Software Engineering I

Prof. Dr. Martin Glinz

Kapitel 12

Produktivitätsfaktoren

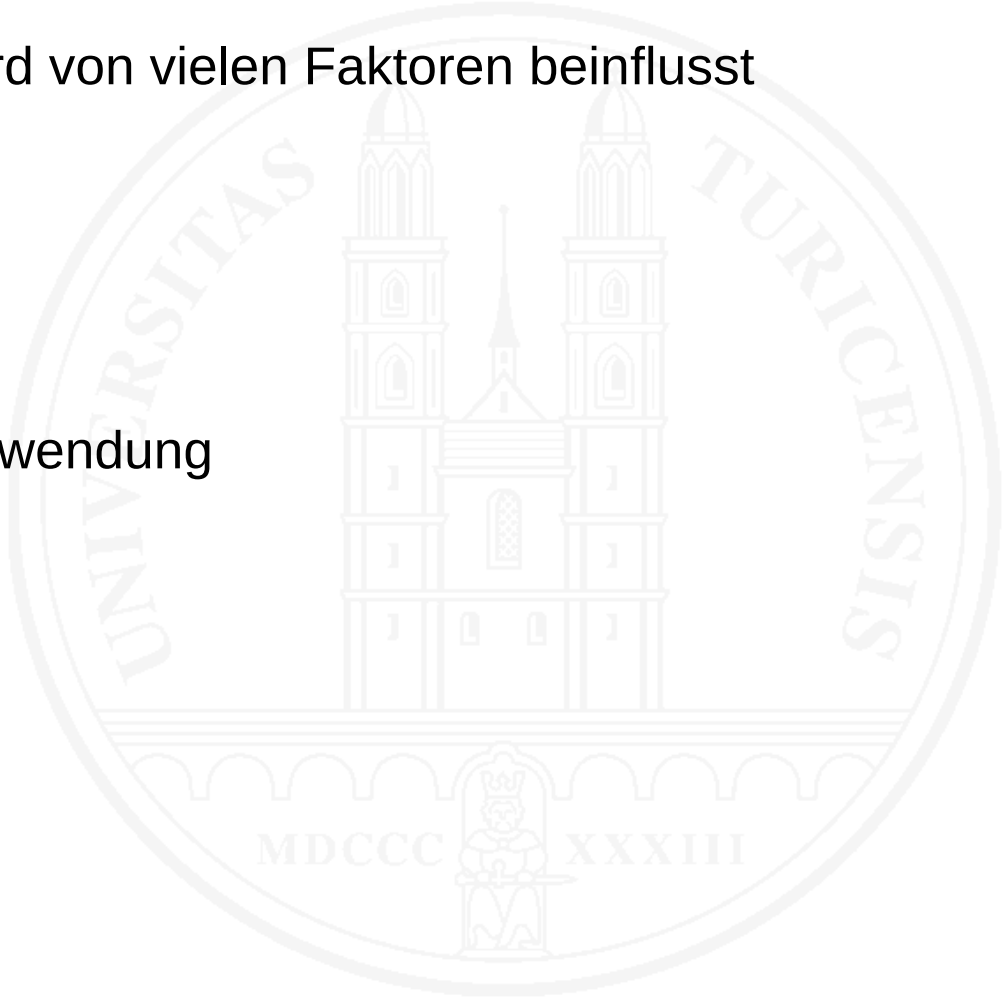
Universität Zürich
Institut für Informatik

Einflussfaktoren

Produktivität wird von vielen Faktoren beinflusst

Hauptfaktoren:

- Werkzeuge
- Mehrfachverwendung
- Menschen



Werkzeuge – 1

Zum Schnitzen braucht es Messer.

... und zum Entwickeln von Software braucht es Werkzeuge.

Aber:
Die besten Messer sind nutzlos ...

... wenn der Schnitzer nicht mit ihnen umgehen kann

... wenn er nicht weiß, was er schnitzen soll.

