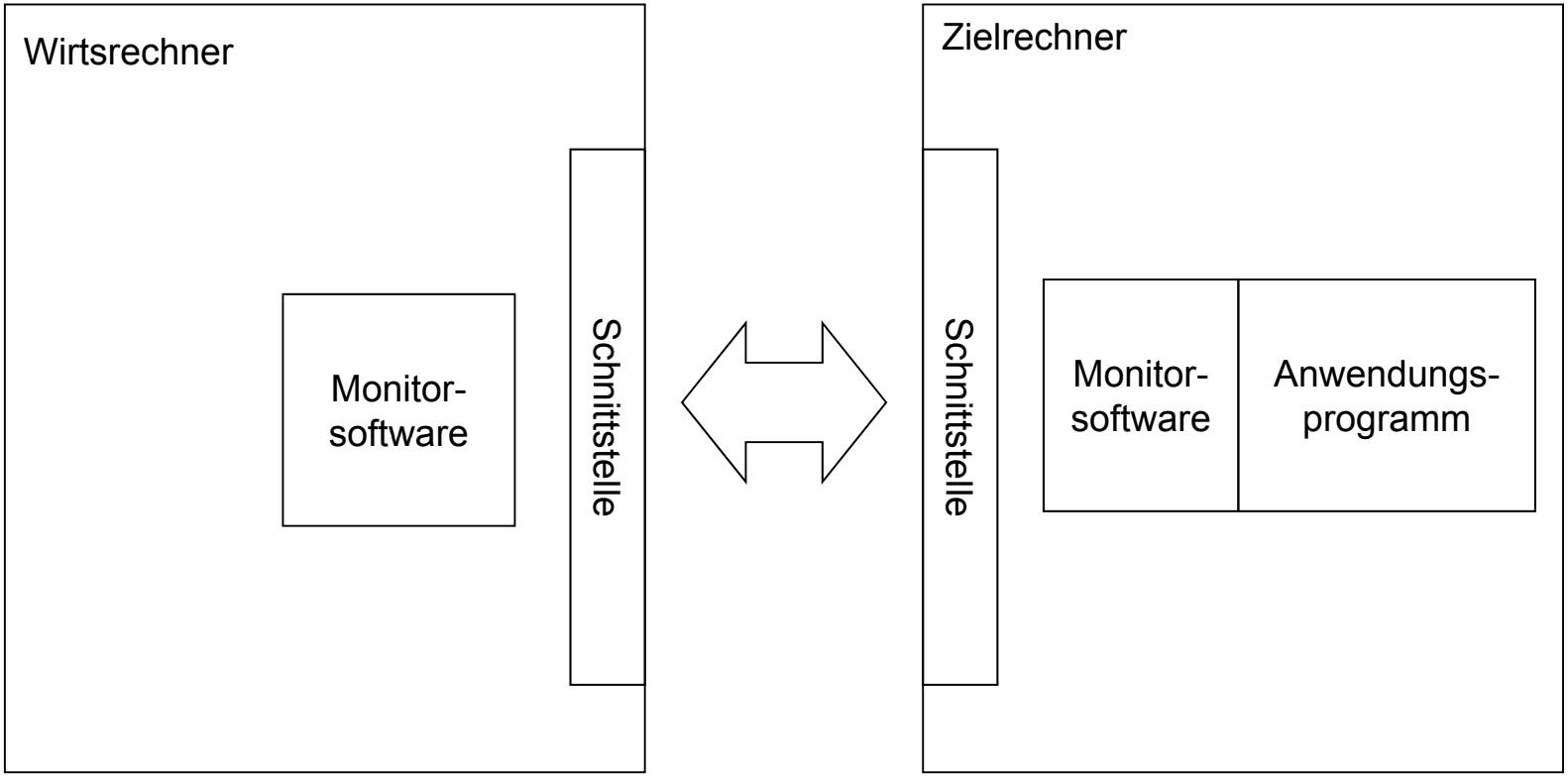
Vorgehensweise



Softwaretechnische Aspekte

1. Vorgehensmodelle
2. Echtzeit Programmierung
3. Testen
    1. Allgemeines
    2. Remote Debugging
    3. In Circuit Emulation
    4. Hardware in the loop / Software in the loop
    5. JTAG

