

KV Software Engineering
Prof. Dr. Martin Glinz

Kapitel 1

Prinzipien des Software Engineerings



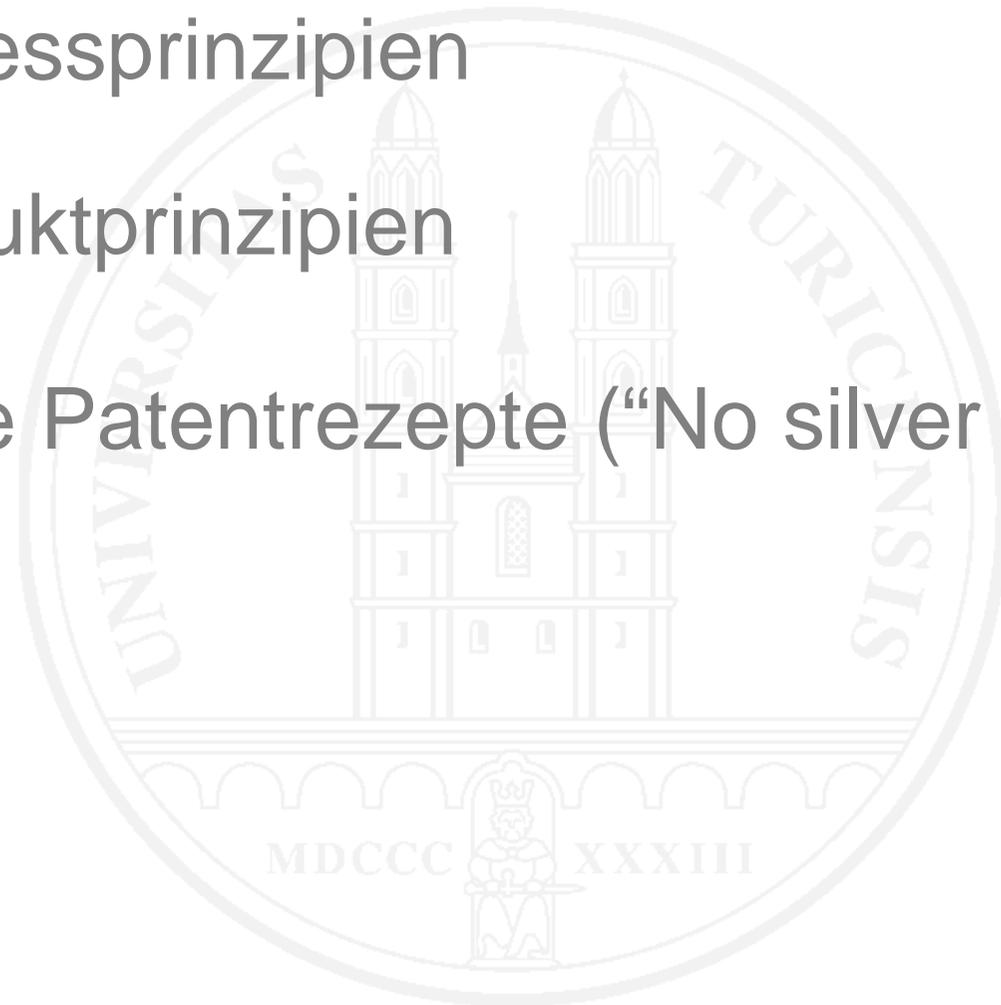
Universität Zürich
Institut für Informatik

