

# K 1 Einführung in die Prozessdatenverarbeitung(PDV) und Begriffe

## 1.1 Einführung und Geschichte der PDV

### Geschichte der PDV

#### 1. Beginn in den 50er Jahren:

- Erste Realzeitrechner (PR) - Prozessrechner (PR)- zur PDV Mitte bis Ende der 50er Jahre
- Entstanden aus kommerziellen und wissenschaftlichen Rechnern
- Besonders an PR: Erweitert um Komponenten zur Prozesskommunikation
- ► damals: Kosten 1-4 Mio. DM, ► Großindustrieeinsatz
- Erste Einsatzgebiete: chemische, petrochemische und Stahlindustrie zur Überwachung von Anlagen
- Probleme:
  - PR sehr teuer, also wurde versucht viele Funktionen parallel zu implementieren ► Überforderung der Leistungsfähigkeit
  - mangelnde Integrationsfähigkeit mit bestehenden regelungs- und steuerungstechnischen Komponenten

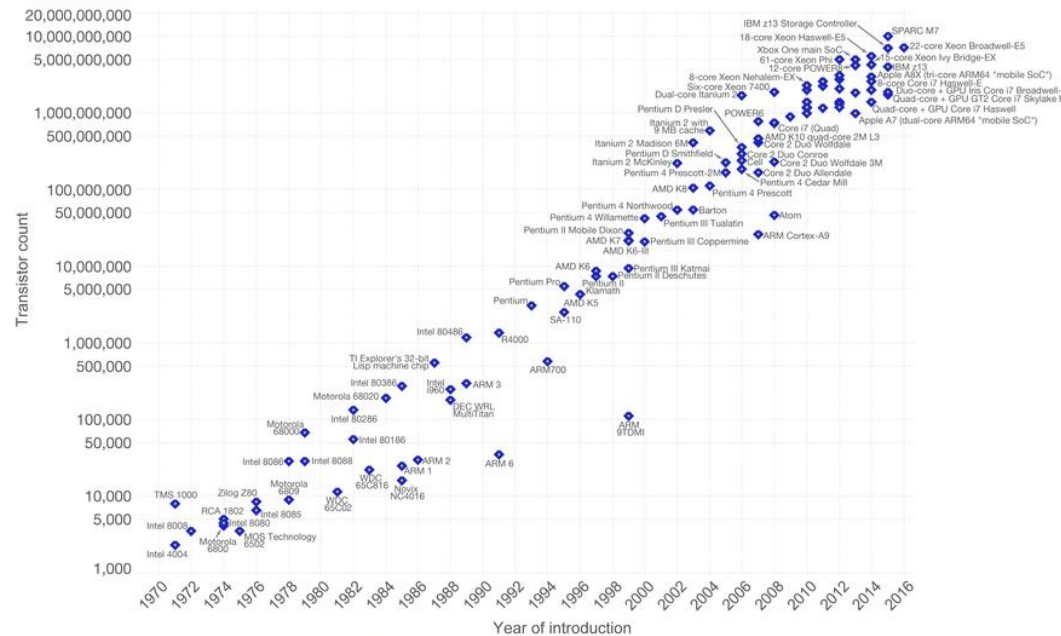
#### 2. Mitte 60er Jahre:

- Aufkommen von Kleinrechnern als PR
- Start der systematischen und stürmischen Entwicklung der Prozessleittechnik
- Meilensteine:
  - Konzepte zur Modularisierung und hierarchischen Aufteilung der Prozessleit- und Regelaufgaben
  - Auf der prozessnahen untersten Ebene Einsatz von Kleinrechnern, obere Ebene Großrechner
- Probleme:
  - brauchbare Realzeitsprachen und Betriebssysteme nur für große PR
  - Assemblerprogrammierung und "aus dem Vollen" entwickeln dominiert
  - Kosten immer noch sehr hoch (Kleinrechner immer noch 50-150 TEURO, DEC PDP 8/11, Siemens Serie 300)

### 3. ab 70er Jahre

- Anfang 70er: Entstehen von brauchbaren Betriebssystemen, höheren Realzeitsprachen bzw. Cross-Software (Entwicklung am Großrechner aber Laufzeit am Kleinrechner) für Kleinrechner, Fortran und Assembler!!
- Strukturgrößen von DRAMs bei rapidem steigender Integrationsdichte
- stürmische Entwicklung der Mikroprozessorbausteine am Beispiel Intel

Moore's Law – The number of transistors on integrated circuit chips (1971–2016) Our World in Data  
 Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



Data source: Wikipedia ([https://en.wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count))  
 The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.  
 Licensed under CC-BY-SA by the author Max Roser.

Bild 1: Quelle Max Roser

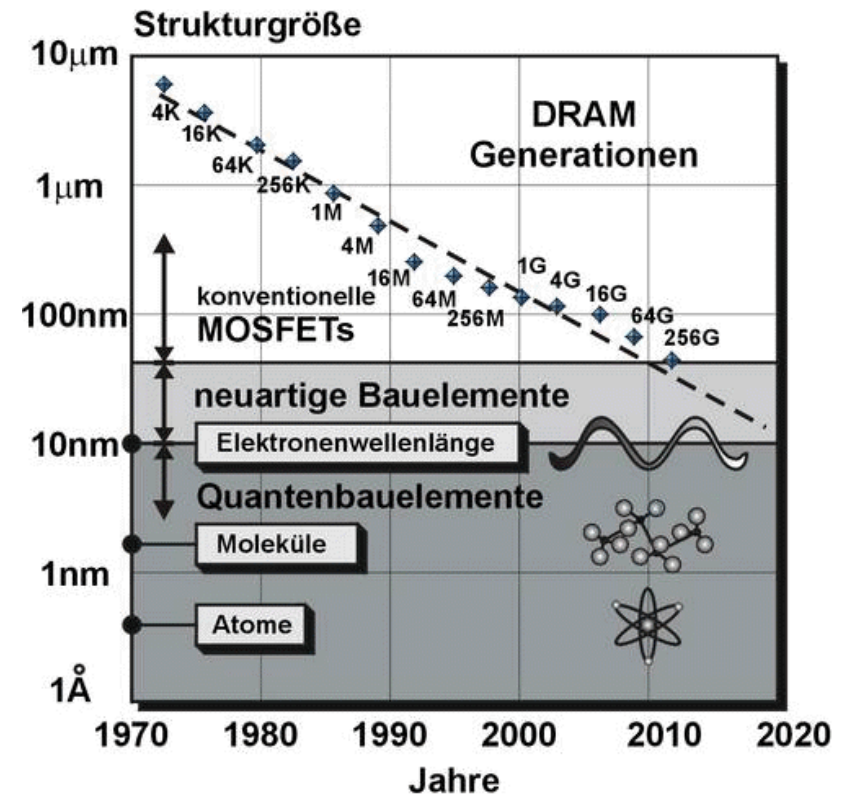


Bild 2: Typische Chip-Strukturgrößen im Wandel der Zeit

#### 4. Heute:

- Mikrorechner:= Mikroprozessor + Speicher + Peripherie ist der typische PR heute => der heutige PR unterscheidet sich von klassischen EDV-Rechnern nur noch geringfügig. d.h.: EDV hat sich in Richtung PR entwickelt.
- "klein" PR zur Anlagenautomatisierung vergleichbar zu 60er Jahre: Kosten heute deutlich unter 10 T€, typisch 2-10 T€, => Technik: Standard-32bit/64bit Micro als Kern, hohe Taktraten (3 Ghz und mehr möglich) , 0,5 -16 Gbyte RAM, Festplatte >>=500 GB, Graphiksubsystem, Maus, Tastatur, vernetzt über (Gb)LAN (Ethernet), Ankopplung der Peripherie über Feldbus / Sensor-Aktorbus oder Sternverkabelung
- Typische Kennzeichen von PR der "untersten" prozessnahen Ebene im Fabrikeinsatz:
  - CPU: Mikroprozessor (16) 32, 64 bit, hohe Taktraten
  - Arbeitsspeicher: dynamische RAM im Ausbau 2(und mehr) Gbyte
  - Universalbus : PCIe PC-BUS (VME-BUS) ....
  - Standardperipherie:
    - LAN-Ankopplung 10/100 Mbit 1/10Gbit (TCP/IP) zu übergeordnetem Leitreehner
    - lokaler Massenspeicher vorhanden z.B.: Platte 500-... GByte
    - entweder (sinkende Tendenz): Peripherieansteuerung über Einsteckkarten und Sternverkabelung zum Prozess
    - oder aktuell: Sensor/Aktor-Bus, Feldbus (PROFINET, CANOpen, EtherCAT, ...)
  - oft: Standardbürorechnerausführung eingebaut im klimatisiertem und staub- und spritzwassergeschützen 19"Schrank vor Ort in Prozessnähe in der Fabrikhalle
- **Wachstumsmarkt:** wirtschaftlicher Einsatz PR bis hinunter zu der Familie der "embedded Systems"

- statt komplexes PR komplettes Automatisierungs-/Steuerungssystem auf einer Platine (embeddedRZ) ► Bestreben eine "Ein-Chip-Lösung" zu erreichen. i.d.R. keine Festplatte, Tastatur, Bildschirm ...
- ► ASICs, FPGAs, SystemOnAChip(SOC), Microcontroller 4/8/16/32/64 Bit, der beinhaltet auf Chip: CPU, FLASHRom, RAM, digitale I/O, Timer, A/D, D/A, PWM, Kommunikationsbausteine... ► gute EMV-Eigenschaften, da oftmals nur noch Leistungselektronik außerhalb des Microcontrollers auf der Platine zu finden ist ► keine externen ROM/RAM Bausteine!!
- typische Hardwarekosten im 0,5-100 € Bereich, 8-32(64)bit Microcontroller mit ROM/RAM -teilw. extra. Einsatz von ASICs in denen alle Sonderfunktionen auf einem Chip integriert sind.
  - Stückzahlen min. 0,5-x Mio. pro Sorte pro Jahr
- embeddedRZ innerhalb von Großserienhaushaltsgeräten z.B.: Waschmaschine, Fotoapparat, Videorecorder, CD/DVD-Player etc.
  - Kosten im Euro Bereich, 8-bit/16-bit/ 32-bit Microcontroller mit ROM/RAM auf dem Chip, Mio.-Stückzahlen/Jahr.
- embeddedRZ innerhalb Automobilelektronik u.a. sog. Steuergeräte w.z.B. Motronik, Airbag, Bordcomputer....
  - Steuergeräte hochgradig vernetzt. Datenaustausch während der Fahrt. Einsatz von Feldbussen: z.B. CAN (► Erfinder Bosch), FlexRay
  - SOCs mit Microcontroller 8/16/32 Bit, 4-128 K RAM on-Chip A/D, D/A, I/O , Feldbus, Spezialperipherie , RAM und FlashROM.
  - Randbedingungen:
    - höchste Zuverlässigkeit bei widrigen Umweltbedingungen w.z.B. hohe Temperaturschwankungen, Vibration und Nässe
  - EMV ► Abstrahlung und vor allem Unempfindlich gegen Einstrahlungen. ESD (Elektrostatikfest)

- Sehr kostensensitiv ► Einsparung von "Centbeträgen" an der Serienhardware, daher individuelle Entwicklungen
- Standardisierung von Echtzeit-Betriebssystemeigenschaften und Vernetzungsfunktionen ► OSEK, AUTOSAR um die Entwicklungszeiten und -kosten zu minimieren.
- embeddedRZ innerhalb Telekommunikation
  - SOC Handy's GSM ein Chip Lösungen/ UMTS drei Chip Lösungen (noch ohne Betriebssystem, Generation UMTS aber teilweise mit Realzeitbetriebssystem) kosten zwischen 8-16 Euro/Chip
  - SOC's Netzwerkkomponenten: Router, Switches....
- embeddedRZ innerhalb Multimedia
  - Set-Top-Boxen des digitalen Fernsehens
  - Video/DVD/Blue-ray-Player/Streaming-Boxes.....
- **Heute immer noch: statt Hardwarekostenproblem nun Softwarekosten- und Testkostenproblem:**
  - Es gibt mittlerweile mehr als genügend leistungsfähige und kostengünstige Echtzeit-Betriebssysteme (>200 verschiedene) und (Realzeit)-Programmiersprachen für PR
  - **Aber:**
    - Seit Mitte der 80er Jahre steigen die Kosten der Erstellung von Software bei sinkender Hardwarekostentendenz ► typische PR-Kosten eines mittleren Automatisierungsprojektes: 50 T€ Software und 10 T€ Hardware
    - Der all-round Automatisierer der 80er musste sich anwendungsnah spezialisieren um quasi-Standard-Anwendungs-Software erstellen zu können damit die Kosten reduziert werden konnten.
    - seit Anfang 90er bis heute verlangt der Anwender (► der Kunde der Automatisierer) nach Softwareengineering-Tools und Softwarehilfsmittel die es ihm gestattet **selbst und wenn möglich**

**ohne Programmierkenntnisse** Automatisierungssysteme zu projektieren, konfigurieren, in Betrieb zu setzen aber vor allem selbst zu pflegen.

