2. Begriffe des Software Engineering

Dr.-Ing. Clemens Reichmann

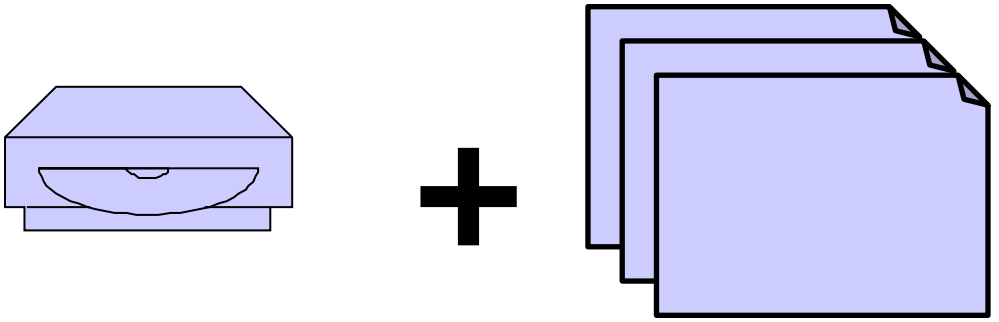
Clemens.Reichmann@vector.com,
Tel. 0721 / 91430-200

Institut für Technik der Informationsverarbeitung
Fakultät für Elektrotechnik & Informationstechnik
Universität Karlsruhe (TH)

Inhalt Kapitel 2

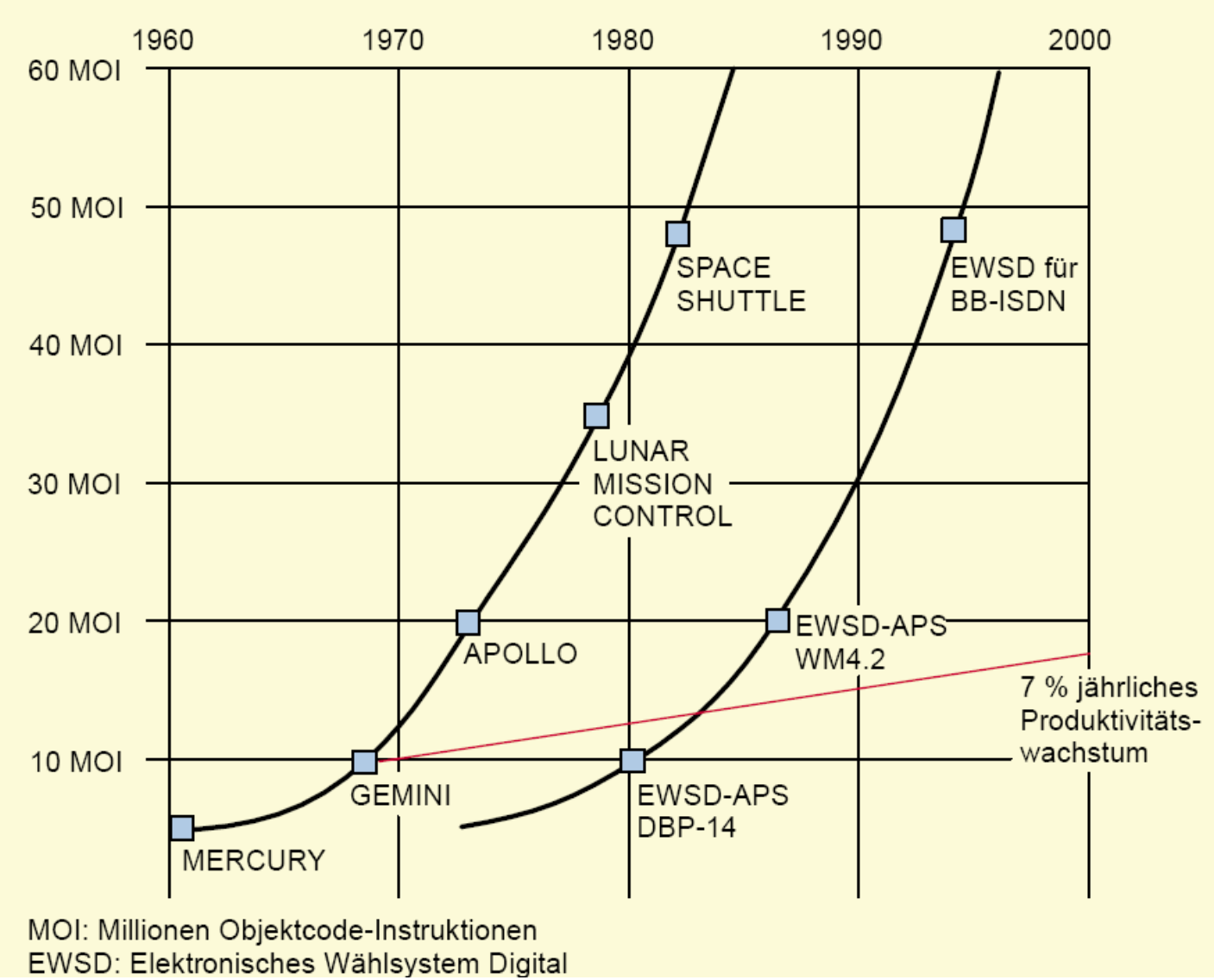
2.1 Begriff: Software

- Software:
  - Computerprogramm und zugehörige Dokumentation.
  - Softwareprogramme können für einen speziellen Kunden oder für den allgemeinen Markt entwickelt werden.
  - Software wird in einer Programmiersprache implementiert



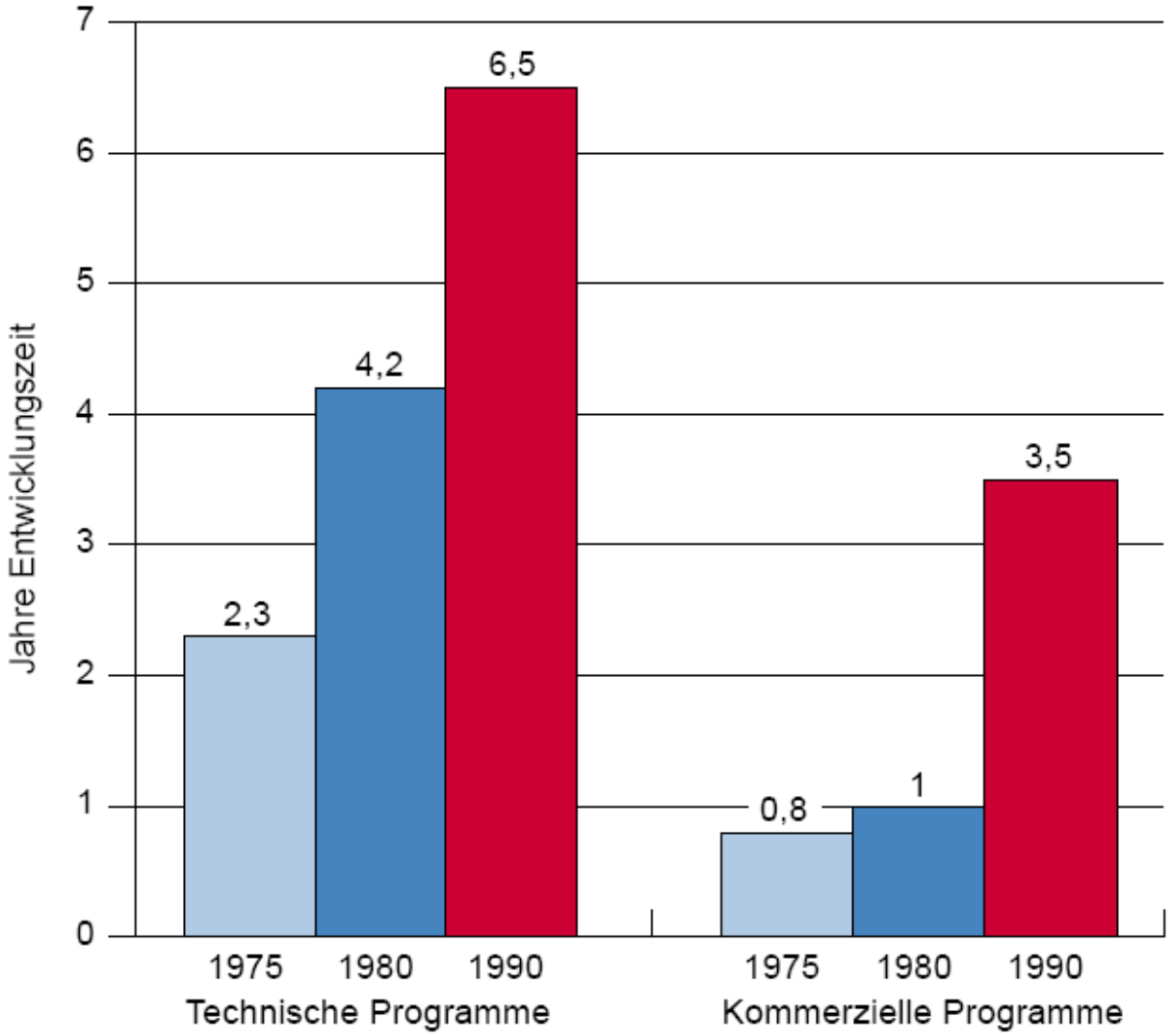
Warum ist Software so schwer zu entwickeln?