1.1 Prinzipien als Grundlage

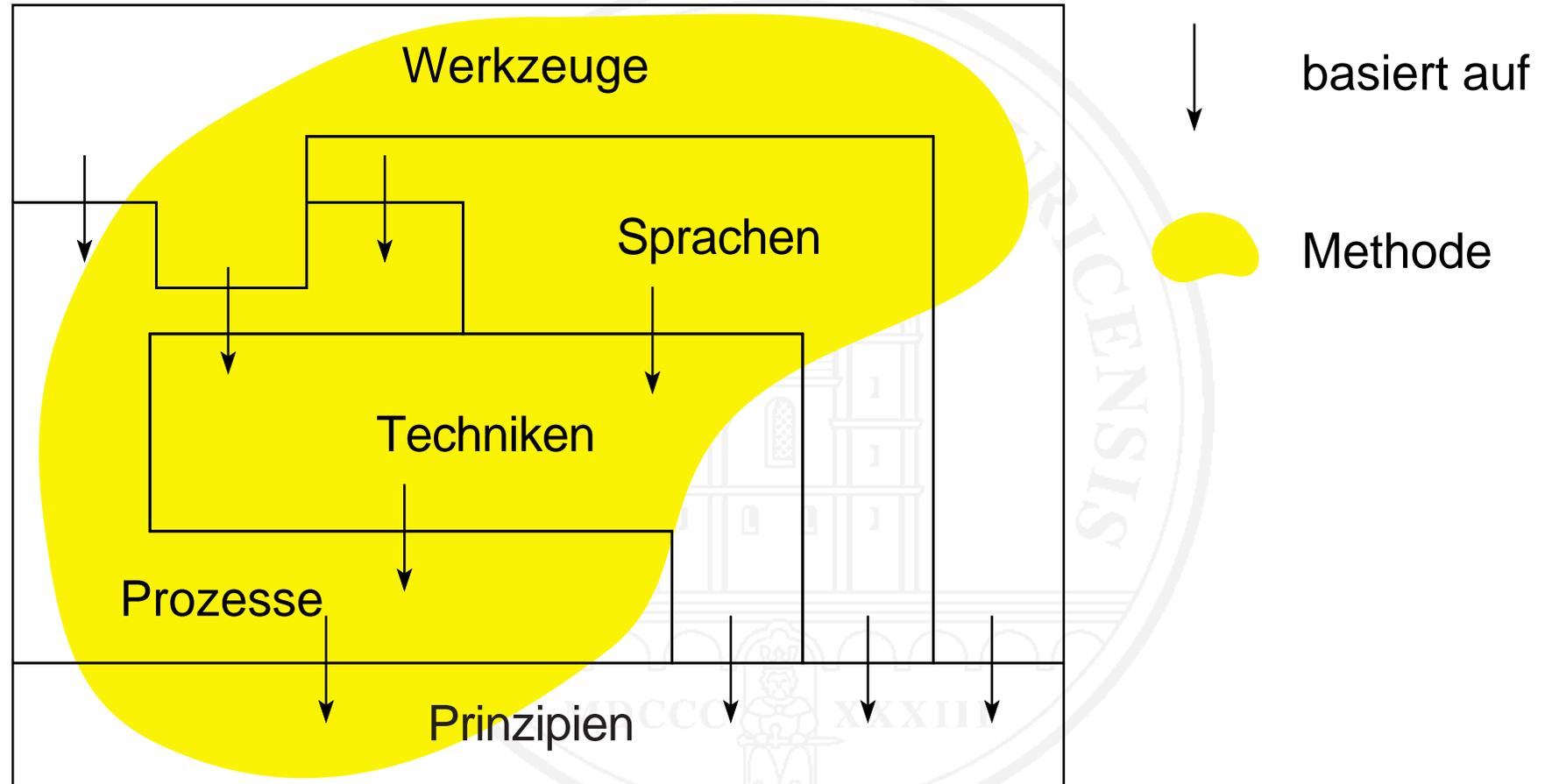
1.2 Prozessprinzipien

1.3 Produktprinzipien

1.4 Keine Patentrezepte (“No silver bullet”)



Überblick

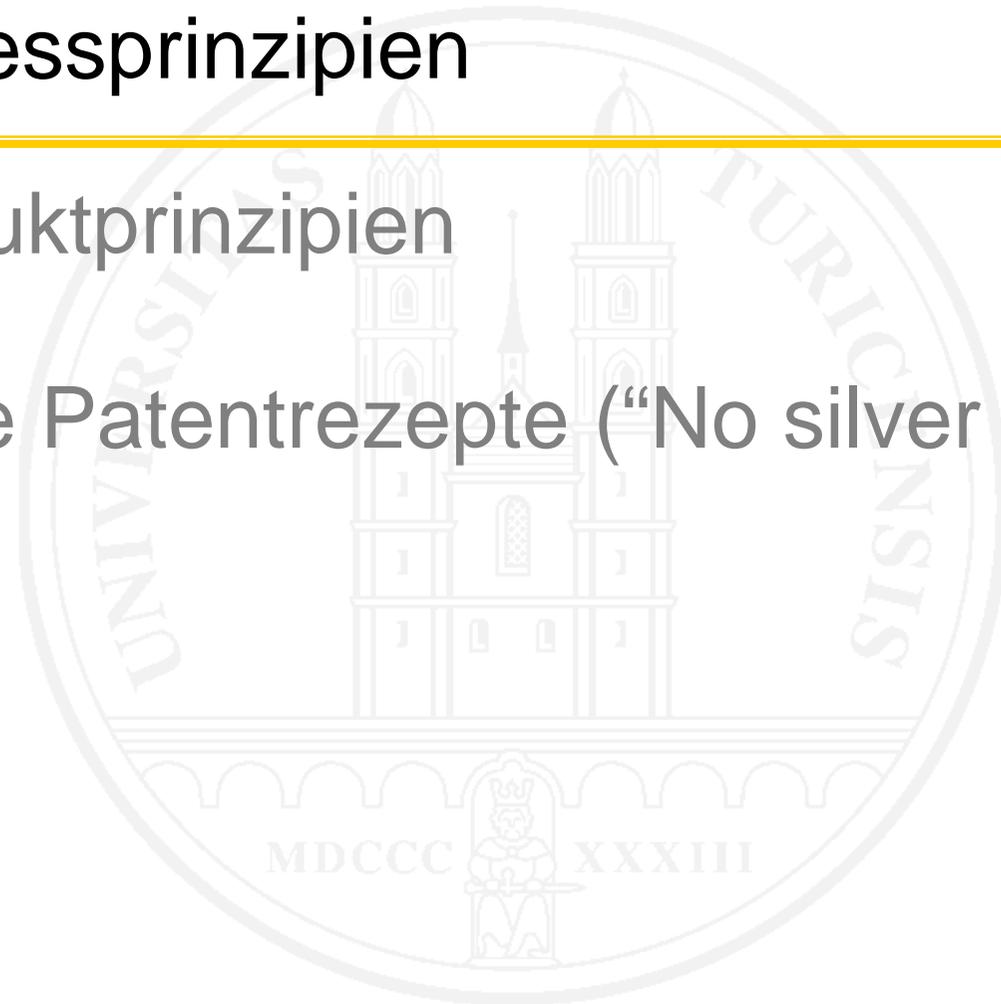


1.1 Prinzipien als Grundlage

1.2 Prozessprinzipien

1.3 Produktprinzipien

1.4 Keine Patentrezepte (“No silver bullet”)



Boehm's sieben Prinzipien (plus ein weiteres)

1. Manage using a phased life-cycle plan
2. Perform continuous validation
3. Maintain disciplined product control
4. Use modern programming practices
5. Maintain clear accountability for results
6. Use better and fewer people
7. Maintain a commitment to improve the process

Das achte Prinzip (neu)