Remote Debugging



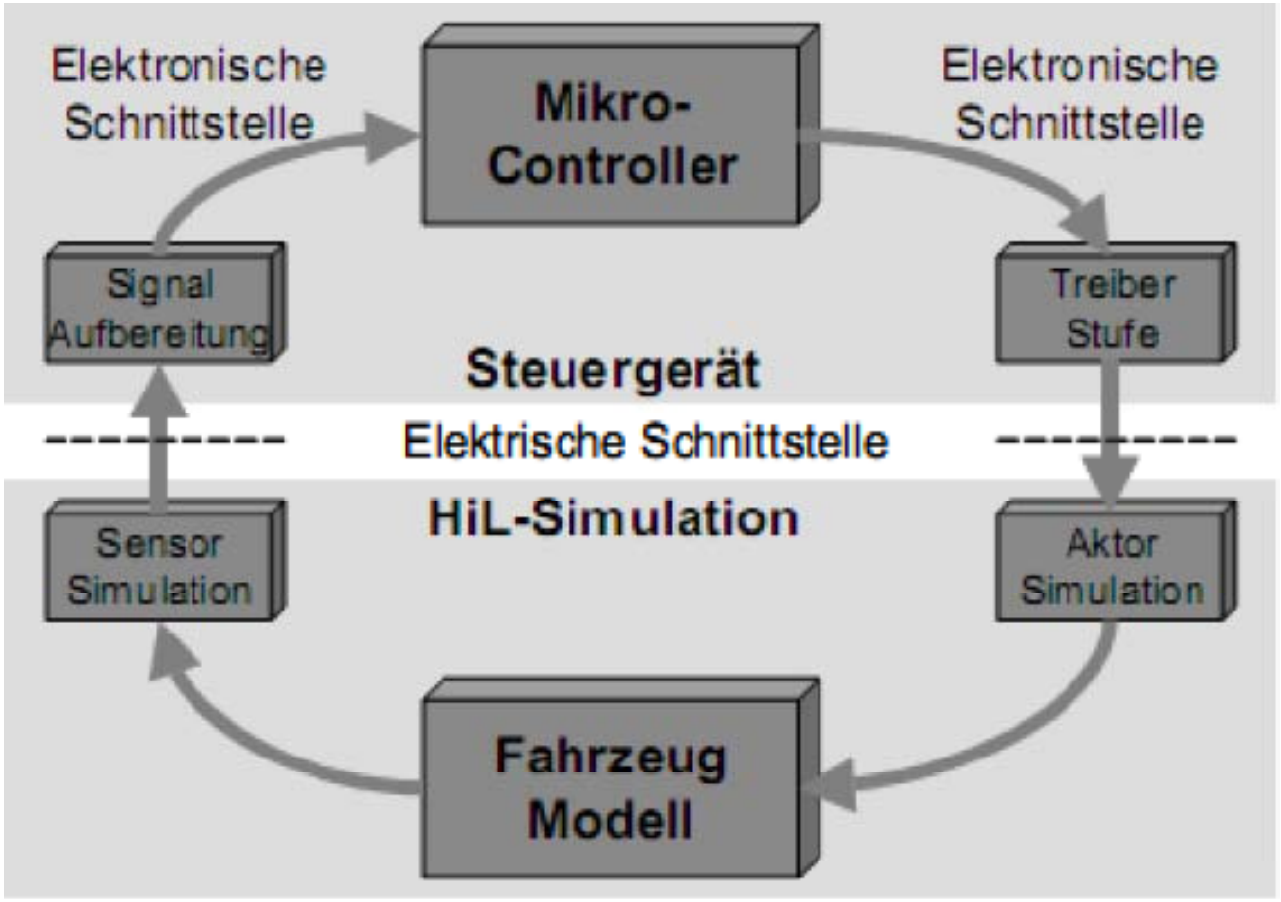
Softwaretechnische Aspekte

1. Vorgehensmodelle
2. Echtzeit Programmierung
3. Testen
    1. Allgemeines
    2. Remote Debugging
    3. In Circuit Emulation
    4. Hardware in the loop / Software in the loop
    5. JTAG