- Anzahl der Programmzeilen (Programmgröße)

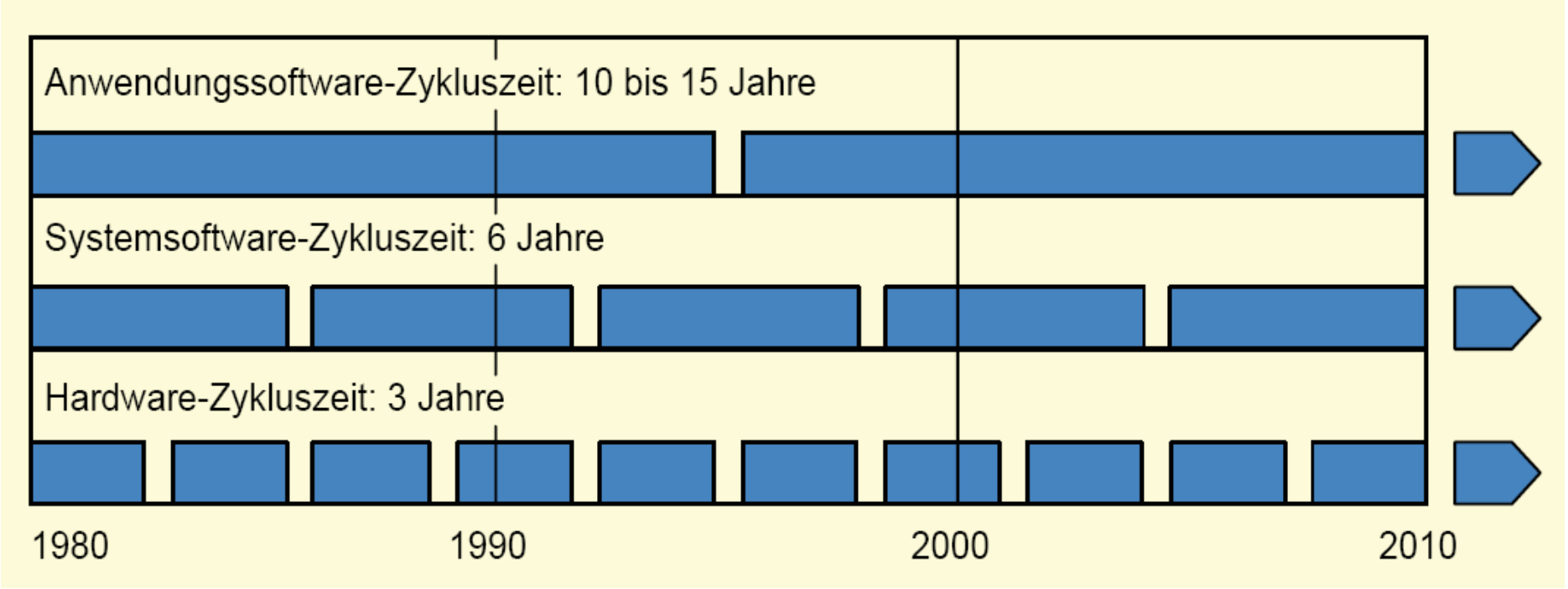


Warum ist Software so schwer zu entwickeln?

- Zunehmende Entwicklungsdauer



Probleme marktreifer Software

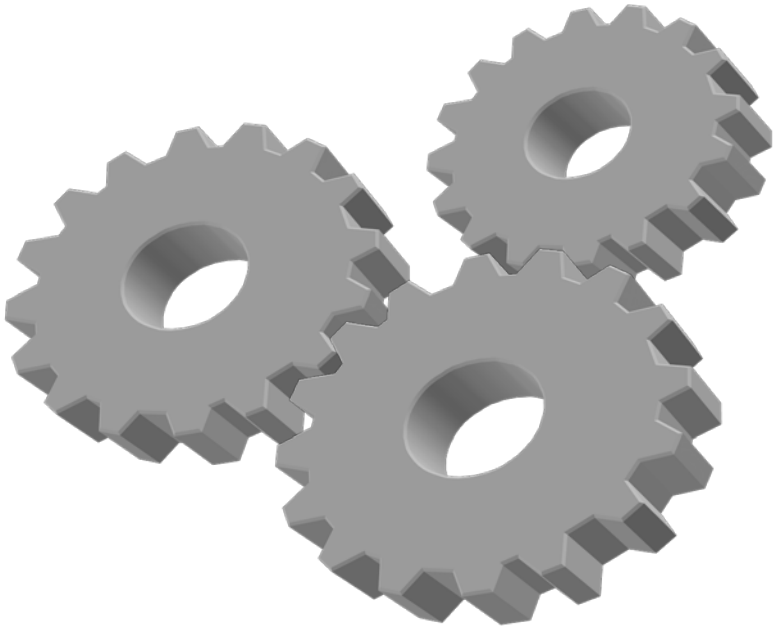


Merkmale guter Software

- Wartbarkeit
  - Weiterentwicklung: Evolutionäres Vorgehen
  - Weiterentwicklung wegen veränderter Kundenbedürfnisse
- Zuverlässigkeit
  - Verlässlichkeit
  - Zugriffsschutz
  - Betriebssicherheit
- Effizienz
  - Ressourcen nicht verschwenderisch nutzen
    - Speicher (RAM, Festplatte)
    - Prozessorkapazität (Reaktionszeit, Verarbeitungszeit)
- Benutzerfreundlichkeit
  - Nutzbarkeit, Bedienbarkeit
  - Dokumentation

Begriff: Informatik

- Informatik:
  - Die Informatik beschäftigt sich mit der Theorie und den Grundlagen.



Probleme und Begriffe

Der Begriff der **„Softwarekrise“** etabliert sich seit den 60ern. Besonders **spektakuläre Fälle** aus den letzten Jahrzehnten:

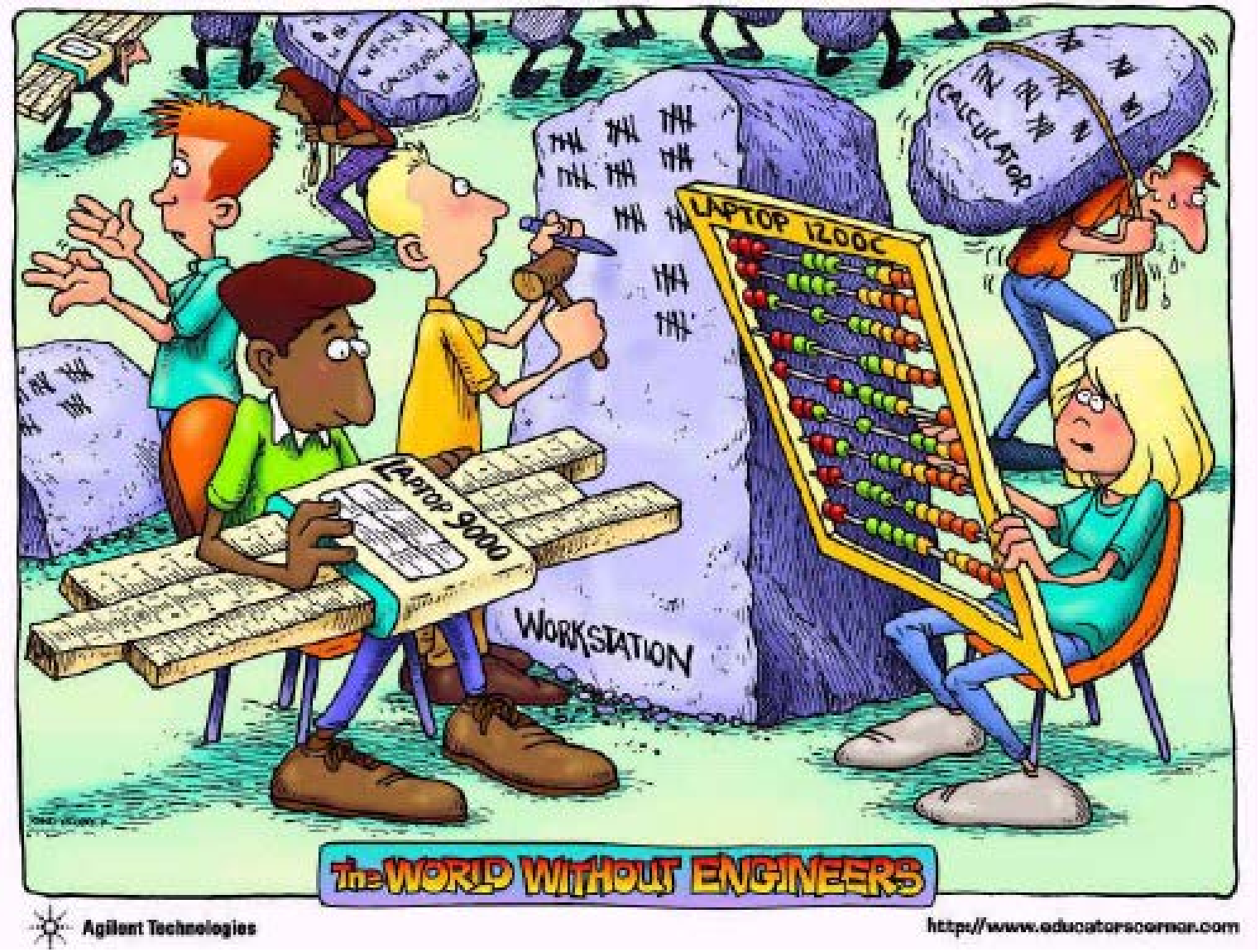
- 1979: **Studie zu Softwareprojekten** (USA), insges. ca. ***7 Mio US-$***.
  - 75% der Ergebnisse nie eingesetzt;
  - 19% der Ergebnisse stark überarbeitet;
  - ***6% benutzbar***.
- 1981: **US Air Force Command & Control Software** überschreitet Kostenvoranschlag fast um den Faktor 10 (***3,2 Mio US-$***).
- 1984: **Überschwemmungskatastrophe** im französischen Tarntal, weil Steuercomputer nach Überlaufwarnung Schleusen öffnet.
- 1996: Absturz der **Ariane 5** wegen eines Software-Fehlers.
- 1997: Entwicklung des **Informationssystems SACSS** für den Staat Kalifornien abgebrochen. Aufgelaufene Kosten: ***300 Mio US-$*** (200 % des Voranschlags).
- Mehrfach: Steuerungsprobleme beim **Airbus**.