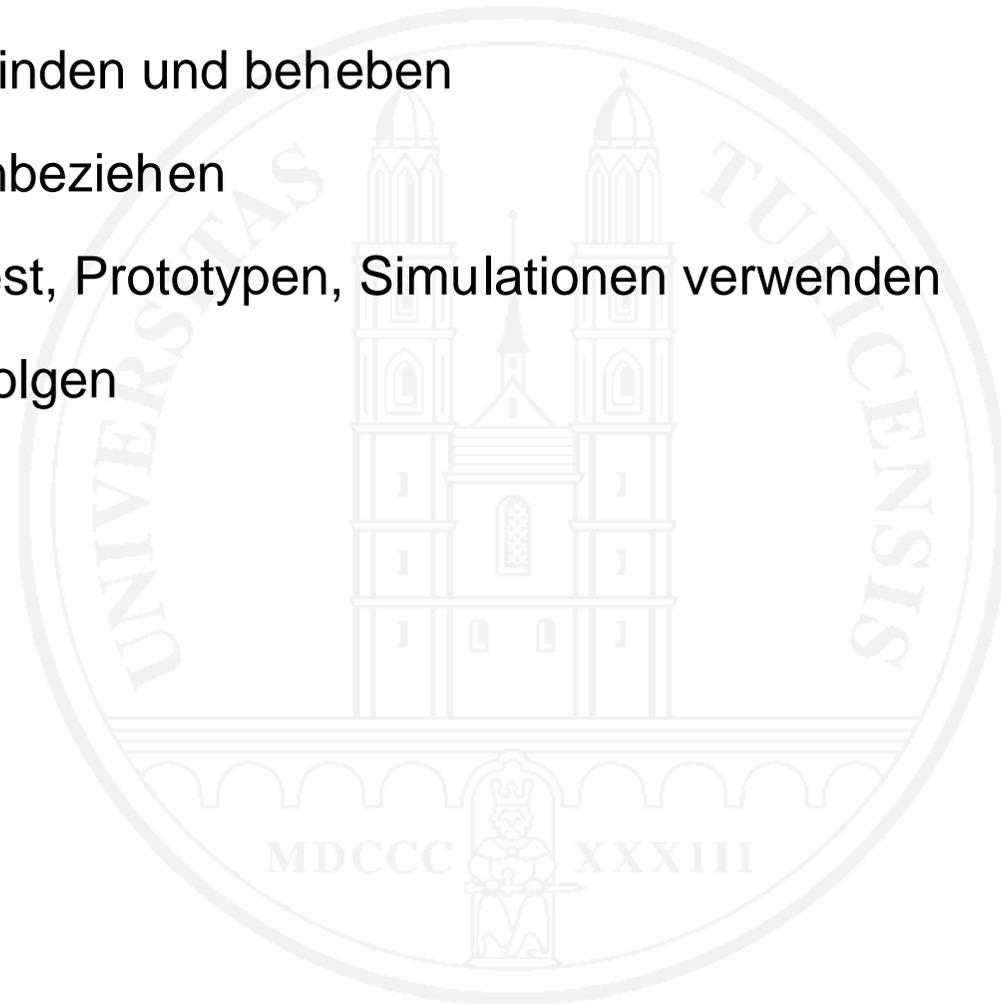
8. Reuse or buy software where you can

1. Manage using a phased life-cycle plan

- Das Projekt planen (Sache, Kosten, Termine, Risiken, Personal, Mittel...)
- (Realistischen) Projektplan erstellen
- Geeignetes Prozessmodell zugrunde legen
- Das Projekt nach dem Plan führen