- Herausforderung an den Automatisierer: Die zugestandenen Entwicklungszeiten werden immer kürzer, die Entwicklungskosten müssen permanent reduziert werden und die Automatisierungsaufgaben werden immer komplexer.
- **Ursache:**
  - der Anwender kennt seinen Prozess selbst am genauesten, weiß ihn bestens zu optimieren
  - die Automatisierung findet immer mehr schritthaltend mit der Optimierung und Produktverbesserung des Gesamtprozesses statt ► Know-how nur beim Anwender der das Produkt herstellt,
  - ► Es dauert zu lange bis jemand das erklärt bekommen hat.
- **Wirkung:**
  - Der typische Anbieter von Automatisierungstechnik wird zukünftig der Standard-Hardware/Softwaretoolslieferant und hilft beim Projektieren des Erstsystems ► Zwang zur **ISO-Standardisierung**, um Mehrfach verkaufen zu können, denn die Entwicklungskosten zahlt längst nicht mehr nur 'ein' Kunde.
  - Der Projektingenieur beim Anwender projiziert, realisiert und wartet die Automatisierungssysteme selbst .
- **Trend:**
  - Seit spätestens der Jahrtausendwende zeichnet sich ab, dass die Multimediaanwendungen für Jeden und Anwendungen der Telekommunikation, w.z.B Handy, den Entwicklungsfortschritt im Echtzeitbereich extrem beeinflussen.
  - Es existieren mittlerweile ausgefeilte (teure) Toolketten, die es gestatten komplette Applikationslösungen integriert vom Entwurf über die Implementierung bis hin zum Test in einem 'Guss' zu entwickeln. (► Rhapsody von IBM Rational...).

### 1.1.1 Ziel der Prozessdatentechnik (PDV)-Vorlesung, oder was ist Prozessdatentechnik?

- Es gibt kein homogenes klar abgegrenztes Gebiet der Prozessdatentechnik (PDV).
- Die PDV ist ein Spezialgebiet der Datenverarbeitung (DV) , bei der als Besonderheit die **Lenkung** eines **technischen Prozesses** im Vordergrund steht

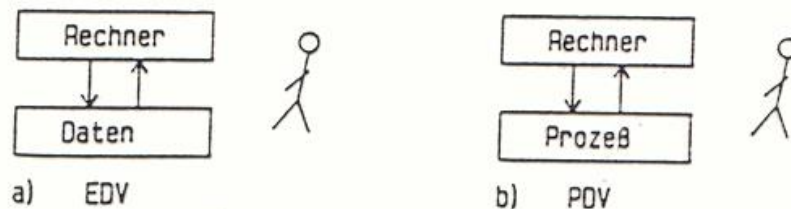


Bild 1.1 EDV und PDV

- a) In der EDV benutzt der Mensch den Rechner als Hilfsmittel zur Verarbeitung von Daten.
- b) In der PDV dient der Rechner vorwiegend zur Lenkung eines (technischen) Prozesses und entlastet den Menschen.

- Der technische Prozess erweist sich als Quelle und Senke von Daten.
- Dabei gilt: Die Erfassung und die Erzeugung von zeitlichen Abfolgen von Daten und die Einhaltung von strikt vorgegebenen Bearbeitungszeiten ist hierbei besonders wichtig:
  - ▶ Das wird auch als **Realzeitbedingungen** bezeichnet.

**Die Einhaltung der zeitlichen Prozessgegebenen Abfolgen definiert die Leistungsanforderungen, den Umfang und Ausprägung der PDV mit der Maßgabe die Verarbeitung schritthaltend mit dem Prozess und die Einhaltung der gegebenen Anforderungen der Betriebssicherheit zu gewährleisten.**

1. Schritt: Wie kann man den technischen **Prozess** näher fassen? ► alles das in Kapitel 1 der Vorlesung

- Existieren gemeinsame Merkmale von unterschiedlichen Prozessen ► gleiche Merkmale, gleiche technischen Probleme zu lösen?
  - Wie lassen sich Prozesse beschreiben? ► SDL, SADT, Zustandereignisdiagramme....
  - Stichworte: Zuverlässigkeit, Fehlertoleranz, Sicherheit
- 
- Die **Steuerung** des Prozesses erfolgt in PDV durch einen **Prozessrechner(PR)** mit **Realzeitbetriebssystem**.
    - **'embedded' Systeme und Maschinensteuerungen** ► keine SPSEN, eigene Vorlesung
    - **keine Prozessperipherie** ► Vorlesung: Messtechnik, Sensortechnik
    - **keine Microcontroller** ► Vorlesung: Microcontrollervorlesung + Assemblerprogrammierung
  - **Schwerpunktthema: Arbeiten mit einem Realzeitbetriebssystem (RBS) und Programmieren von Echtzeit-Anwendungen.** ► E Kapitel und Praktikum
    - Wozu braucht man RBS? ► Was versteht man unter Echtzeitverarbeitung?
    - Multitasking ► quasi zeitparalleles Abarbeiten von Aufgaben am Rechner
    - Aufbau und Funktionen eines RBS am Beispiel OS9
    - Planung der Abarbeitung von Echtzeitaufgaben auf einem PR
    - **Voraussetzung: fundierte C-Programmierkenntnisse**



## 1.1.2 Allgemein: Automatisierung von Prozessen

### 1. Zunächst landläufiges Vorurteil: "**Automat ersetzt Menschen**"

- :-( Mit dem Begriff der Automatisierung wird oft die Vorstellung verbunden: der Mensch soll durch Automaten **ersetzt** werden.
  - 😊 **Über den Einsatz von Automaten zu Unterstützung der Menschen reden**, denn der Automat kann verglichen zu den menschlichen Fähigkeiten doch nur einen sehr begrenzten Einsatzraum abdecken.
  
  - :-( Beim Automat schlecht ausgeprägt oder gar nicht vorhanden sind:
    - freie Entscheidungsmöglichkeiten
    - Kreativität zur Verbesserung
    - Spontanität
    - Mut zum Risiko
    - Taktile Fähigkeiten komplexe Dinge zu handhaben
    - Mustererkennungseigenschaften
    - Abstraktionsfähigkeiten.
  
  - 😊 Beim Automaten gut ausgeprägt: Schnelligkeit und Zuverlässigkeit,
2. Der Vorgang der Einführung von Automaten in irgendwelche Abläufe nennt man **Automatisierung**
3. Den Umfang in dem ein Vorgang oder ein Prozess automatisiert ist nennt man den **Automatisierungsgrad**.

4. Betrachtet man die automatisierten Prozesse näher so lassen sich zwei Ebenen unterscheiden:

a: **Den Vorgang,**

b: **Vorrichtung,** die diesen Vorgang steuert ohne dass der Mensch eingreift: die **automatische Steuerung(Automat)**

5. Wir befassen uns näher mit der automatischen Steuerung.

Nach DIN 19233 ist der Automat definiert als:

**"Ein Automat ist ein künstliches System, das selbsttätig ein Programm befolgt. Auf Grund des Programms trifft das System Entscheidungen, die auf der Verknüpfung von Eingaben mit dem jeweiligen Zustand des Systems beruhen, und Ausgaben zur Folge haben"**

- Die Definition legt zwei wesentliche Merkmale eines Automaten fest:
  - **EVA Struktur:** kausale Folge von **Eingabe, Verarbeitung und Ausgabe**
  - Die Verarbeitung beinhaltet **Entscheidungen zwischen verschiedenen Möglichkeiten die von der Eingabe und/oder!! vom aktuellen Zustand (Gedächtnis) des Systems abhängen.**

### 1.1.3 Ziele der Automatisierung in der PDV

- Prinzipielle Aufgabe der Automatisierung ist eine kostengünstige Vereinfachung beim Betrieb eines Prozesses.
- Es sind typischerweise vier Automatisierungsbereiche zu beobachten:
  - Automatisierung von Arbeiten, die regelmäßig und **monoton** wiederkehrend sind
  - Automatisierung von Arbeiten die **hohe Anforderung an die Konzentration** stellen.
    - ▶ Aufgaben an den Menschen mit stark wechselnden Anforderungen, hier wechseln Zeiten hoher Aktivität und hoher Anforderung an die Aufmerksamkeit und mit Zeiten geringer Aktivität.
  - Automatisierung von Arbeiten, die eine **hohe Arbeitsgeschwindigkeit** oder **Reaktionsgeschwindigkeit** erfordern.  
*Beispiel: Abfüllen von Zigaretten und Verpacken geht so schnell, dass das Auge nicht mehr mitkommt.*  
*Beispiel: Automatisches Abschalten der Drehbank bei Werkzeug- oder Werkstückbruchs*  
Hier muss zwangsläufig 100% Automatisierung erreicht werden.
- Automatisierung von Arbeiten, die eine **große Datenflut** liefern.

- **Warnung: Bisherige Grenzen in der Automatisierung von Prozessen**

- **Problem der Vollständigkeit der Automatisierung.**

Ein automatisches System kann nicht angemessen auf alle real vorkommenden Ereignisse reagieren, sondern nur auf solche, die einprogrammiert wurden.

- **Problem der Korrektheit eines Programms**

- ▶ Programmfehler können lange Zeit unentdeckt bleiben!! wenn entsprechende Zustände und Ereignisse selten sind.

- ▶ Aufgabe: Validierung der Software => Ungelöstes Problem!! wg. Komplexität

- **Problem der mangelhaften Erkennungs- und Erkenntnisfähigkeit des Rechners.**

Abhilfe: Grundlagenforschung!! Es fehlen noch grundlegende in technische Systeme umsetzbare Erkenntnisse der Wirkungszusammenhänge!!

## 1.2 Begriffe und deren Bedeutung aus der PDV

### 1.2.1 Prozess allgemein, Unterscheidung nach Verarbeitungsart und Verarbeitungsgut

- Es gibt die DIN 66 201 die besagt:
  - Ein **Prozess** ist der Vorgang zur **Umformung, zum Transport oder zur Speicherung** von **Materie, Energie oder Information**
- Veranschaulicht: Verarbeitungsart ► Vorgang und das Verarbeitungsgut ► Materie, Energie, Information

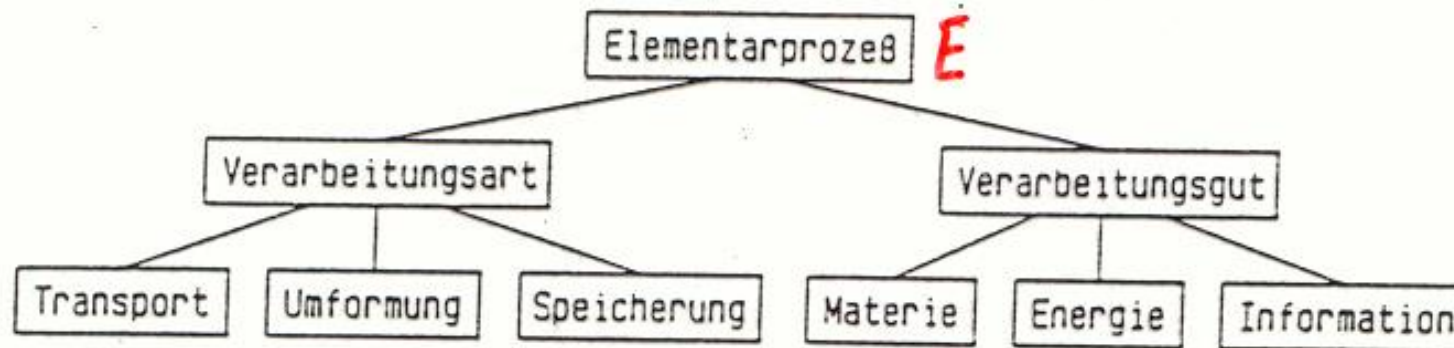


Bild 1.2 Ein Elementarprozeß wird durch eine von 3 Verarbeitungsarten und eines von 3 Verarbeitungsgütern bestimmt. So ergeben sich 9 verschiedene Grundtypen.