Softwaretechnische Aspekte

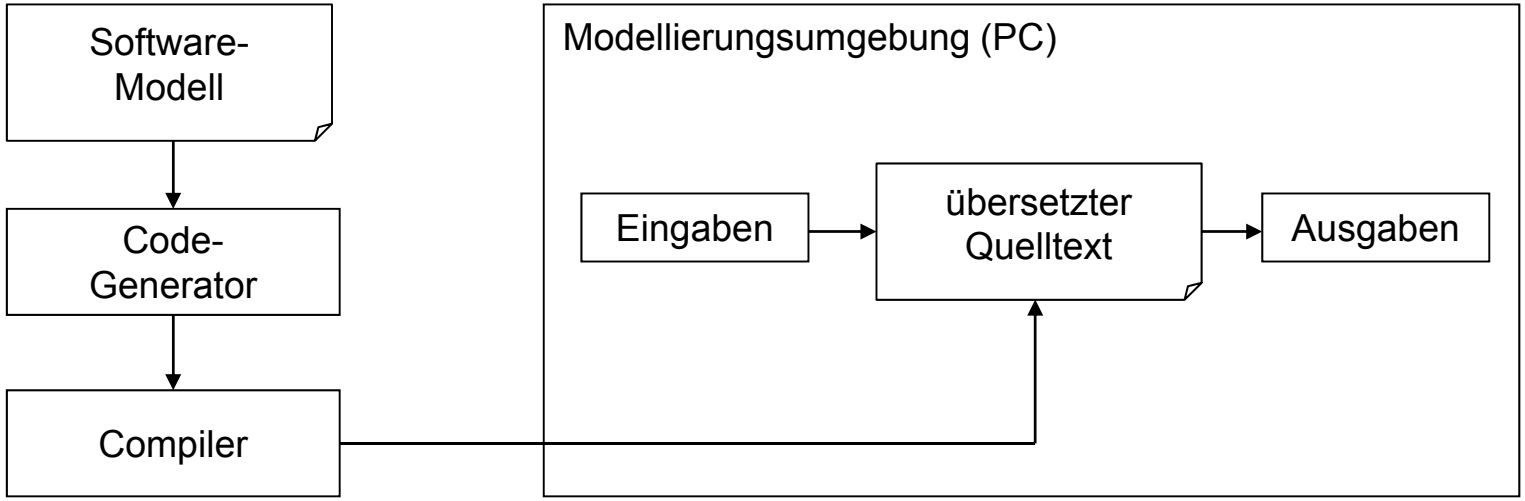
1. Vorgehensmodelle
2. Echtzeit Programmierung
3. Testen
    1. Allgemeines
    2. Remote Debugging
    3. In Circuit Emulation
    4. Hardware in the loop / Software in the loop
    5. JTAG

Hardware in the loop



Quelle: Dissertation „Modellbasierter Hardware-in-the-Loop Test von eingebetteten elektronischen Systemen“ , Dipl.-Ing. Bernhard Spitzer, Mannheim, 2001