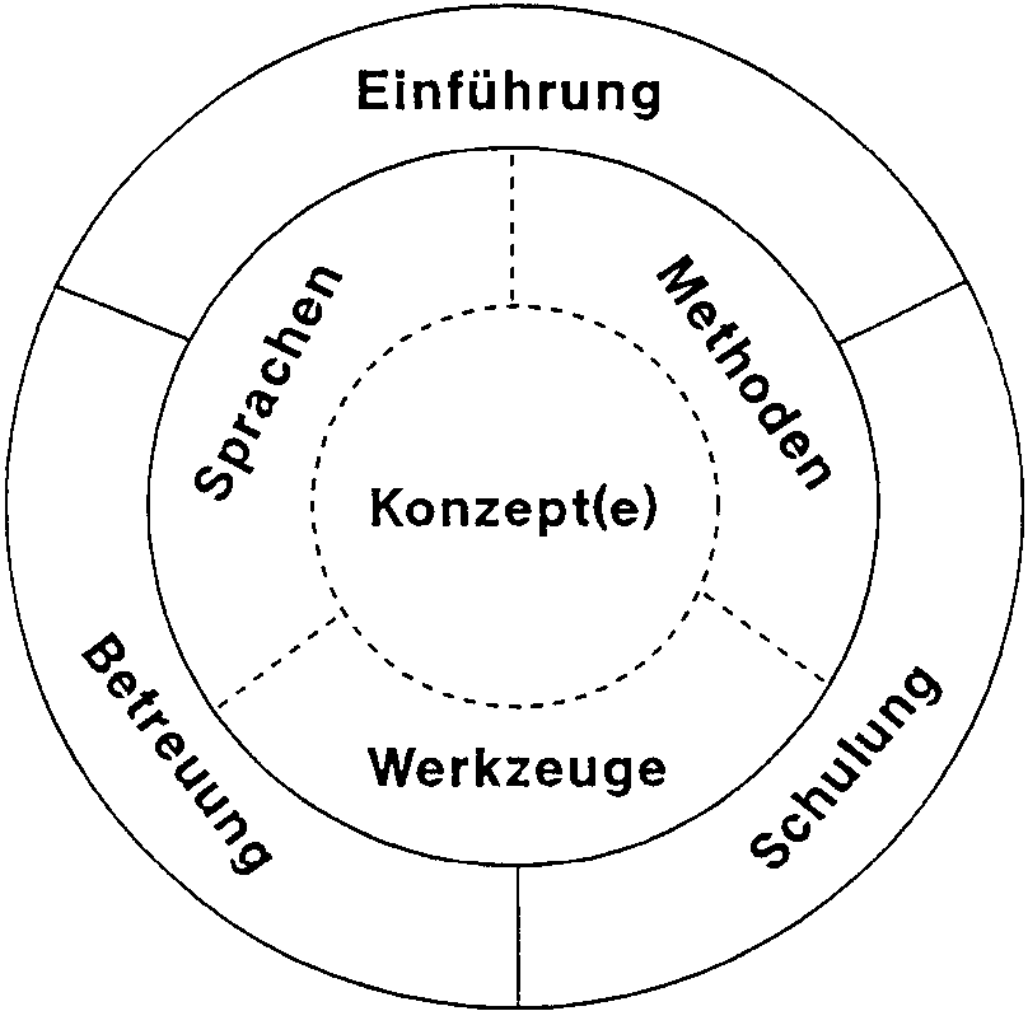
Werkzeug (tool) – rechnergestützte Hilfsmittel für die Entwicklung von Software. Auch: CASE (Computer Aided Software Engineering)

Was Werkzeuge können

- Entlasten von Routineaufgaben
- Bearbeiten Sprachen
- Unterstützen den Einsatz von Methoden
- Vereinfachen Änderungen

- Aber: Werkzeuge sind keine Wunderwaffen:
- Keine Produktivitätssteigerung um Größenordnungen
- Ersetzen eigenes Denken und sorgfältiges Arbeiten nicht
- Machen das Qualitätsmanagement nicht überflüssig

CASE (Computer Aided Software Engineering)



Klassifikation von Werkzeugen

- Editoren
- Spezifikations- und Entwurfssysteme
- Programm-Entwurfssysteme
- Compiler, Browser und Programmierumgebungen
- Programm-Generatoren
- Mess- und Testwerkzeuge
- Konfigurationsverwaltungs-Systeme

Produktivitätsgewinn durch Werkzeuge

Substanzielle Produktivitäts- und Qualitätssteigerungen durch Werkzeug-Einsatz sind realisierbar

Aber: Bei der Einführung sinkt die Produktivität zunächst:

- Schulung
- Eingewöhnung
- Verlagerung von Aufwendungen

Der Gewinn kommt erst mittelfristig:

➪ Werkzeug-Einführung ist eine Investition!