- *Beispiele:*

### **1. Verarbeitungsart**

- **Umformung** erfolgt in Erzeugerprozessen: Drehbank zur mechanischen Bearbeitung Umformung (Materie), Galvanisierbad mit Einsatz von Chemie und elektrischer Energie
- **Transport** von Gütern (Materie, Energie): Bestückung von Leiterplatten, Verkehrsnetze, Energieversorgungsnetze,
- **Speicherung** bezeichnet man als Aufbewahrungsprozesse z.B. Hochregallager in der Fabrik

### **2. Verarbeitungsgut**

- **Materie:** geg. in fester, flüssiger oder gasförmiger Konsistenz ► Umwandlung der Aggregatzustände kann beim Bearbeiten, Verbinden, Umwandeln (Schmelzen) Transport oder Lagerung erfolgen.
  - **Energie:** Erzeugung, Transport
  - **Information:** Verarbeitung PDV-Anlagen
- Der **Elementarprozess** ist gekennzeichnet durch **genau eine** Verarbeitungsart **und ein** Verarbeitungsgut
  - Der **Einzelprozess** stellt die kleinste geschlossene Prozesseinheit dar, bei der **ein kompletter Bearbeitungslauf** durchgeführt wird. Er ist aus mehreren Elementarprozessen zusammengesetzt, die eng miteinander verknüpft sind.
    - Eine Maschine basiert auf mindestens einem/mehreren Einzelprozessen
    - Eine Anlage besteht wiederum aus mehreren Maschinen und basiert ebenfalls auf mindestens einem Einzelprozess.
  - Der **Verbundprozess** besteht aus mehreren Einzelprozessen (Maschinen) die zusammenarbeiten um **eine gemeinsame Aufgabe** zu erledigen.
  - Der **Betriebsprozess** umfasst verschiedene Verarbeitungsbereiche eines Betriebs: Fertigung, Lager, Versand, Buchführung, Auftragsabwicklung, Verwaltung, Entwicklung.
  - Die Einbeziehung **aller** Betriebsbereiche **in einen Prozess** wird mit dem Schlagwort **CIM** (Computer Integrated Manufacturing) bezeichnet.

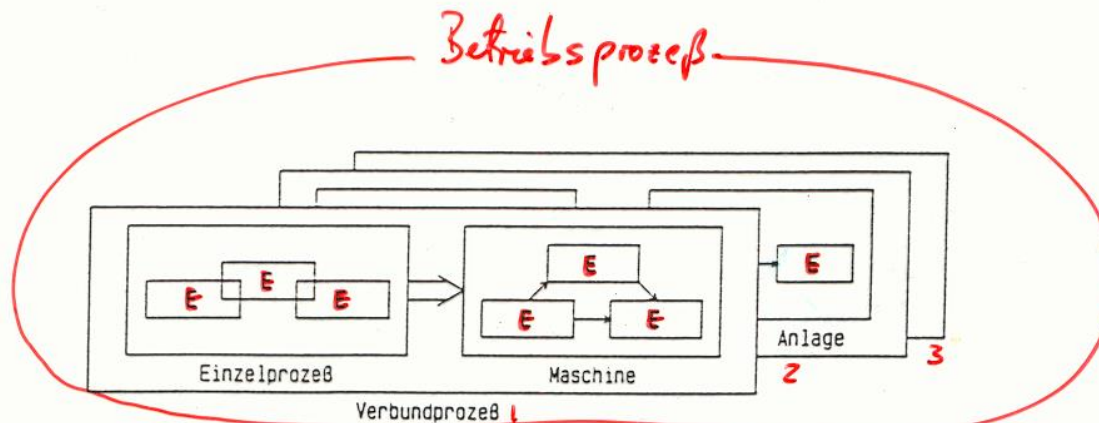


Bild 1.3 Ein Betriebsprozess setzt sich aus mehreren Verbundprozessen zusammen, die jeweils aus Einzelprozessen, Maschinen oder Anlagen bestehen. Diese wiederum sind aus Elementarprozessen (E) aufgebaut, die mehr oder minder stark miteinander gekoppelt sind.

- ▶ Hintergrund ist die Optimierung des gesamten Ablaufs in einem Fertigungsunternehmen, um Kosten und Zeit zu sparen bei Steigerung der Produktivität. ▶ Hier gibt es zwei Vorgehensweisen:
  - globale zentrale Strategie
  - Strategie der kleinen Iterationen: Prozess der stetigen lokalen Verbesserung mit dem Ziel Zeit und Kosten einzusparen bei gleich bleibender Produktivität oder Produktivität zu erhöhen bei konstanten Kosten und Zeit.

## 1.2.2 Der technische Prozess

Wir befassen uns mit einer besonderen Sorte von Prozessen:

- Ein **technischer Prozess** ist ein spezieller Prozess ▶ dessen Zustandsgrößen werden mit technischen Mitteln erfasst und beeinflusst (gemessen, gesteuert, geregelt)
- ▶ Prozessdatenverarbeitungssysteme (PDVS) bestehend aus Hardware und Software stellen u.A. solche technischen Mittel zu Beeinflussung und Erfassung bereit:
  - Die Schnittstelle PDV und Prozess ist die elektrische und logische Schnittstelle des jeweiligen Sensors und Aktors.
  - ▶ **Mit den Grundlagen der PDVS beschäftigen wir uns hier und gehen davon aus, dass die technischen Prozesse und deren Automatisierungsanforderungen bekannt und gegeben sind.**

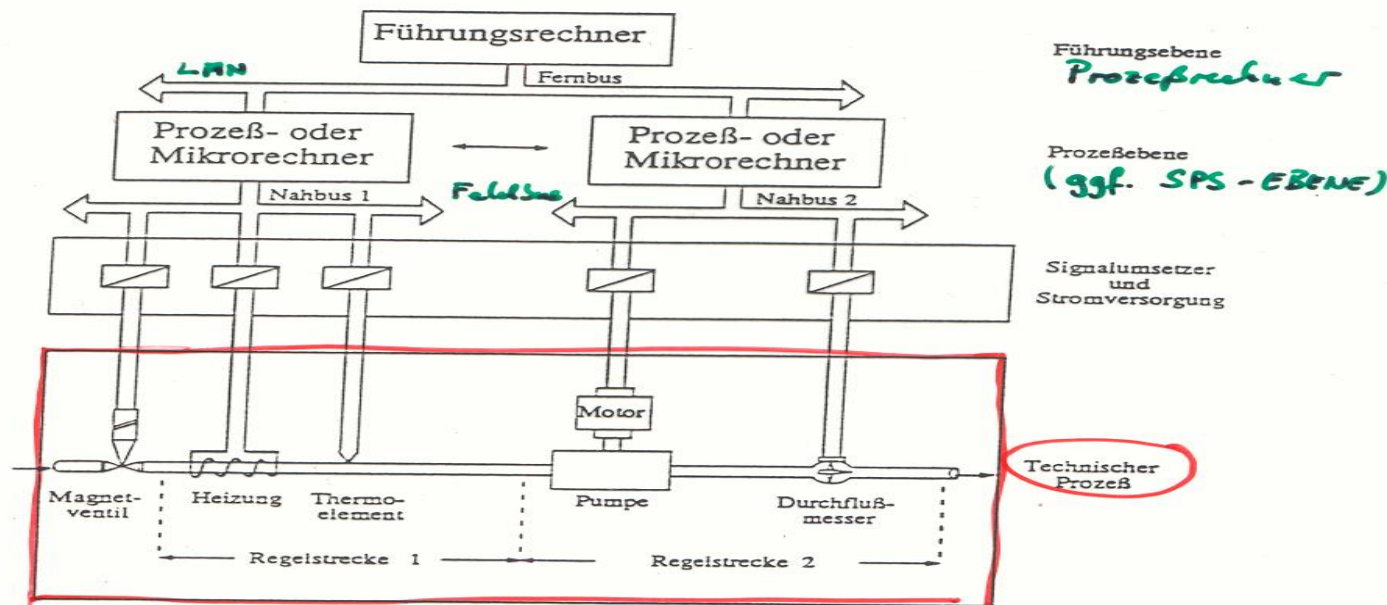


Abbildung 1.7: Steuerung eines technischen Prozesses mit Hilfe einer Rechnerhierarchie



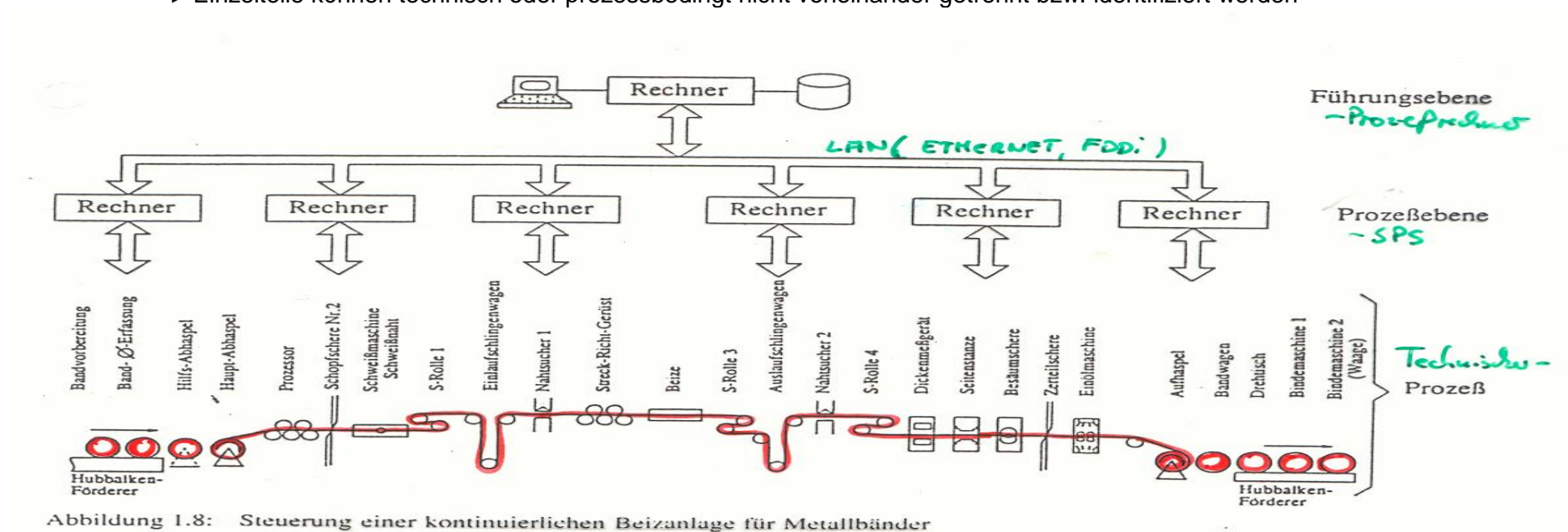
## 1.2.3 Klassifikationsmerkmale von technischen Prozessen

### 1.2.3.1 Unterscheidung nach Verarbeitungsstrukturen

- Es gibt folgende unterschiedliche Verarbeitungsstrukturen von Prozessen:

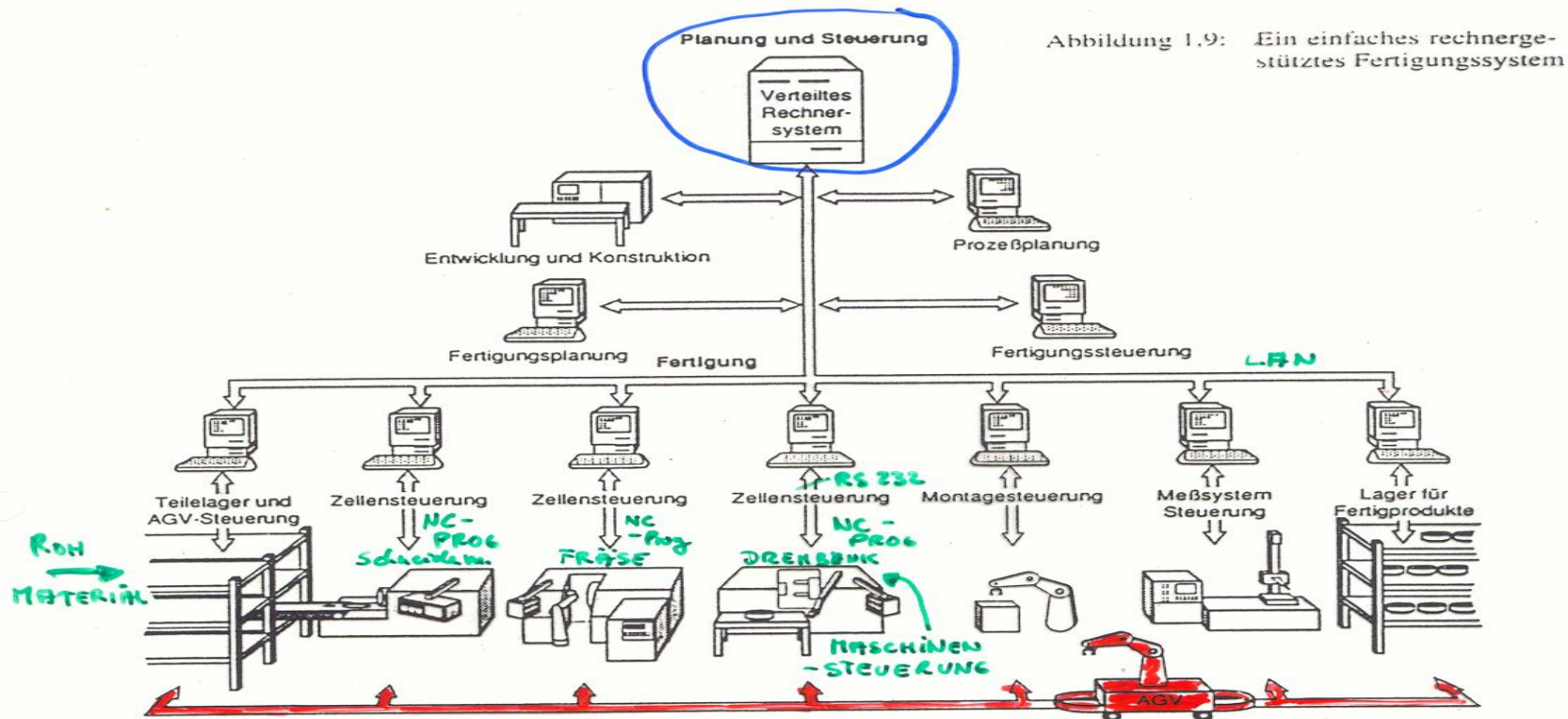
#### 1. kontinuierliche oder auch stetige Verarbeitung

- ▶ stetiger Strom von Materie oder Energie wird transportiert oder umgewandelt
- ▶ Einzelteile können technisch oder prozessbedingt nicht voneinander getrennt bzw. identifiziert werden



## 2. diskontinuierliche oder auch unstetige bzw. diskrete Prozesse

► i.A. sind das Stückprozesse, die bei der Fertigung oder Behandlung von diskreten einzelidentifizierbaren Produkten zu finden sind.



## 3. Hybride Prozesse oder auch Chargenprozesse sind die am häufigsten anzutreffenden Prozesse

► Mischform aus 1. und 2.: Die diskrete Verarbeitung von gebündelten zusammengehörig-identifizierbaren Teilmengen eines sonst kontinuierlichen Gutes. Beispiel: Brauen eines Bieres Bottichweise in Weihenstephan

### **1.2.3.2 Unterscheidung nach der inneren Ablaufstruktur**

Neben der grundsätzlichen Verarbeitungsstruktur innerhalb des Prozesses ergibt sich auch noch eine Ablaufstruktur des Prozesses die es zu berücksichtigen gilt, nach denen sich die PDT zu richten hat.

- **deterministische Ablaufstruktur**
- **stochastische Ablaufstruktur**
- **!►! Wichtig: Wahl der Algorithmen !!!**

### **1.2.3.3 Unterscheidung nach Einsatzgebiet**

- **Fertigung**
  - Verarbeitung von Gütern insbesondere der Verarbeitung von Material in Stückprozessen (diskontinuierlich) und Erzeugungsprozessen (Umformung, Herstellung)
- **Verfahrenstechnik**
  - kontinuierliche Verarbeitung oder chargenweiser Verarbeitung von Gütern und Energie zum Ziel
- **Verteilungsprozesse**
- **Verwaltung**
  - Fachgebiet der EDV
- **Mess- +Prüfprozesse**
  - Forschung und Entwicklung
  - Dienstleistungsbereich

## 1.2.4 Prozessrechner (PR)

- Als Prozessrechner wird ein Rechner bezeichnet der mittels Prozessperipherie direkt an den technischen Prozess gekoppelt ist.

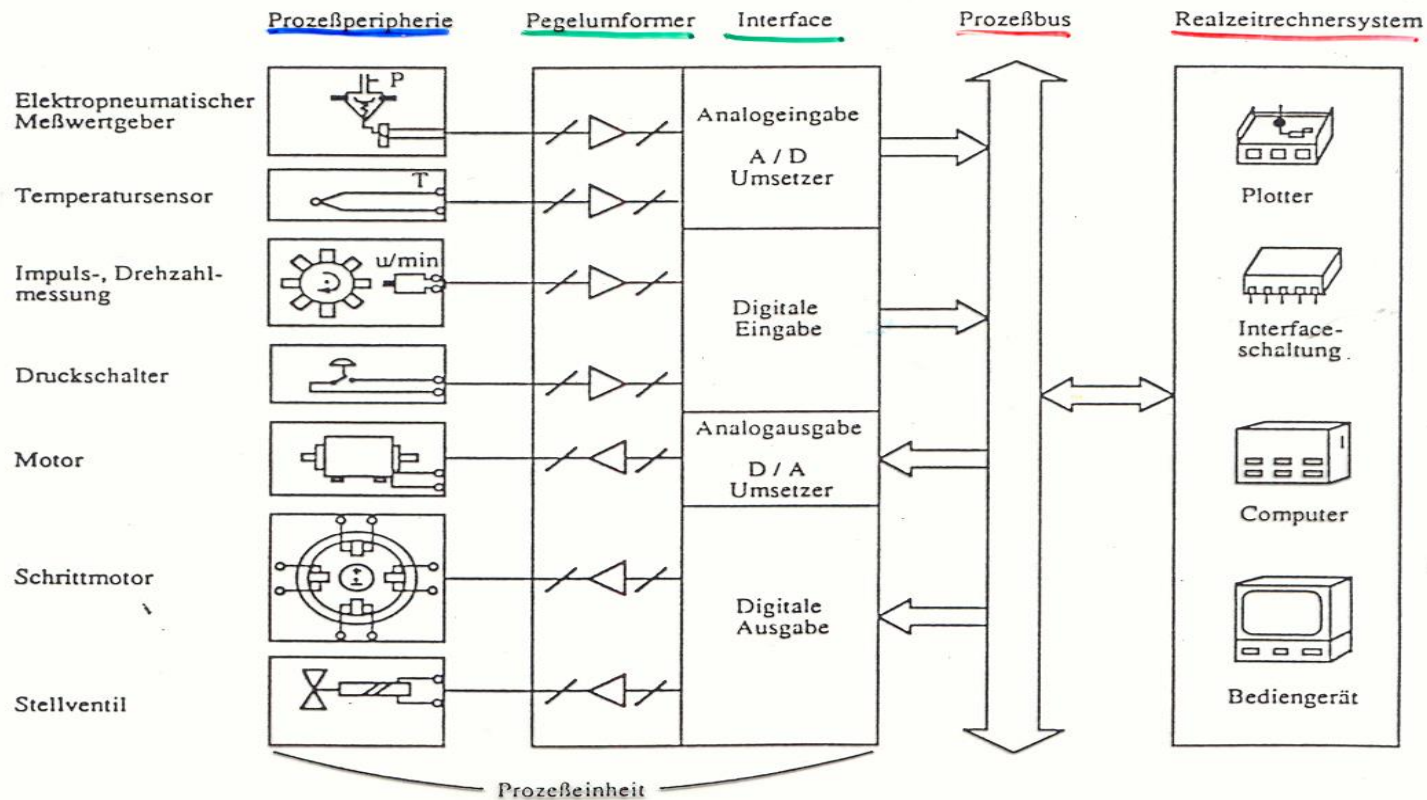


Abbildung 2.22: Realzeit- oder Mikrorechnersystem zur Prozeßsteuerung

### ***Neu Prozessperipherie:***

- ***Elektropneumatischer Messgeber,***
- ***Temperatursensor:***
- ***Impulsdrehzahlmessung:***
- ***Mechanischer Druckschalter:***
- ***Motor:***
- ***Schrittmotor:***
- ***Stellventil:***
- ***Pegelumformer und Prozessinterface***
- ***Prozessbus***

## 1.2.5 Prozessdaten und Prozessgrößen

- **Prozessdaten** werden zwischen PR und Prozess ausgetauscht
- **Prozessgrößen** beschreiben einen technischen Prozess, unabhängig von seiner Lenkung durch den Rechner.
- **Prozessgrößen:**
  - Prozessgrößen sind technische, physikalische oder chemische Größen, die mit technischen Mitteln erfasst und beeinflusst werden können. Diese werden mit geeigneten Maßeinheiten durch Zahlenwerte (Daten) beschrieben.
  - Prozessgrößen sind für jeden einzelnen Prozess und sein Verarbeitungsgut verschieden, aber es gibt für jede Verarbeitungsart charakteristische Größen:
    - Speicherprozess: die Menge ( *kg, J bzw. kWh, Stück* )
    - Transport: der Durchsatz ( *in kg/s, W, Stück/Zeit* )
    - Umformung: Art und Umfang der Änderung; wobei aber nach den Erhaltungssätzen der Physik oder Chemie die Gesamtmengen unverändert bleiben.  
*Beispiel: Lochbohren in Metall: Lochspäne + Werkstück = Gesamtmenge*

- **Prozessgrößen untergliedern sich in:**

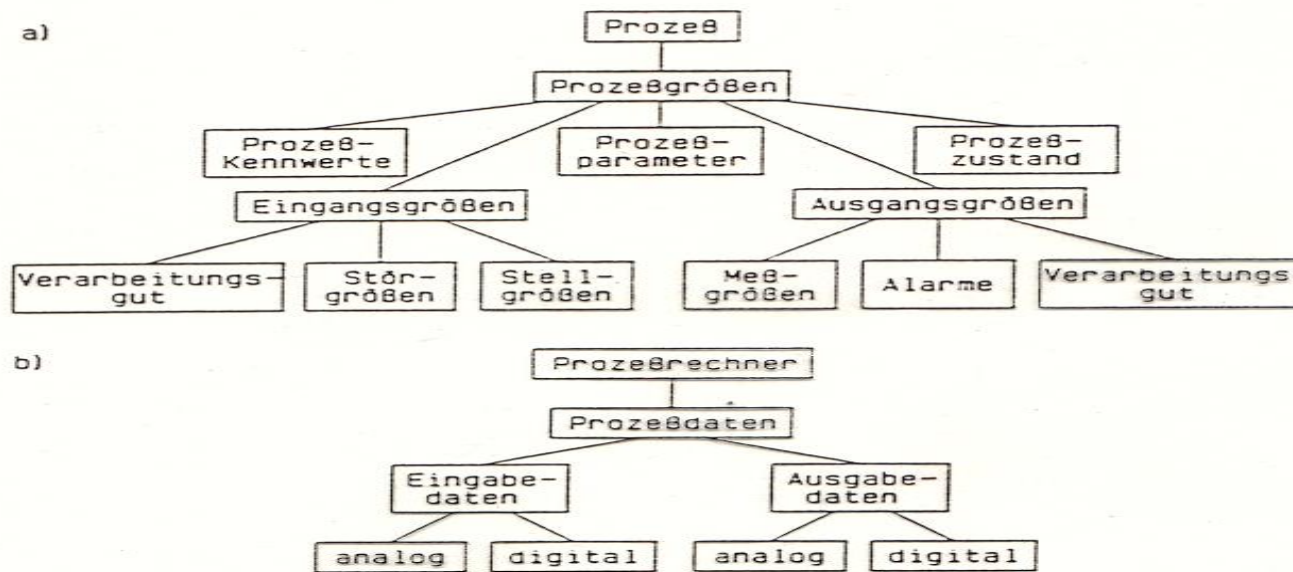


Bild 1.4 Prozeßgrößen und Prozeßdaten

a) Prozeßgrößen charakterisieren einen Prozeß. Manche von ihnen sind fest vorgegeben (statisch), z.B. das Verarbeitungsgut; andere sind veränderlich (dynamisch), z.B. der Prozeßzustand.

b) Prozeßdaten werden immer aus der Sicht des Prozeßrechners betrachtet. Die Eingabedaten sind Ausgangsgroßen des Prozesses während die Ausgabedaten zu Eingangsgroßen des Prozesses werden (vgl. Bild 1.5).