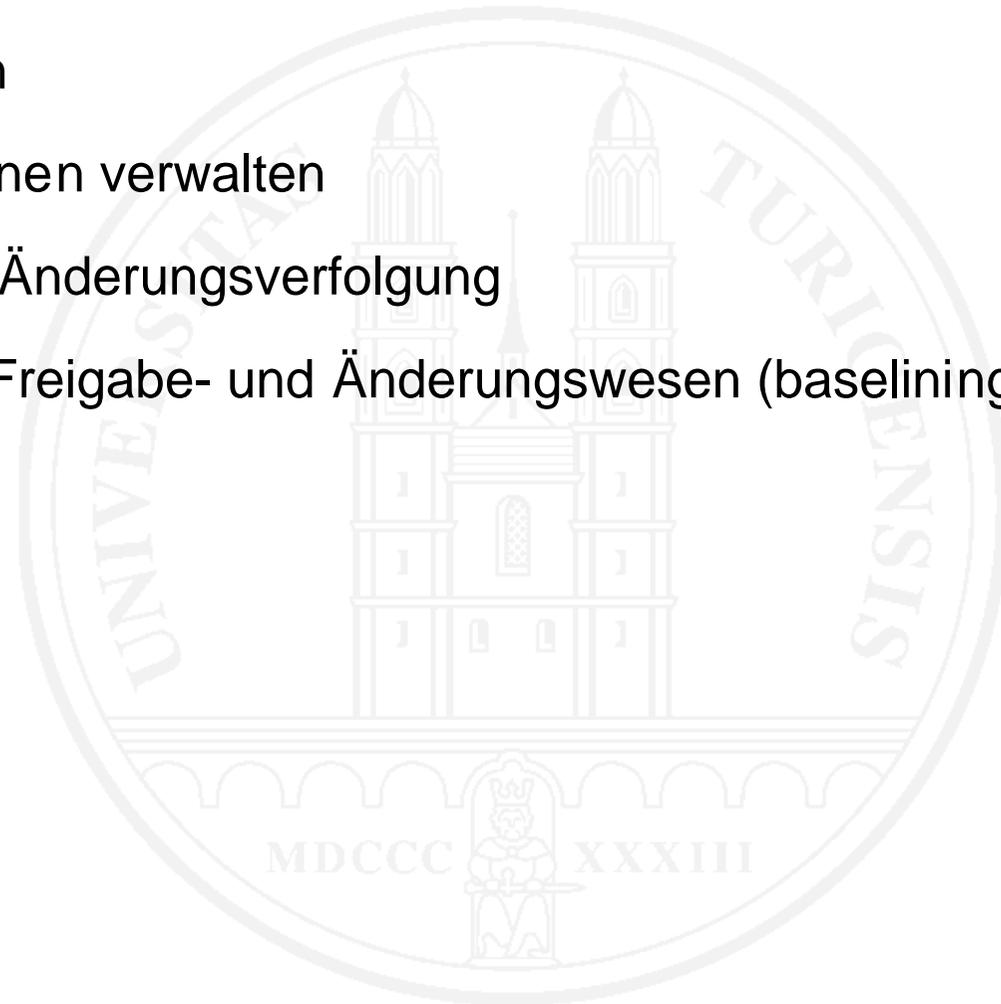
2. Perform continuous validation

- Fehler früh finden und beheben
- Benutzer einbeziehen
- Reviews, Test, Prototypen, Simulationen verwenden
- Risiken verfolgen



3. Maintain disciplined product control

- Identifikation
- Konfigurationen verwalten
- Status- und Änderungsverfolgung
- Geregelter Freigabe- und Änderungswesen (baselining / change control)



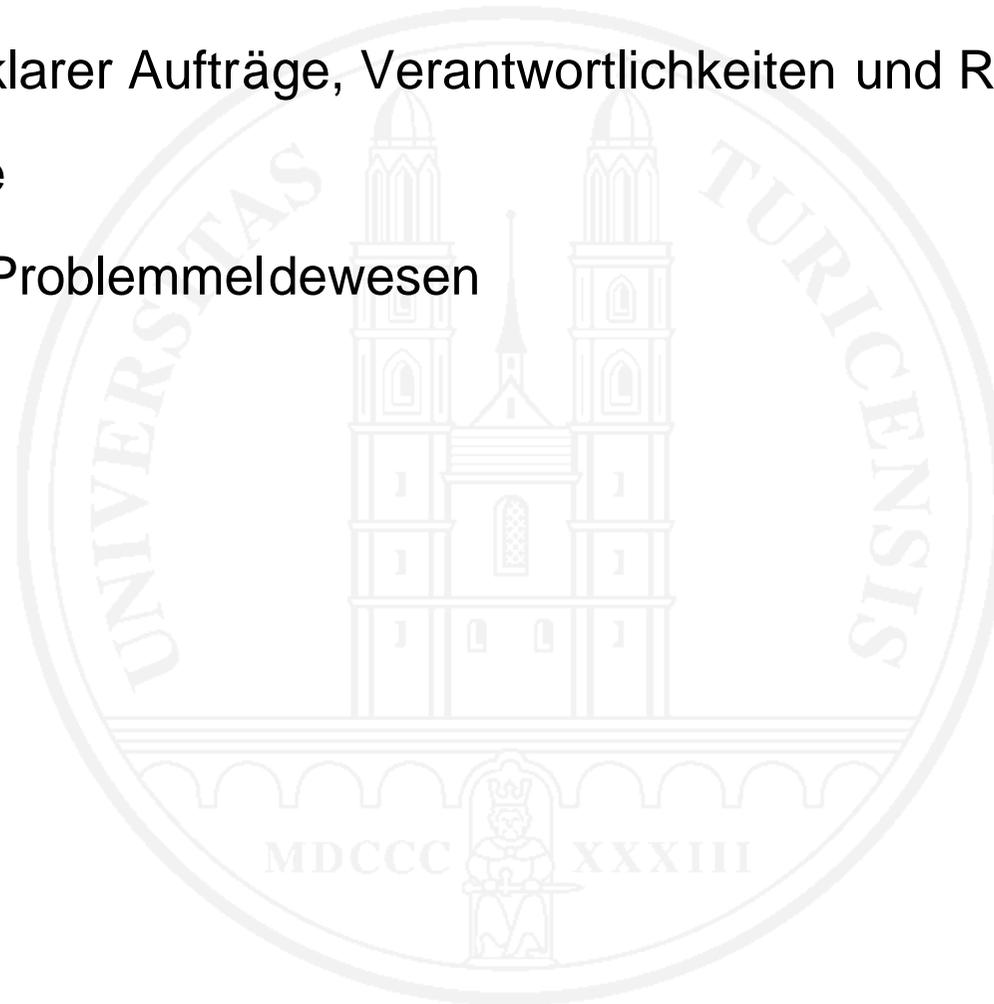
4. Use modern programming practices

- Programmiersprachen
- Spezifikations- und Entwurfssprachen
- Entwicklungsmethoden
- Werkzeuge