Fragen ?

2.2 Begriff: Software Engineering

- Software Engineering:
  - Das Software Engineering ist eine technische Disziplin, die sich mit allen Aspekten der Softwareherstellung befasst.
  - Ist Teil der Systementwicklung
  - Der Begriff wurde 1968 auf einer Konferenz vorgeschlagen. Anlass: SW Krise durch Komplexität d.h. man konnte keine SW entwickeln in einer Komplexität, welche vorhandene HW ausreizte:
    - ***„Software Engineering“*** ist die Entdeckung und Anwendung solider ***Ingenieur-Prinzipien*** mit dem Ziel, auf wirtschaftliche Weise Software zu bekommen, die zuverlässig ist und auf realen Rechnern läuft.
      (F.L. Bauer, NATO-Konferenz Software-Engineering 1968)
  - Teilbereiche:
    - Spezifikation
    - Entwicklung
    - Management
    - Weiterentwicklung

Engineering ….



- ***„Software Engineering“*** ist die Herstellung einer Software, wobei **mehrere Personen** beteiligt sind und / oder **mehrere Versionen** entstehen. (D.L. Parnas)

Ursachen für Komplexität

- ***„Software Engineering“*** is the application of a systematic, diciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the **application of engineering to software** ...
(IEEE Standard 610.12)

systematisches Vorgehen !

systematisches Vorgehen …

„Ingenieur-typisches“ Vorgehen (1)

a) bei der Produktentwicklung:

- ***systematisch*** nach bekannten **Methoden** vorgehen (s.u.):
  - Entwicklungsprozesse für bekannte Situationen
  - Entwicklungsprozesse für Neuentwicklungen
- dabei Einsatz geeigneter **Werkzeuge**
- **Zerlegung** komplexer Gesamtsysteme in handhabbare Subsysteme und Komponenten
- ***Komponenten****:* abgeschlossene Bausteine mit fester Funktionalität, die sich zusammensetzen lassen
- ***normierte Komponenten*** sind leichter (wieder-) verwendbar: man denke z.B. an Schrauben, IC, Stecker, Dichtungsringe,…

„Ingenieur-typisches“ Vorgehen (2)

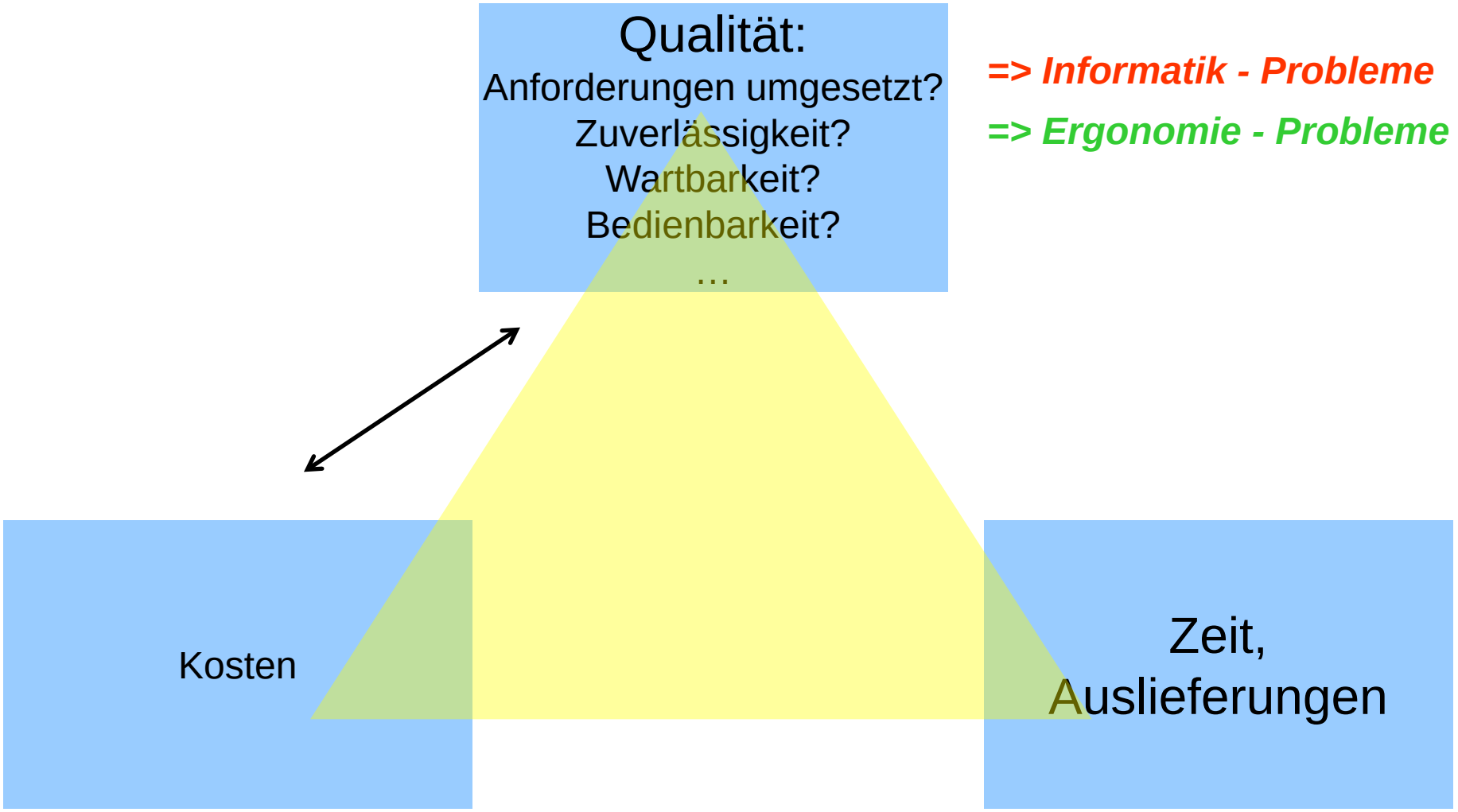
b) bei der Gestaltung des Entwicklungsprozesses:

- strukturiert in **Phasen**:
  - erlaubt Spezialisierung und **Arbeitsteilung**
  - definierte Zwischenergebnisse („Meilensteine“)
  - **wiederholbare** Abläufe
- Prozess begleitende **Qualitätssicherung**:
  - Qualität
    - (lat.: qualitas = Beschaffenheit, Eigenschaft, Zustand)
    - Definition nach DIN EN ISO 9000:2000: Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt.
    - Qualität gibt an, in welchem Maße ein Produkt (Ware oder Dienstleistung) den bestehenden Anforderungen entspricht.
  - der **Erstellungsprozess** als Gegenstand der Optimierung
  - Einhalten von Standards wie **ISO 9000, *S*oftware *P*rocess *I*mprovement and *C*apability *D*etermination** (**SPICE),** Capability Maturity Model Integration **(CMMI)**

→ ***Insgesamt Hauptaugenmerk auf Qualität !***

Spannungsfeld

**Der Kunde soll zurückkommen,
nicht das Produkt**



- ***Arbeitsteilung*** *ist das* ***Leitmotiv des Software-Engineering***.
  - **Abgrenzung** von möglichst unabhängigen Teilaufgaben
  - **Planung** und Verfolgung des Ablaufs
  - **Koordination** durch einheitliche Begriffe und Notationen
  - **Modulare Konstruktion** für parallele Entwicklungsarbeit
  - Nutzung vorhandener Software: **Baukastensysteme**
  - Technische und organisatorische **Infrastruktur** für Teamarbeit
- ***aber:***

> **Brooks' Law (1975)**
>
> Adding manpower to a late software project makes it later.