- Prozesszustandswerte und -daten
  - ▶ sind die im Betrieb auftretenden veränderlichen Größen, die den aktuellen Zustand des Prozesses beschreiben und mit technischen Mitteln erfasst und beeinflusst werden.
- Prozessparameter
  - ▶ sind solche Größen die für einen Prozesslauf konstant sind, aber für einen nächsten Prozesslauf geändert werden können.

- Prozesskennwerte
  - ▶ *oder auch Anlagedaten genannt sind technische Daten, die durch den Aufbau des Prozesses, seine Struktur, Konstruktion und Anwendung festgelegt sind. Es sind also Festwerte, zu denen vor allem auch **Grenzwerte** gehören.*
  
- Eingangsgrößen
  - ▶ *fließen in den Prozess hinein und wirken auf den Prozess. Sie werden auch als freie Variable bezeichnet, da ihre Werte unabhängig vom Prozesszustand und -ablauf angenommen werden können.*
  - Zufluss von Verarbeitungsgut:
    - ▶ *Der Zufluss von Verarbeitungsgut (Materie, Energie) erfolgt kontinuierlich oder in Chargen und beeinflusst dabei den Zustand des Prozesses.*
  
  - Stellgrößen:
    - ▶ *wirken mehr oder minder direkt auf die Prozesszustandsgrößen*
  
  - Störgrößen:
    - ▶ *treten meist als unerwünschter und oft schwer kontrollierbarer Zu- oder Abfluss von Verarbeitungsgut auf, oder sogar als Störung von Stell- und Messgrößen.*
  
- Ausgangsgrößen
  - ▶ *fließen aus dem Prozess heraus, dabei können sie aber auch auf den Prozess zurückwirken. ▶ stets abhängig vom aktuellen Prozesszustand ▶ Bezeichnung deshalb als abhängige Größen*
  - Abfluss von Verarbeitungsgut:
    - ▶ *hängt vom aktuellen Zustand des Prozesses und Stellgrößen ab. ▶ verändert i.d.R den Prozesszustand, oft wie Zufluss nur mit umgekehrtem Vorzeichen.*
  
  - Messgrößen:
    - ▶ *sollen die aktuellen Prozesszustandsgrößen unabhängig voneinander angeben, und sollen **keine** Rückwirkung auf den Prozess haben. Auch die Stellgrößen sollten keine direkten unmittelbaren Einwirkungen auf sie haben.*



- Alarme:
  - ▶ *spezielle Messgrößen signalisieren besondere wichtige Ereignisse, die durch bestimmte Prozesszustandsänderungen definiert sind. Beispiel: Überschreitung der zulässigen Kühlwassertemperatur*

• Prozessdaten untergliedern sich in:

- Ausgabedaten
- Eingabedaten
- Alarmer als Sonderform von Eingabedaten
- ► *Alle Daten können sowohl digital (wertdiskret) als auch analoger (wertkontinuierlich) Natur sein!!*

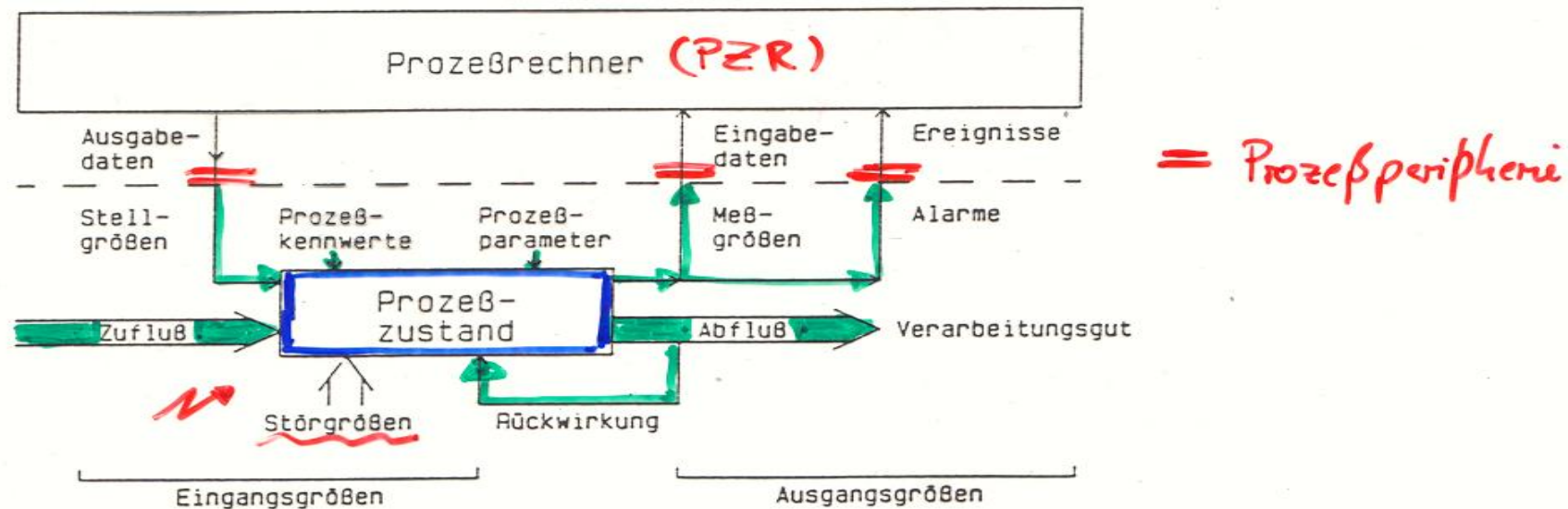


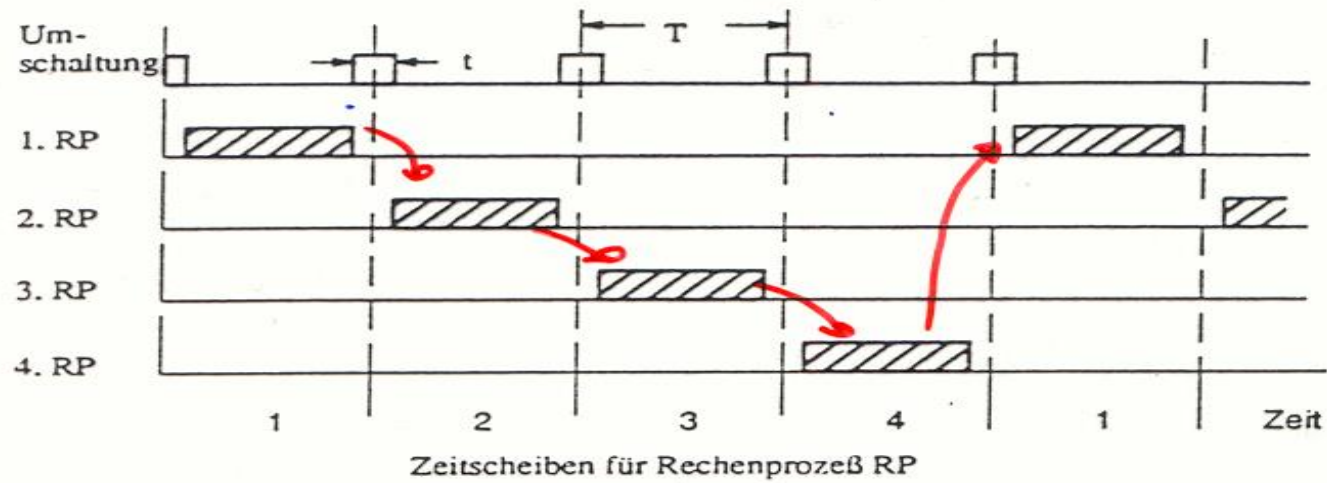
Bild 1.5 Der Prozesszustand wird durch die Eingangsgroßen bestimmt: durch die Stellgrößen, Störgrößen und durch den Zufluß von Verarbeitungsgut; aber auch der Abfluß von Verarbeitungsgut wirkt auf den Prozesszustand zurück.

Der Prozesszustand wiederum bestimmt die Ausgangsgroßen: den Abfluß von Verarbeitungsgut und die Meßgrößen; Alarme werden durch besondere Änderungen des Prozesszustands ausgelöst.

Die Ausgabedaten des Prozessrechners werden zu Eingangsgroßen des Prozesses; Eingabedaten werden aus den Ausgangsdaten des Prozesses gewonnen.

## 1.2.6 Rechenprozess, Task und Multitasking

- Die Software, die auf dem PR "läuft" ist für die Funktionalität bzw. Regelung verantwortlich ► hier spricht man genauer von **Rechenprozessen die die Aufgabe der Verarbeitung der erfassten Messgrößen aus dem technischen Prozess und dessen Beeinflussung haben ("was wie wann getan werden muss")**.
- **Rechenprozesse** haben die Umformung, Verarbeitung und den Transport von **Information** zum Gegenstand.
- Der Ablauf eines Programms in einem Rechner wird als Rechenprozess oder kurz als **Prozess oder Task** bezeichnet.
- ► Das **Programm** selbst ist in diesem Zusammenhang die **statische** Aufschreibung von Befehlen => der Algorithmus => die "Bedienungsanleitung" zum technischen Prozess
- ► Der **Rechenprozess (Task)** ist die **dynamische** Ausführung des Programms => die Ausführung der Bedienungsanleitung in Abhängigkeit der Anforderungen des zu bedienenden technischen Prozesses.
- Auf **einem** Rechner können, sofern die Rechnerleistung ausreicht, quasi gleichzeitig mehrere Rechenprozesse ausgeführt werden. Diese Betriebsart eines Rechners nennt man **Multitasking** (oder auch Multiprocessing).
- Mehrere Rechenprozesse können in **einem** Rechner **dasselbe** Programm benutzen (ihr Ablauf muss hierzu mehrmals angestoßen werden) allerdings benötigt **jeder Prozess seinen eigenen Datenbereich**.
- **Zu einer Zeit kann in einem Rechner** (Single-Prozessor-System) **nur ein Rechenprozess aktiv** sein. Die quasi gleichzeitige Abarbeitung aller Rechenprozesse wird durch eine Stückelung und Verteilung der verfügbaren Rechnerzeit auf die einzelnen Rechenprozesse erreicht.



kontextswitche

Abb. 1.1: Einfaches Modell für die Aufteilung eines Prozessors auf Rechenprozesse