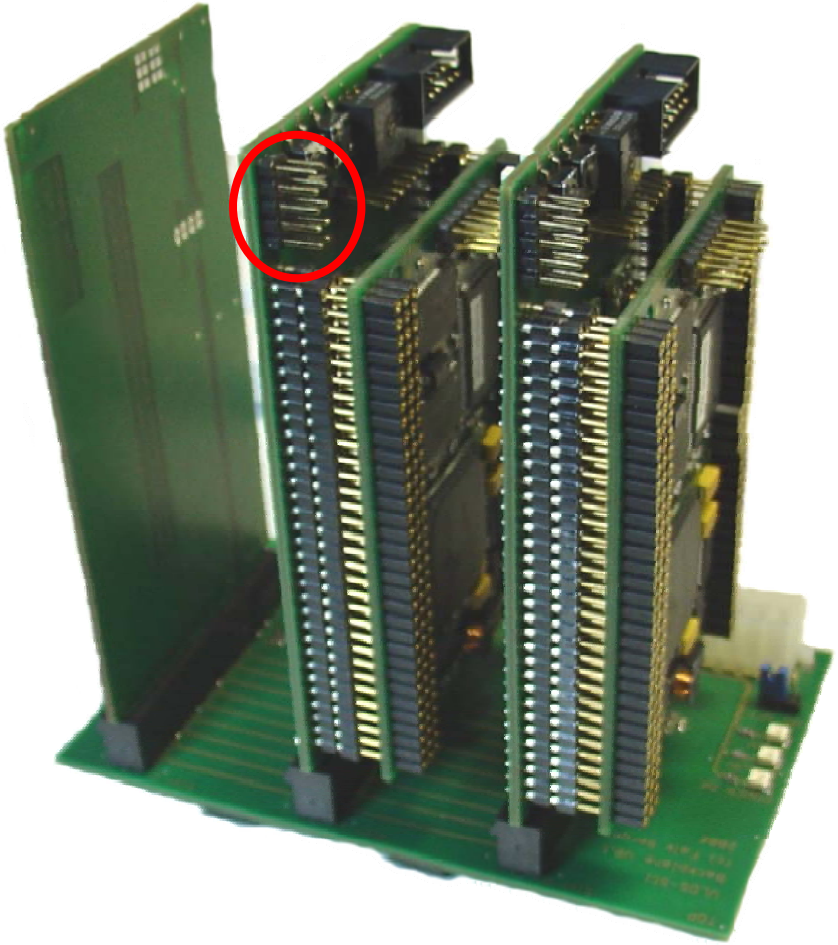
Software in the loop



Softwaretechnische Aspekte

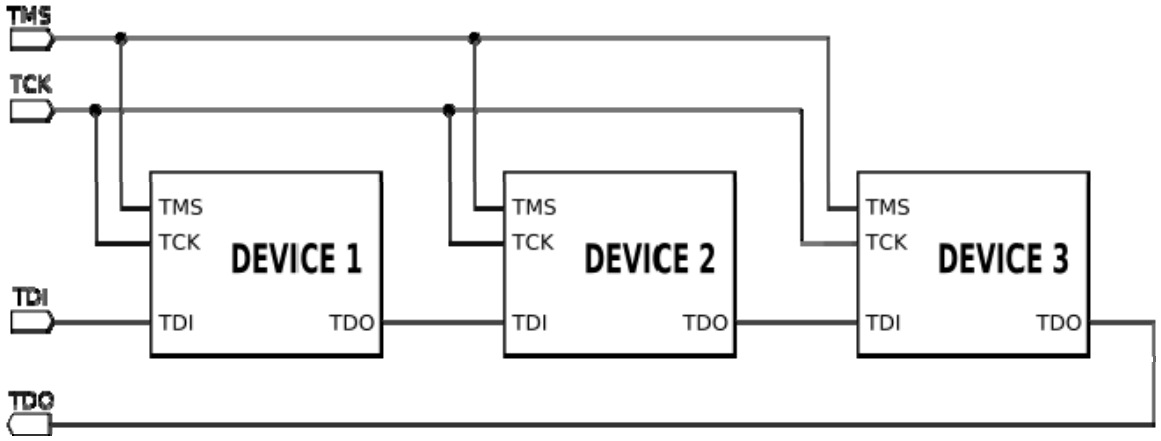
1. Vorgehensmodelle
2. Echtzeit Programmierung
3. Testen
    1. Allgemeines
    2. Remote Debugging
    3. In Circuit Emulation
    4. Hardware in the loop
    5. JTAG