Folgen der Arbeitsteilung

- ***Teamarbeit*** erfordert zusätzlichen Aufwand
  - Rollen im Team festlegen
  - Kommunikation innen/außen

=> Management - Probleme

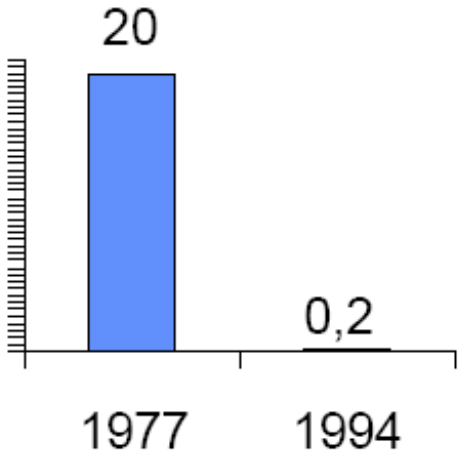
- ***Aufteilung*** in unabhängige Teile
  - zu erstellendes System
  - Erstellungsprozess

=> Informatik - Probleme

- Verschiebung vom ***programmierenden Benutzer*** zum ***fachfremden Programmierer***

=> Ergonomie - Probleme

Folgen des Wachstums



Anzahl Fehler auf 1000 LOC

LOC: Line of Code

Zunehmende Qualitäts-Anforderungen

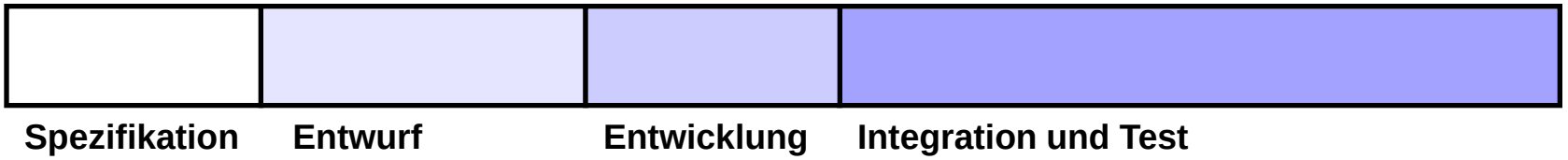
- **Für 50% der Ausfälle im industriellen Sektor sind Software-Fehler verantwortlich**
- **Nach Cusumano haben sich die gefundenen Defekte in jeweils 1000 Zeilen Quellcode folgendermaßen entwickelt:**
  - **1977: durchschnittlich 7- 20 Defekte**
  - **1994: durchschnittlich 0,2 - 0,05 Defekte**
    - **In 13 Jahren konnte die Defektrate also um ungefähr das 100fache gesenkt werden.**

Zunehmende Qualitäts-Anforderungen

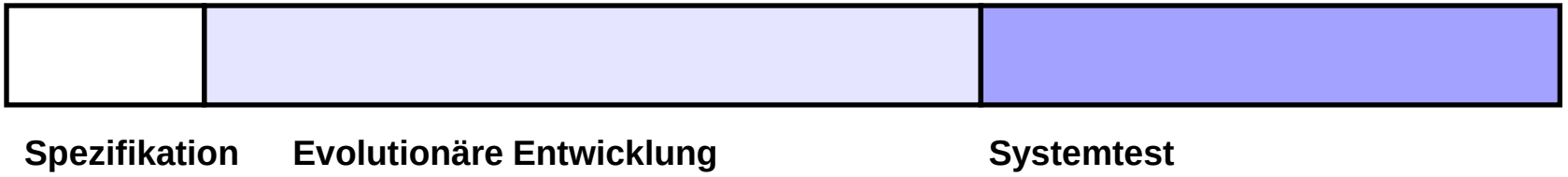
- **0,1%-Defektniveau bedeutet:**
  - **pro Jahr:**
    - **20.000 fehlerhafte Medikamente**
    - **300 versagende Herzschrittmacher**
  - **pro Woche:**
    - **500 Fehler bei medizinischen Operationen**
  - **pro Tag:**
    - **16.000 verlorene Briefe in der Post**
    - **18 Flugzeugabstürze**
  - **pro Stunde:**
    - **22.000 Schecks nicht korrekt gebucht**

Kosten

- Hängt von Prozess und Software ab
- Kostenverteilung:
  - Verteilung der Entwicklungskosten (Test: ca. 40-50%)



  - Kosten bei evolutionärer Entwicklung



Kosten (II)

- Kostenverteilung:
  - Kosten der Weiterentwicklung (ca. 25% Systementwicklung)

  - Kosten der Produktentwicklung

  - Kostenschätzung: COCOMO (siehe Vorlesung SSE)

Fragen ?

- Software Engineering:
  - Qualität, Kosten, Zeit

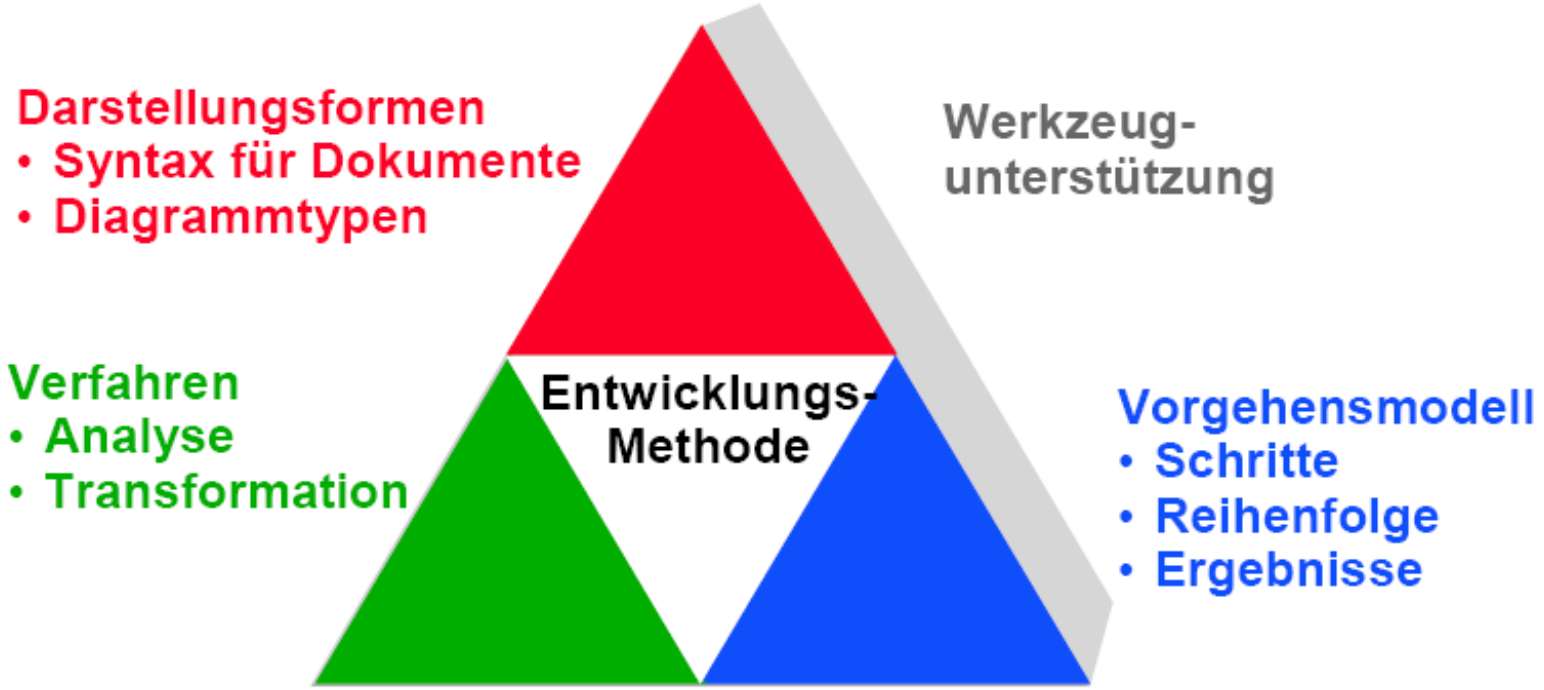
- Softwareprozess:
  - Satz von Tätigkeiten, die zusammenhängende Ergebnisse erzeugen, mit dem Ziel eines Softwareprodukts
  - Tätigkeiten werden von Softwareentwicklern durchgeführt
  - Prozessaktivitäten:
    - **Softwarespezifikation**
      - Definition der Funktion der SW und Randbedingungen für Einsatz
    - **Softwareentwicklung**
      - Erstellen von SW, die Spezifikation erfüllt
    - **Softwarevalidierung**
      - Validieren der SW um sicherzustellen, dass sie tut was der Kunde will
    - **Softwareevolution**
      - Weiterentwicklung der SW