## Kontextswitch und Overhead

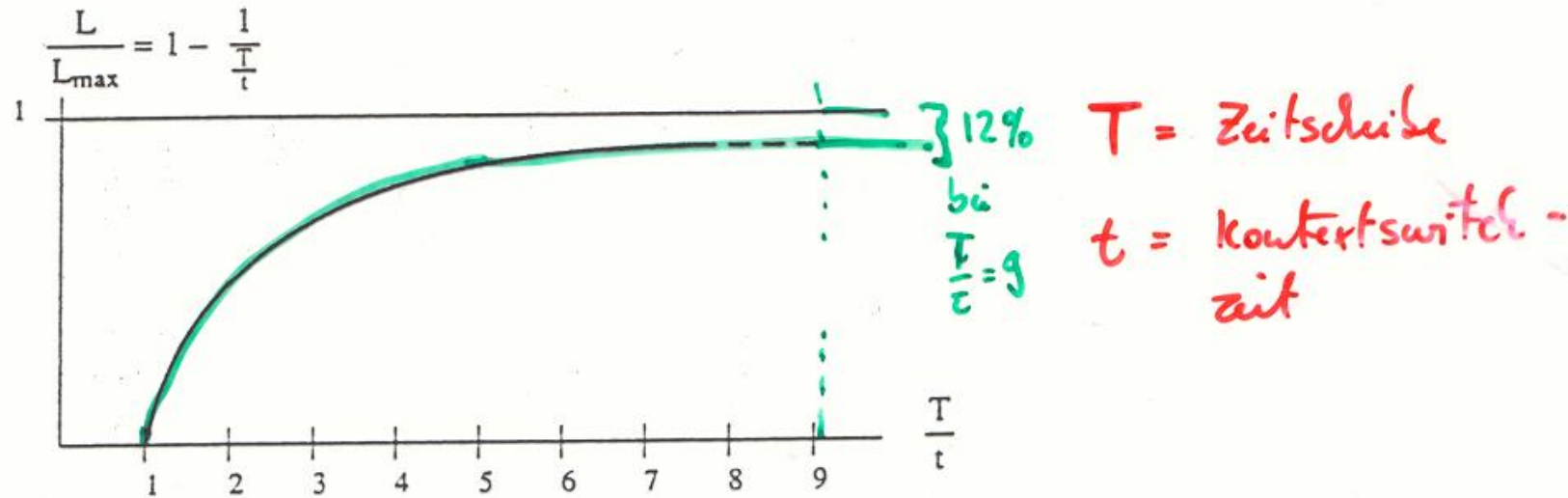


Abb. 1.2: Rechenleistungsverlust bei kleinen Zeitscheiben

$$L = L_{max} - \frac{t}{T} \cdot L_{max}$$

$$\downarrow$$
  
 Max. Rechenleistung

$$\underbrace{\quad}_{\text{"Verlust"}}$$

$t \downarrow$  "schnelle HW"  
 $T \uparrow$  ! Realzeitbdg!

$$\frac{L}{L_{max}} = 1 - \frac{t}{T}$$

$$\frac{L}{L_{max}} = 1 - \frac{1}{\frac{T}{t}}$$

z.B.  $\frac{T}{t} = 9$   
 $T_1 = 300 \mu s$   
 $T_2 = 180 \mu s$   
 $T_3 = 54 \mu s$

$t_1 = 100 \mu s$  (80C51)  
 $t_2 = 20 \mu s$  (68040)  
 $t_3 = 6 \mu s$  (PowerPC 501)

OS 30ms =  $T_1$

Der **Kontextswitch**, vom laufenden Prozess zum nächsten wartenden Prozess stellt einen Verbrauch von Rechnerzeit dar, der keinem Rechenprozess zugute kommt, sondern sogar die für die Anwendungen verfügbare Rechnerzeit reduziert, er ist also unerwünschter aber notwendiger **Overhead**.

*Der Overhead wird in % der Dauer einer Zeitscheibe angegeben.*

*Kontextswitchdauer  $t=0.1\text{ ms}$  , Zeitscheibe  $T=1\text{ms}$  Overhead=  $0.1 / 1.0 * 100\% = 10\%$*

- Rechenprozesse laufen sinnvoll nur mit Kommunikation zu anderen Prozessen ab.
- Rechenprozesse sind also kommunizierende "kommunizierende sequentielle Prozesse"  
    ▶ (CSP, Communicating Sequential Processes)".
- Die **Rechenprozesse** eines Prozessrechnersystems laufen
  - **auf einem zentralen Prozessrechner**
  - **auf mehreren Rechnern verteilt** (verteiltes Prozessrechnersystem)

## 1.2.7 Grundprinzipien der Rechner-Prozesskoppelung

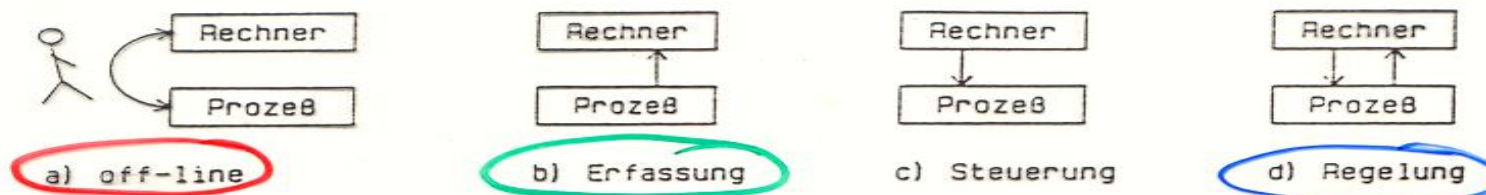


Bild 1.6 Kopplungsarten

- Im off-line-Betrieb transportiert der Mensch die Daten zwischen Rechner und Prozess.
- Bei der Erfassung von Prozessdaten dient der Rechner zur Aufbereitung der Daten für den Menschen, der dann regelnd in den Prozess eingreift.
- Bei der Steuerung eines Prozesses durch einen Rechner werden nur Stellwerte für den Prozess aufbereitet. Voraussetzung dafür ist ein deterministischer Prozess.
- Bei der Regelung eines Prozesses durch einen Rechner werden die aktuellen Prozessdaten erfasst, mit den vorgegebenen Sollwerten verglichen, Stellwerte berechnet und zum Prozess ausgegeben. Der Mensch kann sich auf die Beobachtung des Prozesses beschränken, und greift nur im Ausnahmefall ein.

- **offline Betrieb** (heute selten) ► indirekt gekoppelter Betrieb. ► Der Mensch transportiert die Daten zwischen Prozess und Rechner
- **online Betrieb (Stand der Technik)** ► direkt gekoppelter Betrieb Rechner und Prozess sind über Prozeßperipherie gekoppelt.
  - **offener Betrieb**
    - **Datenerfassung** one-way vom Prozess zum PR
    - **Datenausgabe, Steuerung** one-way vom PR zum Prozess
  - **closed-loop Betrieb**
    - **Regelung (Stand der Technik)** geschlossenes digitales Regelsystem, bei dem der PR den Prozess steuert in Abhängigkeit der Messgrößen und Vorgaben.

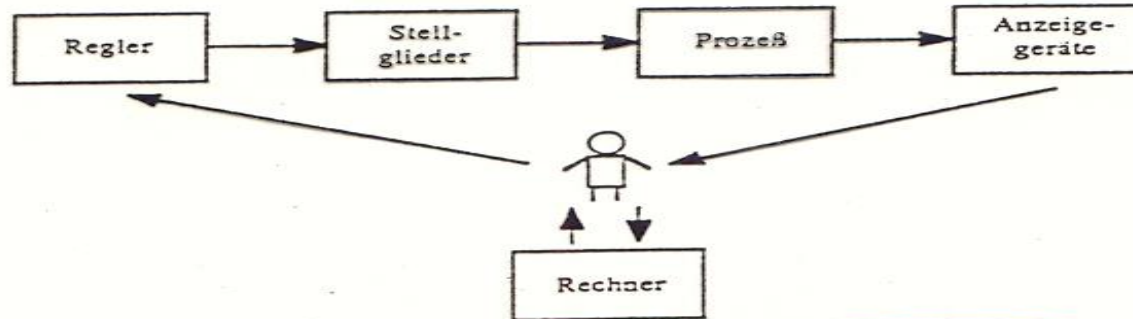
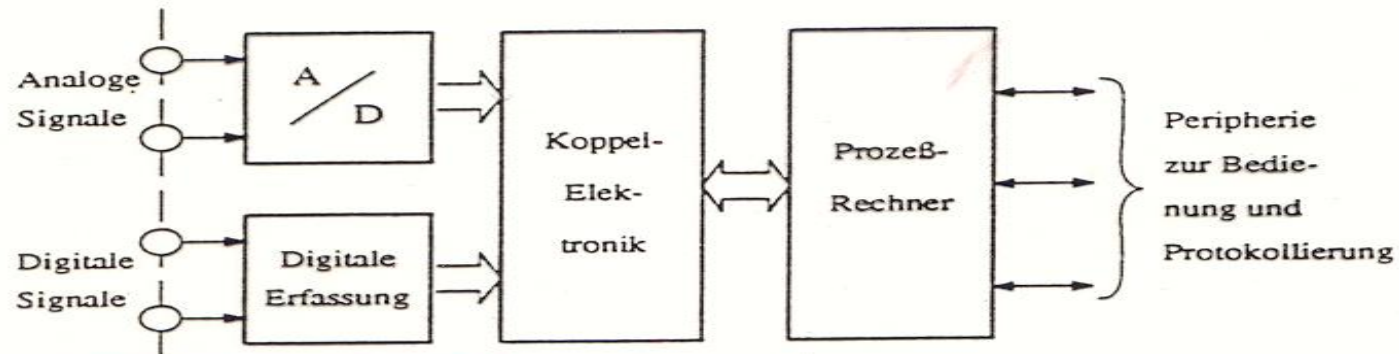
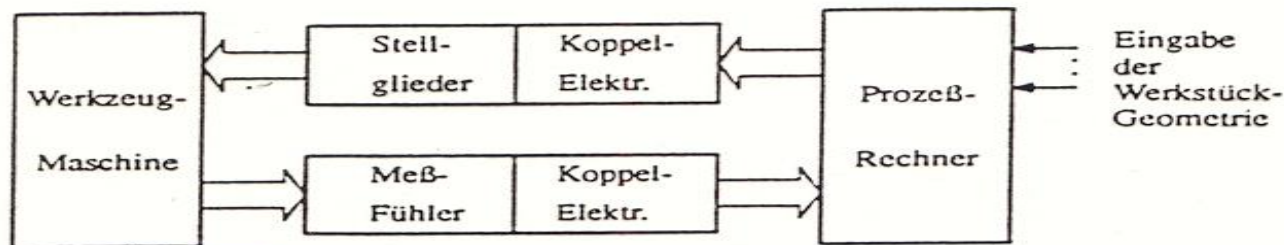


Abbildung 1.15: Indirekte **Off-Line-Rechner-Prozeß-Kopplung** über Datenträger



**Abb. 1.5: Online-Datenerfassung**



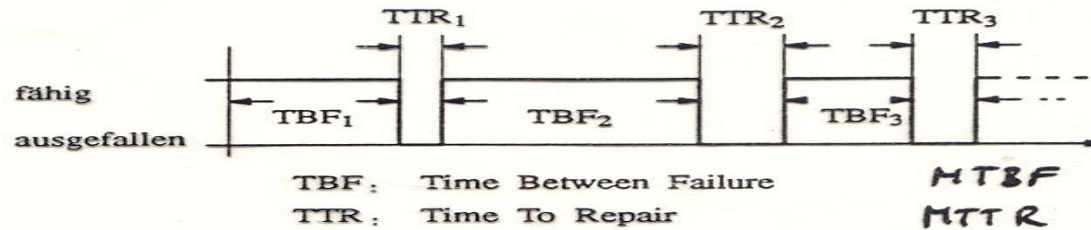
**Abb. 1.6: Closed-Loop-Betrieb**



## 1.2.8 Zuverlässigkeit und Sicherheit

- **Zuverlässigkeit:** Unter Zuverlässigkeit einer technischen Anlage versteht man die Fähigkeit des Systems unter vorgegebener Zeitdauer und zulässigen Betriebsbedingungen die spezifizierte Funktion zu erbringen.
- **Sicherheit:** Die Sicherheit ist das Nichtvorhandensein einer Gefahr für den Menschen, die Umwelt und den Sachwert. Oder „weicher“: “Sicherheit ist eine Sachlage, bei der das Risiko nicht größer als das Grenzkrisiko ist.” [DIN/VDE 31000 Teil2] Als Grenzkrisiko ist dabei das größte, noch vertretbare Risiko zu verstehen [Hal99].
- Die Zuverlässigkeit wird i.d.R. beeinträchtigt durch **Fehler** und gegebenenfalls **Funktionsausfall**.
- An die Zuverlässigkeit von Prozessrechnern werden besondere Anforderungen gestellt: Prozessrechner müssen oft im 7-Tage-24-Stunden-Betrieb funktionsfähig sein.
- ► Funktionsausfall kann dazu führen:
  - damit die gesamte Produktionsanlage ausfällt, womit sehr hohe Kosten verbunden sein können,
  - sich die Produkt-Qualität verschlechtert (oder der Energie- bzw. Rohstoff-Bedarf),
  - hohe Kosten bei der Wiederinbetriebnahme entstehen.
- Es wird immer wirtschaftlicher, hochzuverlässige Prozessrechner einzusetzen, ggf. fehlertolerante Systeme, welche auch beim Ausfall einzelner Komponenten ihre Funktionen weiter erfüllen.
- Es gibt auch sicherheitskritische Anwendungen von Prozessrechnern, deren unkontrollierter Ausfall die Sicherheit von Menschen gefährdet.