Heute: Programmierung ⇔ Software Entwicklung und -pflege

5. Maintain clear accountability for results

- Zuweisung klarer Aufträge, Verantwortlichkeiten und Ressourcen
- Meilensteine
- Geregeltes Problemmeldewesen



6. Use better and fewer people

- Weniger Leute = weniger nicht produktiver Aufwand (Kommunikation!)
- Die Leistungsspanne zwischen guten und schlechten Leuten ist viel größer als die Gehaltsspanne ⇨ Wirtschaftlichkeit
- Leute fördern und ausbilden
- Win-Win-Situationen schaffen: Die richtigen Leute am richtigen Ort

7. Maintain a commitment to improve the process

- Messen
- Analysieren, Auswerten, Handeln
- Verbesserungsprogramme
- Prozesse nicht in Stein meißeln
 - Prozesse leben und evolvieren
 - Prozesse sind Mittel zum Zweck

8. Reuse or buy software where you can (neu)

- Kauf- /Beschaffungsabklärungen in alle Prozesse integrieren
- Kein falscher Entwicklerstolz („Wir machen alles selbst“)
- Komponentenorientiert arbeiten

1.1 Prinzipien als Grundlage

1.2 Prozessprinzipien

1.3 **Produktprinzipien**

1.4 Keine Patentrezepte (“No silver bullet”)

Produktprinzipien – 1

**Ghezzi/Jazayeri/
Mandrioli (1991)**

Fairley (1985)

Glinz (1998)

Rigor and formality

Abstraction

Abstraction

Strukturen und
Abstraktionen

Modularity

Modularity

Modularität

Separation of
concerns

Information Hiding

Geheimnisprinzip
(Information Hiding)

Schnittstellen und
Verträge

Anticipation of change

Generality

Produktprinzipien – 2

**Ghezzi/Jazayeri/
Mandrioli (1991)**

Fairley (1985)

Glinz (1998)

Incrementality

Concurrency

Nebenläufigkeit

Aesthetics

Ästhetik

Verification

Qualität

Berücksichtigung der
Ressourcen

Aspektorientierung

Nutzung von
Vorhandenem

1.1 Prinzipien als Grundlage

1.2 Prozessprinzipien

1.3 Produktprinzipien

1.4 Keine Patentrezepte (“No silver bullet”)

Keine Patentrezepte ("No silver bullet")

- "No silver bullet" (Brooks, 1987)
- **Essentielle** Probleme des Software Engineerings
- **Routine**probleme
- Die meisten gelösten Probleme sind **Routineprobleme**
- Die Kernprobleme des Software Engineerings sind hart und bleiben es

Fortschritt im Software Engineering

- Es gibt **echte Fortschritte** im Software Engineering
- zum Beispiel durch konsequente Anwendung der oben genannten Prinzipien
- Aber : Das Angehen immer größerer und komplexerer Probleme **frisst die Fortschritte** fortlaufend **auf**
- Software-Entwicklung ist und bleibt eine anspruchsvolle, herausfordernde Aufgabe, die
 - **ernst genommen** werden muss,
 - **ausgebildet** werden muss,
 - **gute Leute** erfordert.

Literatur

- Boehm, B. (1983). Seven Basic Principles of Software Engineering. *Journal of Systems and Software* **3**, 3-24.
- Brooks, F.P. (1995). *The Mythical Man Month. Essays on Software Engineering*. Anniversary Edition Reading, Mass.: Addison-Wesley.
- Brooks, F.P. (1987). No Silver Bullet. Essence and Accidents of Software Engineering. *IEEE Computer* **20**, 4. 10-19.
- Davis, A.M. (1995). *201 Principles of Software Development*. New York: McGraw-Hill.
- Fairley, R. (1985). *Software Engineering Concepts*. New York: McGrawHill.
- Ghezzi C., M. Jazayeri, D. Mandrioli (1991). *Fundamentals of Software Engineering*. Englewood Cliffs, N.J.: Prentice Hall.
- Glinz, M. (1998). *Software Engineering I*. Vorlesungsskript, Universität Zürich.
- Meyer, B. (1992). Applying "Design by Contract". *IEEE Computer* **25**, 10 (Oct. 1992). 40-51.
- Parnas, D.L. (1972). On the Criteria To Be Used in Decomposing Systems into Modules. *Communications of the ACM* **15**, 12 (Dec. 1972). 1053-1058.