2.4 Begriff: Vorgehensmodell

- Vorgehensmodell:
  - Vereinfachte Beschreibung eines Softwareprozesses (aus einer bestimmten Perspektive)
  - Abstraktion des tatsächlichen Prozesses (Modell)
  - Beschreibung von Tätigkeiten und Rollen (Personen)
  - Arten von Vorgehensmodellen:
    - Arbeitsablaufmodell
      - Abfolge von Aktivitäten, Eingabe, Ausgabe, Abhängigkeiten
    - Datenfluss- oder Aktivitätsmodell
      - Menge von Aktivitäten mit Datenumwandlung je Aktivität
    - Rolle/ Aktions-Modell
      - Rolle des Menschen im Prozess, Verantwortlichkeiten der Rollen

Vorgehensmodell

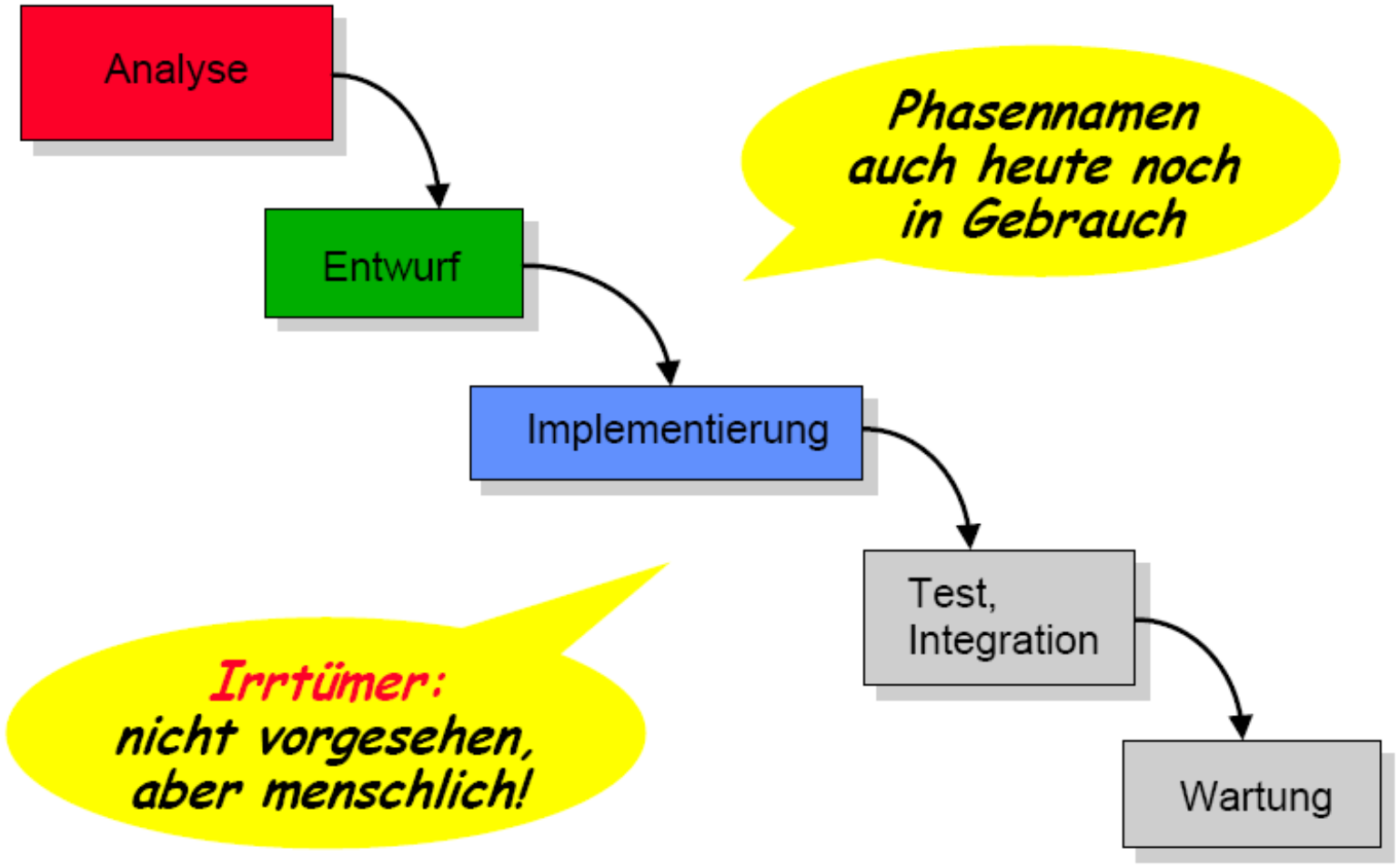
- Teil einer ***Software - Entwicklungsmethode***
- Zweck: Hilfe bei der Erstellung von Software
  - Bessere Planbarkeit der ***Entwicklung***
  - Bessere Struktur des ***Produkts***



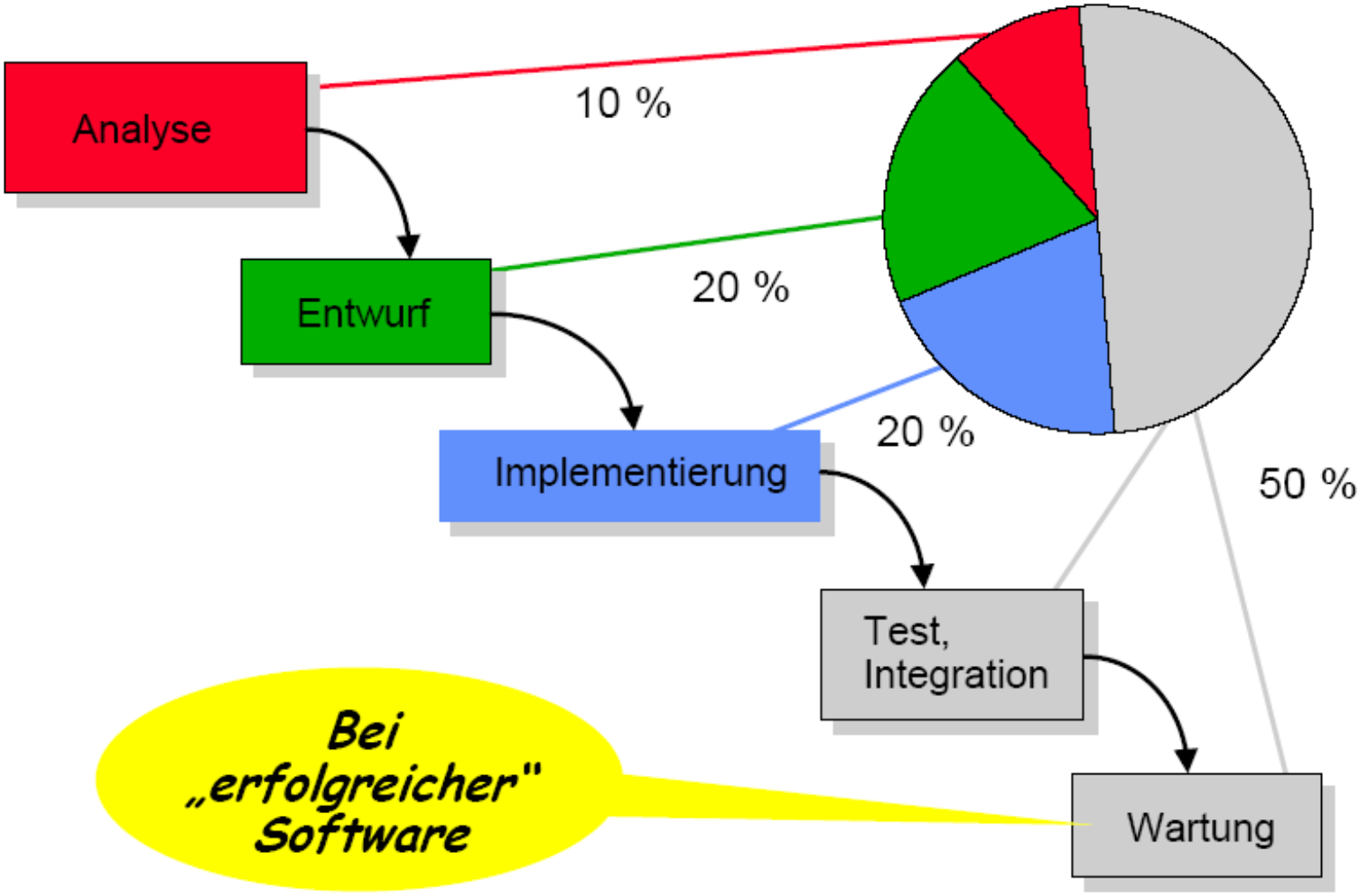
Beispiele von Vorgehensmodellen

- Wasserfallmodell → Phasen
- V-Modell → Vorgehensmodell
- Spiralmodell → Fokus auf Risiken
- Evolutionäre Entwicklung → Fokus auf schnelle Änderungen
  - Agile Methoden (eXtreme Programming XP, usw.)
- Prototypen → Entwicklung ist durch Prototypen getrieben
- Formale Umsetzung → Fokus auf dem Formalen
- Zusammensetzung des Systems aus wieder verwendbaren Komponenten → Fokus: Baukastensystem