## Verfügbarkeit (1)



MTBF Mean TBF  
 MTTR Mean TTR  
 (inkl. Fehlererkennung!)

Abb. 2.69: Zeitverlauf des Systemzustands

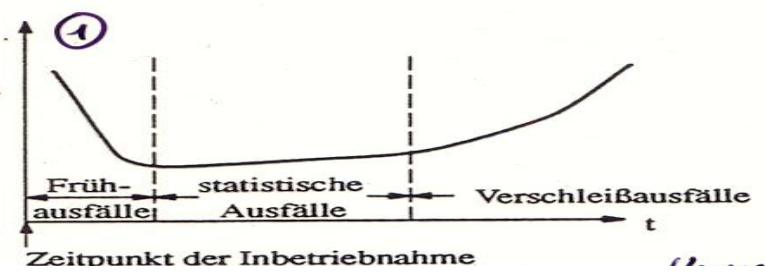
Verfügbarkeit  $p = \frac{MTBF}{MTTR + MTBF}$  und  
 Unverfügbarkeit  $q = \frac{MTTR}{MTTR + MTBF}$   
 mit  $p + q = 1$

stationärer Fall  
 $p, q \neq f(t)$

- $\lambda = \frac{1}{MTBF}$  Ausfallrate
- $r = \frac{1}{MTTR}$  Reparaturrate

### Nicht stationärer Fall:

q(t)  
 Unverfügbarkeit



① MTTR ↑ "lange Reprzeit"  
 "Neue Anlage"  
 MTBF ↓ "Früh ausfälle"

Abb 2.71 Zeitverlauf der Unverfügbarkeit q(t) „Badewannenkurve“

## Verfügbarkeit (2)

- Annahme: Einzelkomponenten voneinander statistisch unabhängig
- ? Gesamtverfügbarkeit eines aus Einzelkomponenten zusammengesetztes System

### a) Serienschaltung



$K_i$ : Einzelkomponente

$P_i$ : Verfügbarkeit der Einzelkomponente  $K_i$

$q_i$ : Unverfügbarkeit  $K_i$

$$P_{\text{gesamt}} = \prod_{i=1}^n P_i$$

$$\lambda_{\text{gesamt}} = \sum_{i=1}^n \lambda_i$$



$$MTBF_{\text{gesamt}} = \frac{1}{\sum_{i=1}^n \frac{1}{MTBF_i}}$$

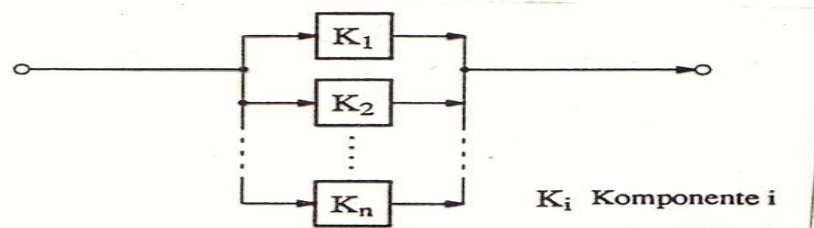
$P_i \approx 1!$

$q_i \ll 10^{-2}$

$$P_{\text{gesamt}} = 1 - q_{\text{gesamt}} = \prod_{i=1}^n (1 - q_i) \approx 1 - \sum_{i=1}^n q_i$$

oder:  $q_{\text{gesamt}} \approx \sum_{i=1}^n q_i = n \cdot q_i!$  "Serie"

### b) Parallelschaltung



$q_{\text{gesamt}} = \prod_{i=1}^n q_i!$  "parallel"

$P_{\text{gesamt}} = 1 - q_{\text{gesamt}} = 1 - \prod_{i=1}^n q_i$

## Fehlertolerante Prozeßrechnersysteme (PZR) zur Erhöhung der Verfügbarkeit (1)

- Ein Fehler soll hier verstanden werden als: die Abweichung des Systems oder Teile des Systems vom gewünschten -geplanten- Verhalten
  - ==> Konsequenz: Zuverlässigkeit der Anlage wird negativ beeinträchtigt
  - ==> 1. Wunsch: Fehlervermeidung ('Perfektion') wenn möglich
  - ==> 2. Wunsch: **Fehlertoleranz** ('Redundanz') wenn 1.Wunsch nicht erfüllbar
- **Fehlertolerant** bedeutet, dass das PZR seine Funktion, den zu steuerenden technischen Prozeß zu bedienen, weiterhin erfüllt, obwohl ein Fehler aufgetreten ist. Voraussetzungen:
  1. **Ein Fehler wird sicher als solcher erkannt (Fehlerdiagnose,Fehlererkennung)**
  2. **Es ist Redundanz (==>Ersatz) vorhanden**, um den Fehler zu kompensieren ==> Vergleiche 'Parallelschaltung' im Zuverlässigkeitsersatzschaltbild
  3. In Folge einer Fehlererkennung kompensiert der Ersatz den erkannten Fehler, durch Erbringung der gewünschten (Teil-)Funktion

- **Fehlertypen:**
  - **permanente Fehler** ==> Fehler tritt auf und bleibt dauernd bestehen (gut zu finden!)
  - **transiente Fehler** ==> Fehler tritt auf und verschwindet wieder (10 Mal häufiger als permanente Fehler, oft schwierig zu finden!)
- Kategorien von **Fehlern:**
  - Entwurfsfehler
    - Spezifikationsfehler
    - Implementierungsfehler
    - Dokumentationsfehler
    - ==> Minimierung durch Entwicklung sog. diversitären Lösungen (2 EW-Teams)
  - Herstellung- oder physikalische Fehler (Altern, Verschleiß, Überlastung....)
    - ==> Minimierung von Herstellungsfehlern durch Qualitätssicherung
  - Betriebs- und Bedienfehler
    - (unvorhergesehene) Störungen der Prozeßumwelt (EMV, mechanische thermische Einflüsse...)
    - Bedienfehler (menschliches Versagen)
    - Wartungsfehler
    - ==> Minimierung der Bedienfehler durch intensives Training des Personals mittels 'Procedures'

## Fehlertolerante Prozeßrechnersysteme (PZR) zur Erhöhung der Verfügbarkeit (2)

- **Die zu erkennenden Fehler und die daraus resultierenden Verfahren zur Erkennung** derselben werden zur Entwurfszeit des PZR definiert ==> Erstellung eines **Diagnosemodells (FMEA Failure Mode + Effect Analysis)**
- Die **Fehlertoleranz** wird ebenfalls zur Entwurfszeit des PZR eingebaut.
- **Fehlererkennung durch:**
  - zyklische Hard- und/oder Softwaretests
  - Hardware-Voter ( 2 aus 3 , 3 aus 5) ==> Probleme: Verbindungsleitungen, Hardware des Voters muss auch redundant sein
  - Software-Voter: Ergebnisvergleich nach redundanter Berechnung derselben Aufgabe in unterschiedlichen Softwareprozessen auf dem gleichen Rechner oder unterschiedlichen Rechnern
  - Assertions (Zusicherungen): Bei vorab bekannten Zwischenergebnissen ein Vergleich mit den aktuell berechneten Ergebnissen

- Arten der **Redundanz**:
  - dynamische Redundanz: Der Ersatz übernimmt erst dann die Funktion wenn ein Fehler aufgetreten ist, ansonsten steht er für andere Aufgaben zu Verfügung (==> z.B. Entwicklungsrechner...)
    - 1. Problem: sichere Fehlererkennung (==> Assertions, Voter nicht einsetzbar)
    - 2. Problem: Einphasen des Ersatzsystems
      - ==>Backward-Recovery-Systeme: Es gibt definierte Checkpoints in definierten zeitlichen Abständen an denen gewiss ist, dass das System i.O. ist. An diesen Checkpoints wird der gesamte interne Zustand so gesichert, dass der zuletzt erreichte Checkpoint im Fehlerfall als Anlaufzustand des Ersatzsystems dient.
  - statische Redundanz: Es gibt quasi keinen Unterschied zwischen Ersatz und Hauptsystem. Beide bearbeiten parallel dasselbe:
    - räumlich parallel ==> mehrfach vorhandenen Hardware
      - ==> SRU-Konzept (smallest replaceable unit) Wenn ein Fehler erkannt wird, dann wird die betroffene defekte Hardware (manuell) durch Wechseln der betreffenden SRU ersetzt. (SRU kann sein: Bauteil,Subsystem,System...)
    - zeitlich sequentiell ==> derselbe Bearbeitungsablauf wird zeitlich hintereinander -in Hardware oder/und Software- wiederholt

## Sicherheitsaspekte

- **Vor der Verfügbarkeit geht die Sicherheit der Anlage**
  - Wird ein sicherheitskritischer Fehler erkannt ==> das System muss -falls vorhanden- in den "fail-safe" Zustand gebracht werden (=Sicherheitsbetrieb) ==> Reduktion der Verfügbarkeit des Gesamtsystems denn:
  - Fehlererkennung ist nur bei vorhandener Redundanz möglich ==> mehr Hardware, mehr Fehlerquellen, höhere Unverfügbarkeit insgesamt
  - Wie beeinflussen Sicherheitsüberlegungen die Verfügbarkeit der Anlage?
  - Gegeben: 3-fach redundante Hardware  $q_{hw} \ll 10^{-4}$ , idealer Voter ( $q=0$ ).
1. **Sicherheitsbetrieb:** Tritt ein Fehler auf -d.h. eine Hardware liefert ein anderes Ergebnis- wird die Anlage in den 'fail-safe' Zustand gebracht

Unverfügbarkeit:  $q_{ges} = 1 - p_{ges} = 1 - (1 - q_{hw})^3$  ; mit ( $q_{hw} \ll 10^{-4}$ )  $\approx 3 \cdot q_{hw}$  (Sobald einer ausfällt, fällt das Gesamtsystem aus)

Wahrscheinlichkeit eines unsicheren Zustands (unerkannter Fehler)  $W_a$ : Alle 3 Hardwaren fallen zeitgleich aus **und** produzieren den gleichen Fehler ==> Voter findet keine Abweichung trotz Fehler ==> Betrieb läuft fälschlicherweise weiter.

$$W_{as} \leq q^3 \quad q_{hw} \approx 10^{-4} \quad W_{as} \approx 10^{-12}$$



## 2. TMR-Betrieb (Triple Modular Redundancy) : Mehrheitsentscheidung 2-aus-3

Unverfügbarkeit:  $q_{\text{ges}} = q^3 + 3 * (q^2 * (1-q))$   
 $q^3$       Wahrscheinlichkeit alle ausgefallen  
 $q^2 (1-q)$  Wahrscheinlichkeit zwei ausgefallen, einer funktioniert

$W_a$ : 3 oder 2 Hardwaren ausgefallen und liefern gleiches falsches Ergebnis ==> dann bleibt der Fehler unerkannt und ggf. die einzig fehlerfrei arbeitende Hardware wird weggeschaltet.