JTAG

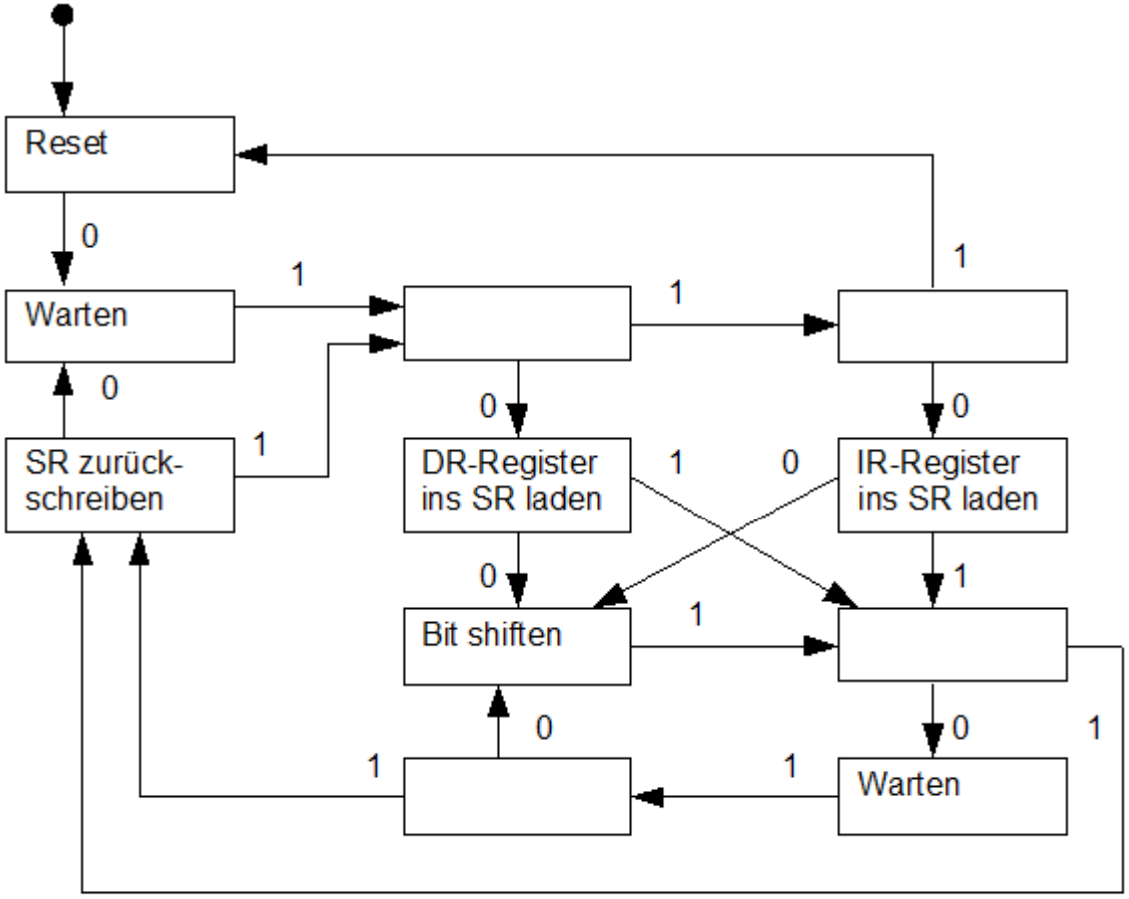


JTAG

- Aufbau als Kette aus Schieberegistern, serielles Lesen & Schreiben
- Signale
  - Test Data In (TDI)
  - Test Data Out (TDO)
  - Test Mode Select (TMS)
  - Test Clock (TCK)
  - Test Reset (TRST), optional



Auswahl des Registers über das TMS Signal



Legende
SR – Schieberegister
DR – Datenregister
IR – Instruction Register

JTAG

- Register (herstellerspezifisch)
  - Instruction Register (IR)
    Auswahl des Datenregisters, Ausführen herstellerspezifischer Befehle
  - Datenregister (DR)
    - Boundary Scan Register
      Repräsentiert die PINS des IC
    - IDCODE Register
      Enthält Herstellercode und Produkttyp zur Identifizierung des IC
    - Bypass Register
      Dummy-Register, wird verwendet, wenn bei mehreren ICs in der Schiebekette nur ein IC gelesen werden soll
    - Sonstige Register
      Weitere herstellerspezifische Datenregister, die beispielsweise interne Zustände enthalten und zum Debugging verwendet werden