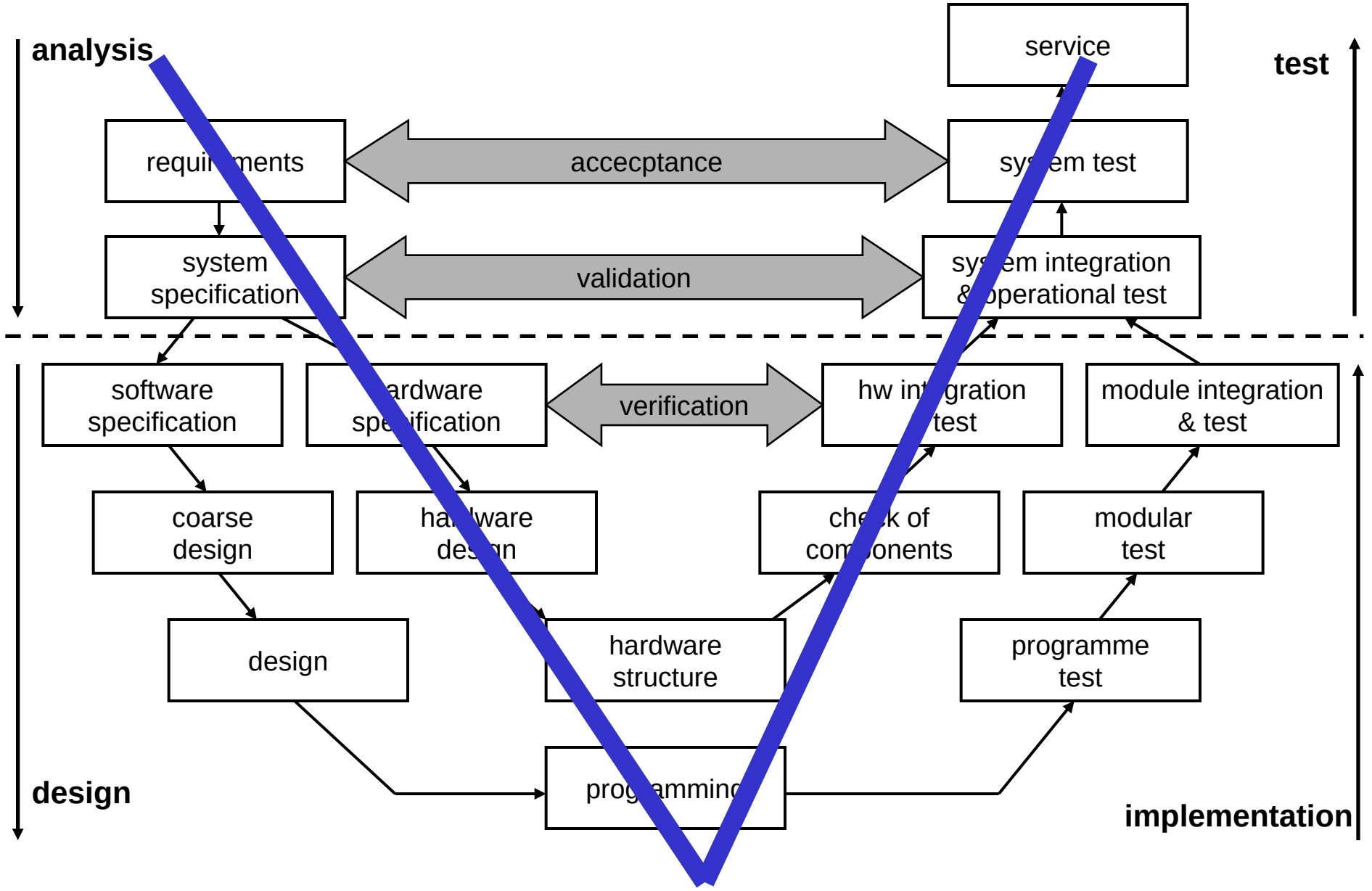
Wasserfall-Modell (nach Royce, ca 1970)



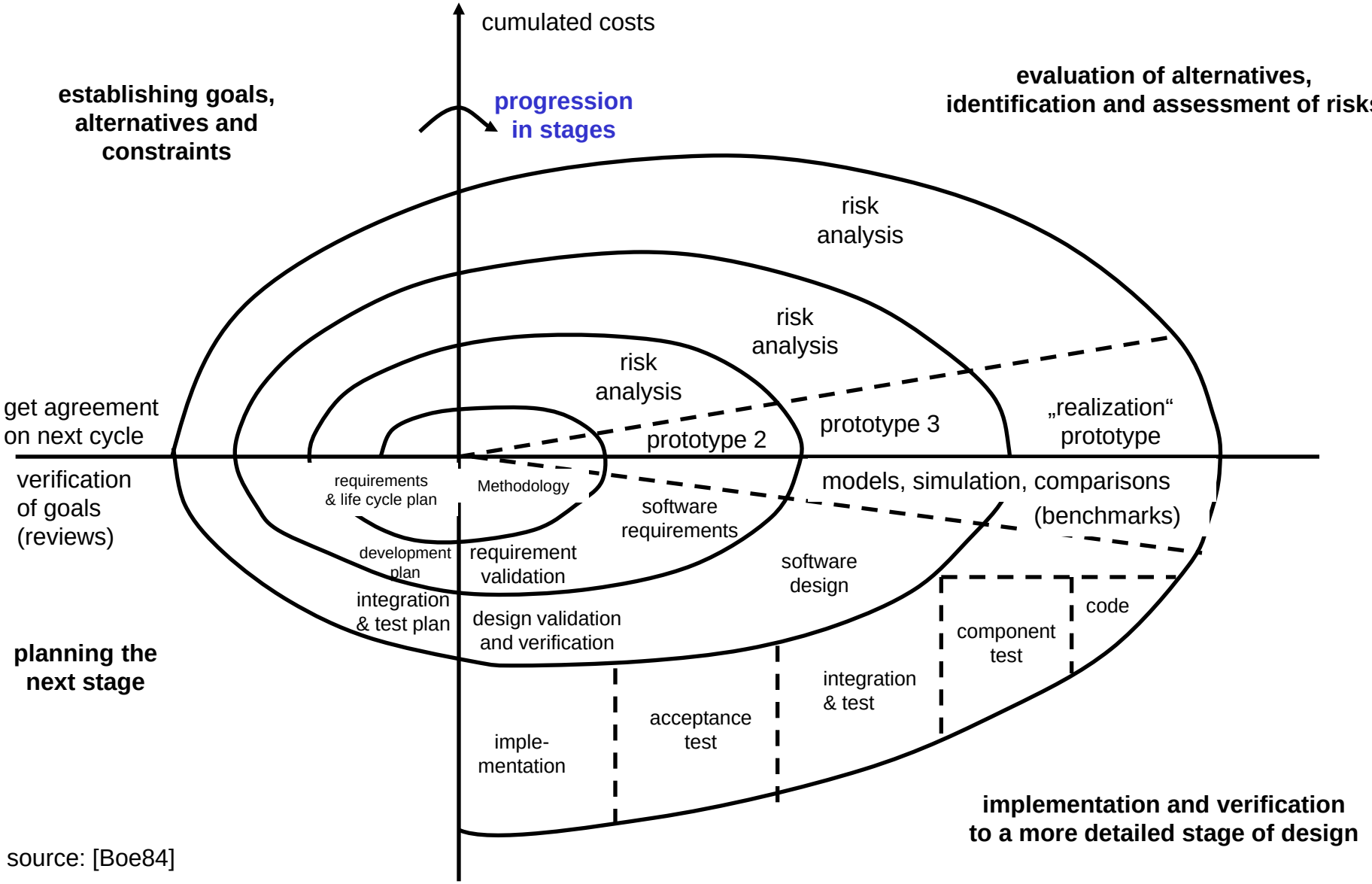
Ungefähre Verteilung des Arbeitsaufwandes



V- Model of system design process



Spiral model of the software life cycle



source: [Boe84]

Evolutionäre/Agil Entwicklung: Extreme Programming

Wesentliche „Zutaten“:

(1) ***Minimale „Dokumentation“***

- ***Story-Board (Use-Cases), Testfälle und Programmcode***

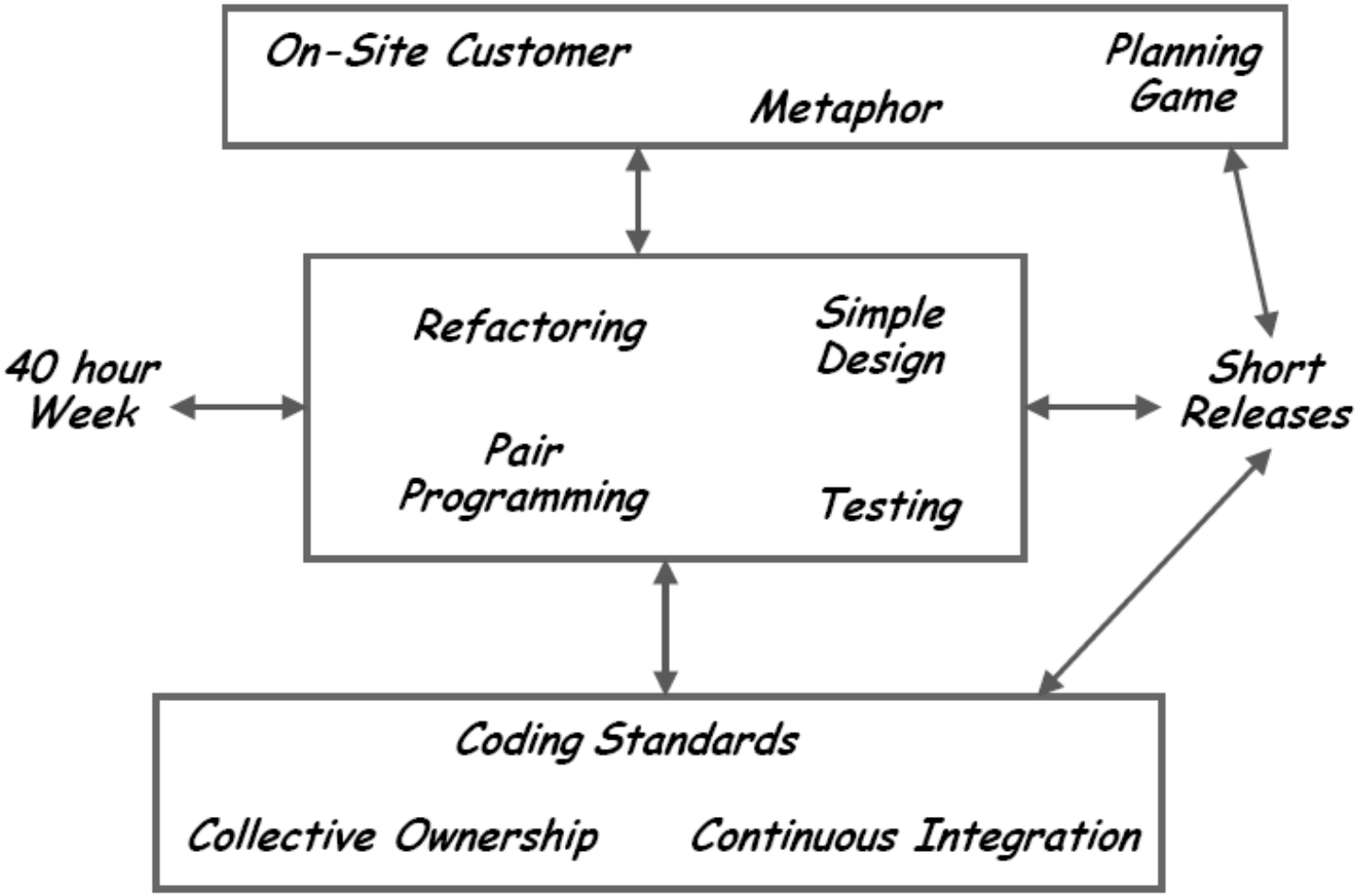
(2) ***Keine vorausschauende Planung***

- ***Nur sofort Notwendiges im Entwurf berücksichtigt***
  → You Aren't Going to Need It (JAGNI)
- ggfs ***„Refactoring“*** (Umbau des Programmcodes)
- Vorgabe der ***Prioritäten*** durch Nutzer (welche Story?)
- ***kurze Planungszyklen*** (wenige Wochen)

(3) ***Techniken***

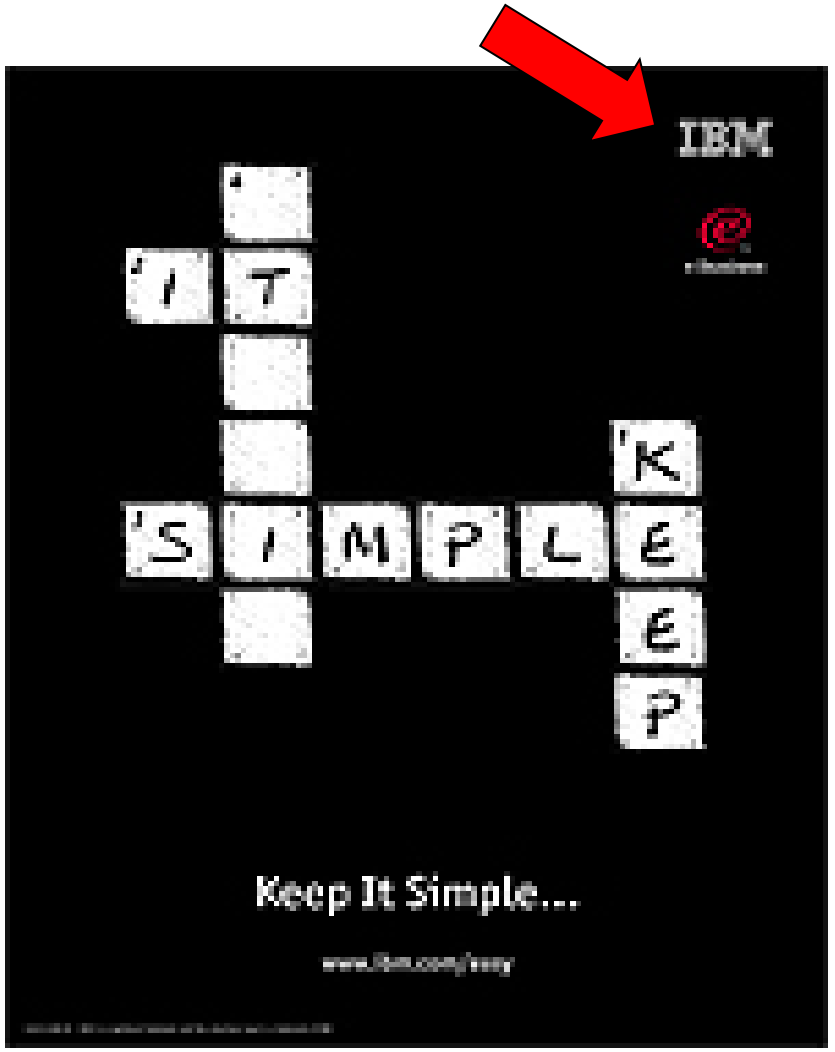
- ***„Pair Programming“***
- ***nach jeder Ergänzung alle Testfälle ausführen***
- ***laufend Integration => stets ausführbarer Prototyp***
- ***So einfach wie möglich***
  → Keep It Simple and Stupid (KISS)

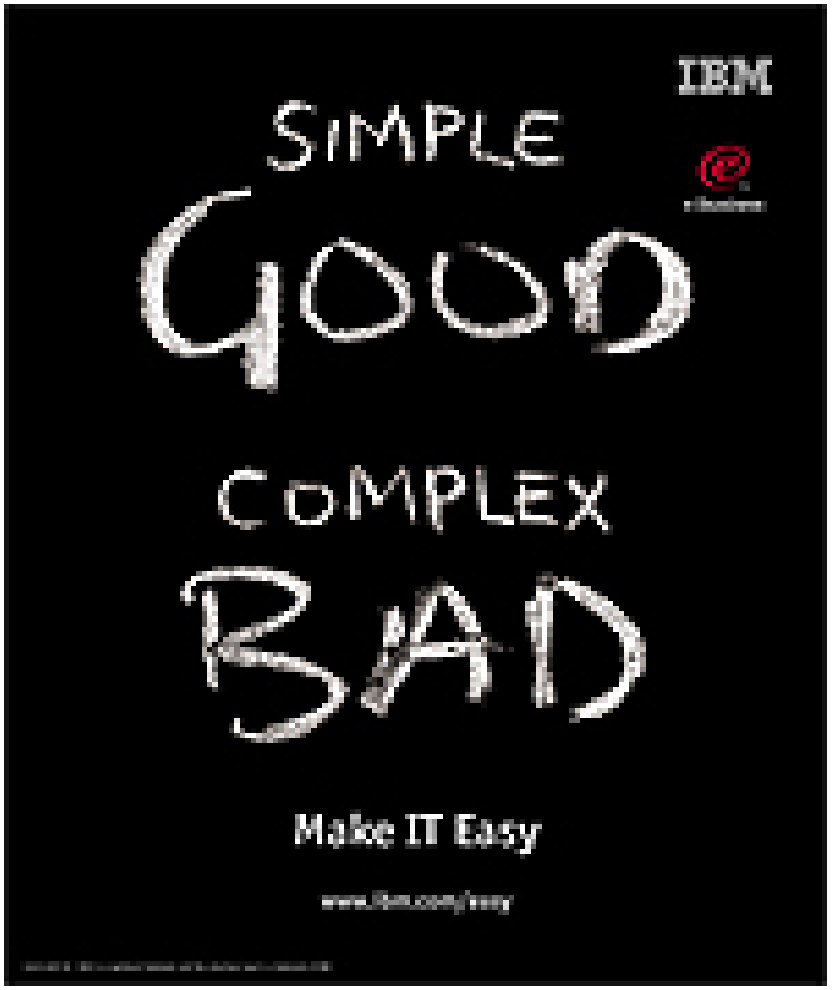
„Extreme Programming“ - Praktiken



http://www.extremeprogramming.org/

10. Nov. 2003





http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/650

Fragen ?