$$W_{\text{atmr}} \leq q^3 + 3 * (q^2 * (1-q)) ; \quad W_{\text{atmr}} = 10^{-12} + 3 * 10^{-8} * 0.9999 \approx 10^{-7}$$

## 3. Höchste Verfügbarkeit: Solange noch eine Hardware funktioniert wird weitergemacht.

Unverfügbarkeit:  $q_{\text{ges}} = q_{\text{hw}}^3$  (Erst wenn alle ausgefallen sind, fällt das Gesamtsystem aus)

$W_{\text{av}}$ : Der Voter entscheidet sich für den fehlerhaften Rechner als richtigen Rechner wenn ein Fehler auftritt.

$$W_{\text{av}} \leq 3*q$$

### **1.3 Prozessbeschreibung (-modelle) und Systembetrachtung**

- Um einen Prozess steuern bzw. regeln zu können wird ► die **Prozessbeschreibung** und das **Prozessmodell** benötigt
- Ein Prozessmodell wird erstellt, um folgendes aufzuzeigen und zu überprüfen:
  - Prozessstruktur
  - Zusammenhänge: Eingangsdaten, Ausgangsdaten und Zustandsdaten
- Mögliche Modellformen:
  - Gegenständliches Modell (im verkleinerten Maßstab oder in Zeitlupe)
  - abstrakten oder mathematischen Modell
  - Rechnermodell: Simulation mit einem Computer

### 1.3.1 Prozessbeschreibung

- **Vorgang der Erstellung: Strukturierung** des Prozesses ► **Zerlegung** in Teilprozesse ► Verfeinerung bis zu **Elementarprozessen** (► entspricht der strukturierten Programmierung)
- ► **Ergebnis: Verteilung** der **Beschreibung** des Gesamtprozesses **auf die Elementarprozesse selbst und deren Zusammenhänge** untereinander.
- **Elementarprozesse** werden meist **mittels mathematischer** Modelle beschrieben: Gleichungen über technische, chemische oder physikalische Vorgänge.
- **Prozessstrukturen** werden meist **durch graphische Methoden** beschrieben:
  - **statische Strukturen** des Prozesses durch **Blockschaltbilder**
  - **stationäre Abläufe des Prozesses**
    - meist im Blockschaltbilder als Wirkungs- oder Transportwege eingezeichnet.
    - Flussplan, Struktogramm !! Unterschied zu Programmierung: Parallelität der Aktionen !!
    - Netzpläne (DIN69 900), Netze aus Instanzen und Kanälen (später)
  - **dynamische Abläufe**
    - Zustandsdiagramme ► Automobilbau "Statemate, Rhapsody, Matlab/Simulink"
    - PETRI-Netze, ► UML 2.x (Unified Modelling Language) ► SDL Structure and Description Language ► Telekommunikation

Zu beschreibende Zusammenhänge: Ein- und Ausschaltvorgänge, zeitlichen Veränderungen und Zusammenhänge beim Arbeitspunktwechsel (kann auch durch plötzliche innere oder äußere Störgrößenaufschaltung geschehen!!)

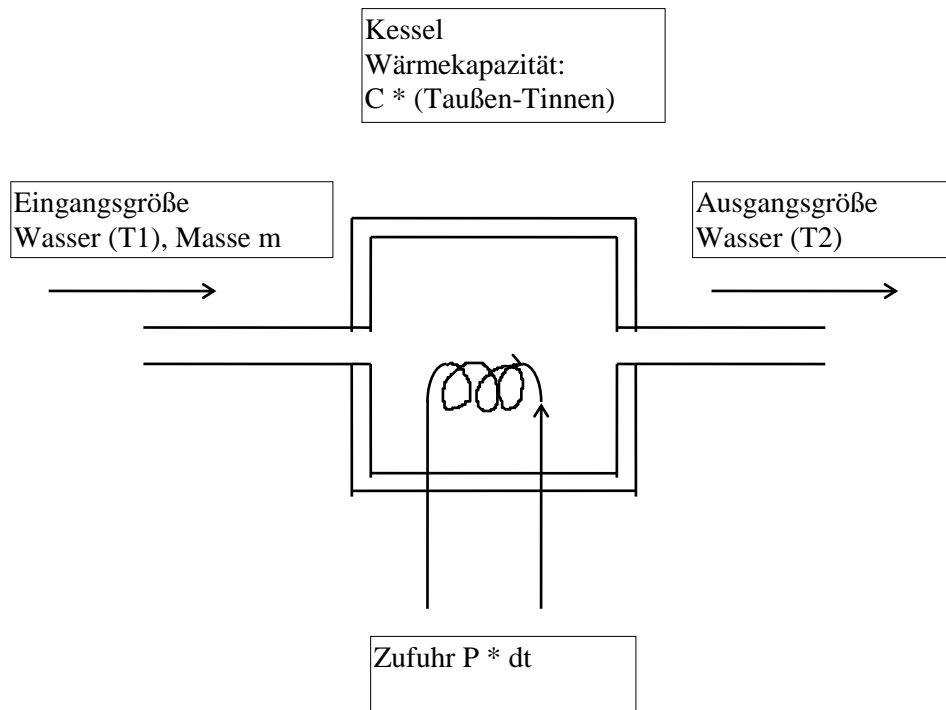
### 1.3.2 Prozesserkennung (Modell-Bildung)

- Die Erstellung eines (abstrakten) Modells bezeichnet man als **Prozesserkennung**
- Zwei unterschiedliche Methoden:
  - **empirische Methode**  
Gegeben i.d.R. der fertig realisierte technische Prozess, der **nachträglich** automatisiert werden soll  
Die Prozessbeschreibung erfolgt **empirisch** an Hand der Erfahrung.  
(=> bei Verwendung der Prozessdaten zur Ermittlung der Anlagedaten spricht man von **Prozessidentifikation**)
  - **analytische Methode**  
Ablauf wird durch Gleichungen beschrieben =>Automatentheorie  
Ergebnis: analytisches Prozessmodell. Alle Prozessgrößen analytisch zu jedem Zeitpunkt berechenbar.
- **Adaptive Prozessmodelle:**
  - Im Modell werden die unbekannt Prozessgrößen (insbesondere Störgrößen) im laufenden Betrieb erfasst und als anpassbare Parameter in die Berechnungen einbezogen werden.
  - Problem: =>unbekannte Prozessgrößen schlecht vorhersagbar aber die Kompensation leicht im Modell zu beschreiben.

*Beispiel: automatische Spracherkennung. Aufgabe sprecherunabhängig Sprache in Schriftzeichen zu übersetzen.  
Problem: Das Erkennen basiert auf Mustervergleichsmethoden. Die vorgegebenen Muster sind die eines Sprechers.*

- **lernende Prozessmodelle** bezeichnet man als solche:
  - => die in lernende Programme eingebaut sind, die im Betrieb nicht nur passiv die Parameter auf Grund von Prozessdaten bestimmen, sondern die optimalen Werte aktiv selber suchen.
  - => Diese Art von Modellbildung gehört in das Gebiet der künstlichen Intelligenz und ist im praktischen Betrieb sehr selten zu finden.
- Die Auswahl der Methode zur Erstellung eines Modells richtet sich nach dem Prozess und der Eignung für die jeweilige Prozessbeschreibungsebene.

• **Beispiel:**



Prozesserkennung:

1. Elementarprozeß, Aufheizen von Wasser durch elektrische Energie;

1a. Ablauf:

- ▶ Kessel wird gefüllt, kein Abfluss
- ▶ heizen, Eingang- und Ausgang geschlossen
- ▶ Kessel wird geleert kein Zufluss.

2. Merkmale: Umformung von Energie, als Chargenprozess, deterministisch;

3. Prozesserkennung:

empirisch: "Zufuhr von elektrischer Energie erhöht die Temperatur." (vereinfacht!!)

analytisch:  $W = c * m * (T_{ende} - T_{anfang}) = P * t [J]$

Prozessmodell, hier: statisches Modell mit Prozessgrößen:

Eingangsgrößen: Wassermenge ( $m [kg]$ ), Wassertemperatur ( $T1$ ), Elektrische Leistung ( $P$ ) Heizzeit ( $dt$ );

Parameter: spezifische Wärme von Wasser ( $c$ )  $4,18 [kJ/(kg \cdot K)]$

Störgrößen: Wärmekapazität des Behälters ( $C$ ) und dessen Temperatur ( $Tk$ ) ▶ Wärmeverluste durch Wärmeleitung und -strahlung des Behälters

Prozess-Zustandsgrößen: Temperatur  $T$  des Wassers im Behälter

Prozess-Ausgangsgrößen: Wassertemperatur ( $T2$ )

- stationäres Modell (ohne Wärmeverluste):

$$\text{aus } P \cdot dt = c \cdot m \cdot (T2 - T1) + C \cdot (T2 - Tk)$$

$$\text{folgt } T2 = (P \cdot dt + c \cdot m \cdot T1 + C \cdot Tk) / (c \cdot m + C)$$

- dynamisches Modell:

*Es beschreibt die Ausbreitung der Wärmeenergie und der Temperaturverteilung im Verlaufe der Zeit innerhalb des Behälters. Dazu müssen Wärmeleitung und -konvektion durch Differentialgleichungen höherer Ordnung berechnet werden. => hier nicht!*

- *Fazit Beispiel: Großer Aufwand, der für die Erstellung eines zuverlässigen Prozessmodells*
- In der Praxis ist der Normalfall dass das neu erstellte Modell noch unzulänglich ist und das Prozessverhalten nur unzureichend beschrieben wird.
  - ► häufige Ursache: Störgrößen nicht genügend genau bekannt oder unvollständig beschrieben.
  - ► Folge: iterative Verbesserung des Modells zu jeder Zeit:
- Standard-Vorgehen : analytische, numerische Untersuchungen oder Simulation am Computermodell der Auswirkung von kleinen Änderungen der Eingangsgrößen auf den Prozess.
  - => Problem: Die unvermeidlichen Vereinfachungen durch das eingesetzte Modell verdecken u.U. Prozesseffekte die dann im Betrieb auftreten und gravierende Auswirkungen auf den Prozess haben können
  - => wie minimiert man das Problem in den Fällen in denen keine Iteration am realen Prozess erfolgen kann?: Verbesserung auf Grund eines Vergleichs eines gegenständlichen Modells und einer Computermodell-Simulation -> dadurch sind 2 verschiedene Modelle vorhanden
  - => Annahme dabei: verschiedene Verfahren werden auch verschiedene Abweichungen ergeben

## Inhaltsverzeichnis Kapitel PDV K

K 1 Einführung in die Prozessdatenverarbeitung(PDV) und Begriffe.....	1
1.1 Einführung und Geschichte der PDV .....	1
1.1.1 Ziel der Prozessdatentechnik (PDV)-Vorlesung, oder was ist Prozessdatentechnik? .....	7
1.1.2 Allgemein: Automatisierung von Prozessen .....	9
1.1.3 Ziele der Automatisierung in der PDV .....	11
1.2 Begriffe und deren Bedeutung aus der PDV .....	13
1.2.1 Prozess allgemein, Unterscheidung nach Verarbeitungsart und Verarbeitungsgut .....	13
1.2.2 Der technische Prozess .....	16
1.2.3 Klassifikationsmerkmale von technischen Prozessen .....	17
1.2.3.1 Unterscheidung nach Verarbeitungsstrukturen .....	17
1.2.3.2 Unterscheidung nach der inneren Ablaufstruktur .....	19
1.2.3.3 Unterscheidung nach Einsatzgebiet .....	19
1.2.4 Prozessrechner (PR).....	20
1.2.5 Prozessdaten und Prozessgrößen.....	22
1.2.6 Rechenprozess, Task und Multitasking .....	27
1.2.7 Grundprinzipien der Rechner-Prozesskoppelung.....	31
1.2.8 Zuverlässigkeit und Sicherheit.....	33
1.3 Prozessbeschreibung (-modelle) und Systembetrachtung .....	42
1.3.1 Prozessbeschreibung.....	43
1.3.2 Prozesserkennung (Modell-Bildung) .....	44