- Vorgehensmodelle und ihre Einordnung?
  - Wasserfall
  - V
  - Spiral
  - XP

2.5 Begriff: Methode

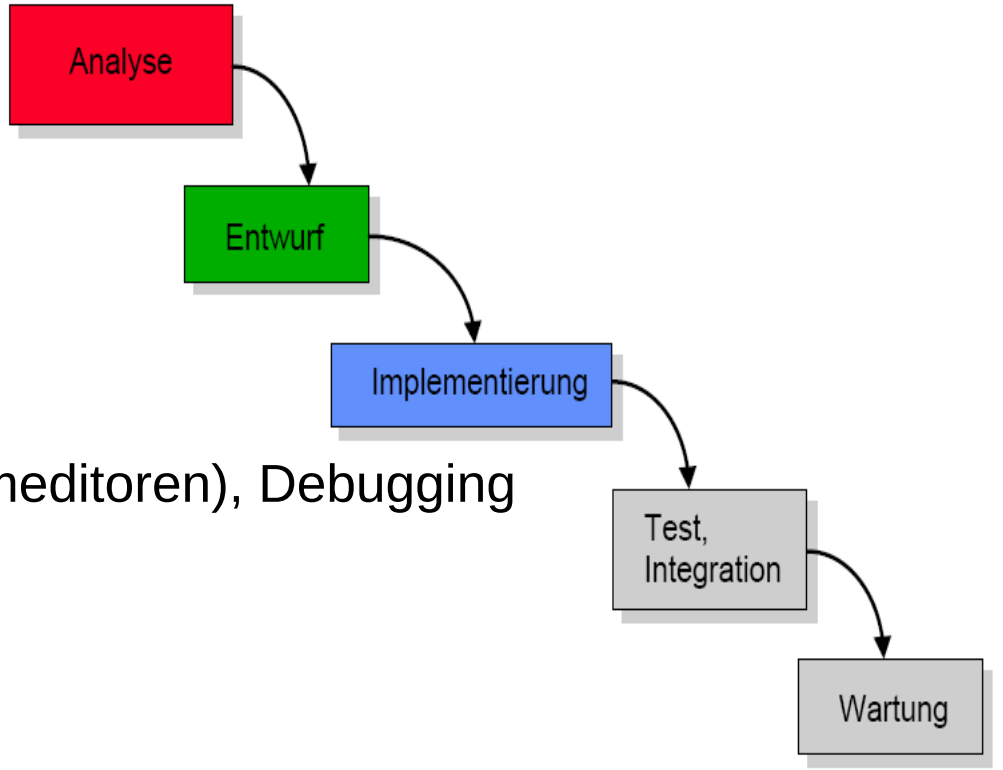
- Methode
  - Strukturierter Ansatz für die Softwareentwicklung
  - Oft grafische Modelle eines Systems, die zur Systemspezifikation oder zum Systementwurf verwendet werden
  - Ziel:
    - Kostengünstige Systeme
    - Hohe Qualität
  - Historie:
    - 70er: **Funktionsorientierte** Methoden (Bestimmung der Funktionalen Komponenten)
      - Strukturierte Analyse (SA) (DeMarco 1978)
      - Jackson Structured Design (Jackson 1983)
    - 80er, 90er: **Objektorientierte** Methoden
      - erweitern Funktionsorientierte Methoden
      - Sprache: Unified Modeling Language (UML)
        - » Booch 1994, Rumbaugh 1991, Fowler and Scott 1997

Begriff: Methode (II)

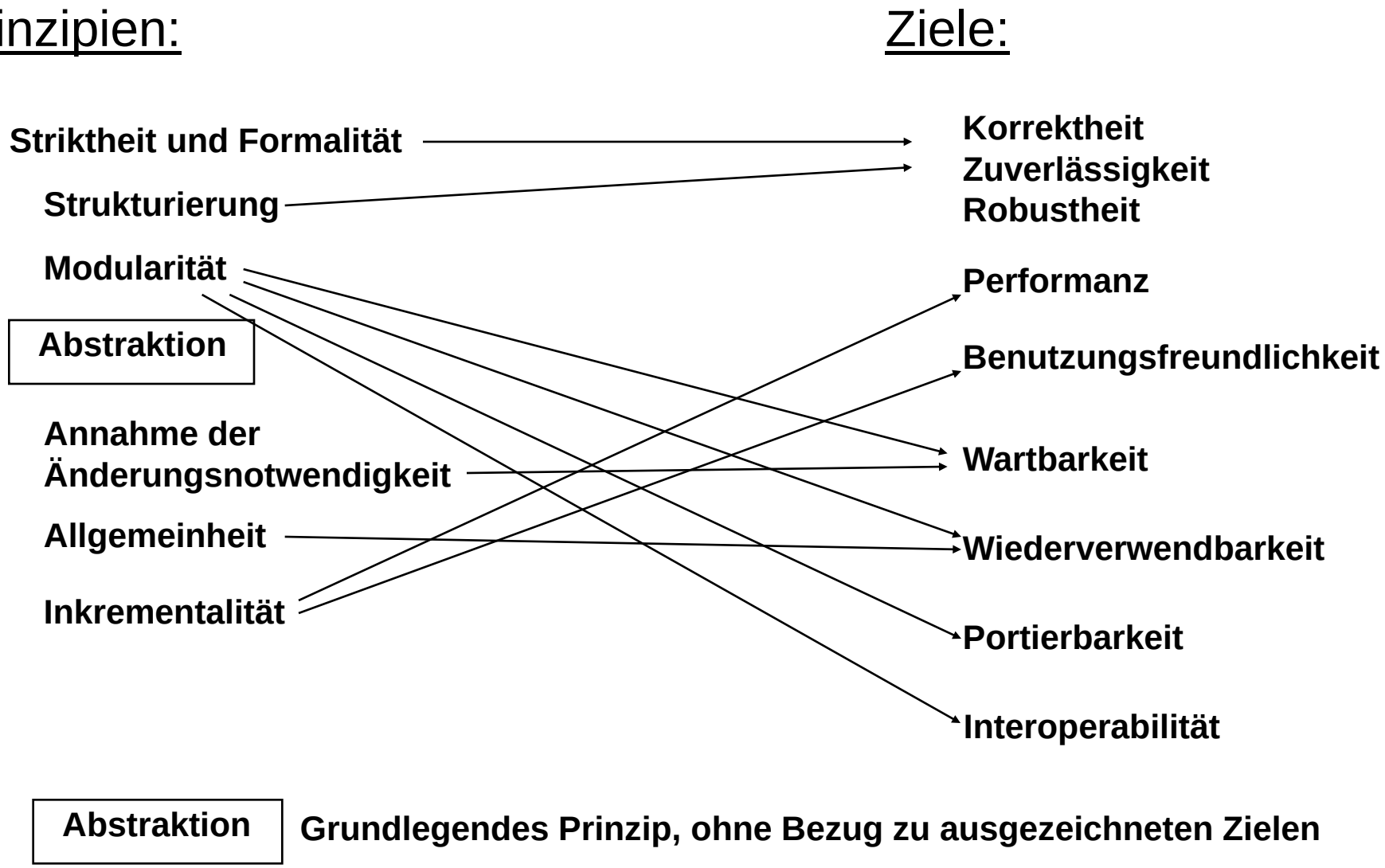
- Bestandteile einer Methode
  - **Beschreibung von Systemmodellen**
    - Definition der verwendeten Notation
    - Z.B. Klassenmodelle, Datenflussmodelle, Zustandsmodelle, usw.
  - **Regeln**
    - Beschränkungen, die für Systemmodelle immer gelten
    - Z.B. Name innerhalb eines Modells muss eindeutig sein
  - **Empfehlungen**
    - Verfahren eines guten Entwurfstils (Entwurfsmuster, ...)
    - Führt zu einem gut organisierten Systemmodell
    - Z.B. Kein Objekt sollte mehr als 7 untergeordnete Objekte haben
  - **Anleitung zum Vorgehen**
    - Beschreibung der durchzuführenden Aktivitäten
    - Organisation der Aktivitäten
    - Z.B. Objektattribute sollen dokumentiert werden bevor Operationen definiert werden

2.6 Begriff: CASE

- CASE
  - Computer Aided Software Engineering (computerunterstütztes Software Engineering)
  - **Werkzeuge** (Softwareprogramme)
  - Unterstützen Aufgaben:
    - Anforderungsanalyse
    - Systemmodellierung
    - Fehlerbehebung
    - Tests, Testfallgeneratoren
    - Codegenerator
    - Implementierung (Programmeditoren), Debugging
    - Usw.



Prinzipien des Software Engineering



Abstraktion **Grundlegendes Prinzip, ohne Bezug zu ausgezeichneten Zielen**

Fragen ?

- Methode
- CASE
- Prinzipien