Planung des Werkzeugeinsatzes

- Was soll unterstützt werden?
- Wie wirtschaftlich ist der Einsatz?
- Welche Entwicklungskonzepte (Methoden, Sprachen) werden eingesetzt?
- Ist die Schulung geregelt?
- Wie sieht die Einführungsstrategie aus?
- Ist die Betreuung sichergestellt?

Mehrfachverwendung

Die Masse macht's.

Kostensenkungen über hohe Stückzahlen

- Große Produktserien mit identischer Software
- Mehrfachverwendung der gleichen Software in mehreren Produkten

Die Rolle der Menschen im Software Engineering

Software wird von Menschen gemacht.

- Die Software-Leute sind ein entscheidender Produktivitätsfaktor
  - Können
  - Motivation
  - Arbeitsumfeld

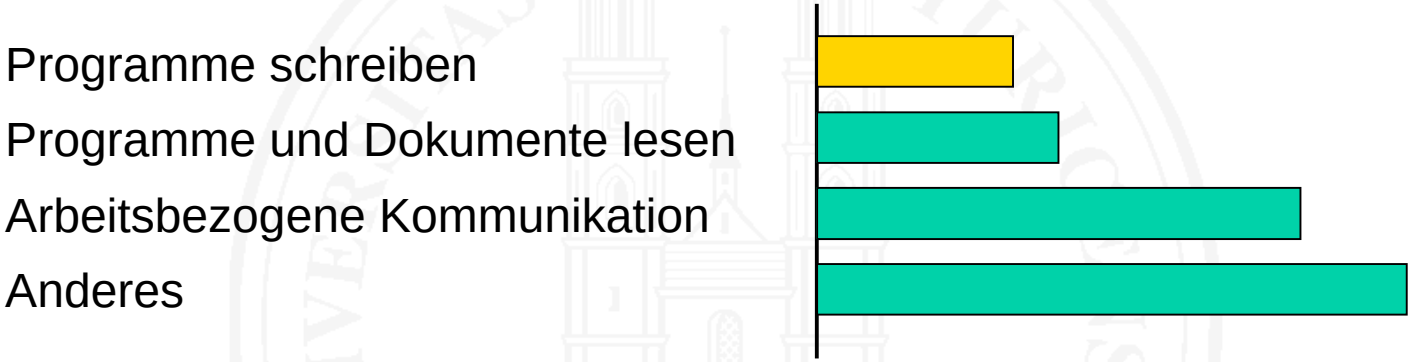
Gesetzmäßigkeiten über Software-Leute – 1

- Produktivität
  - Enorme Schwankungsbreiten, bis 20:1
  - Selbst bei Gruppen noch Schwankungsbreiten bis 4:1

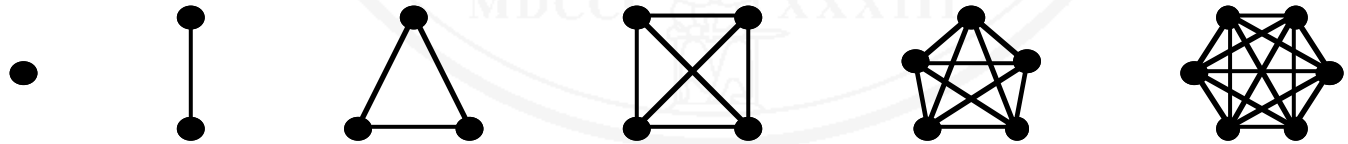
- Disponibilität
  - Personalbestände nur langsam auf- und abbaubar
  - Zu jedem Aufwand eine optimale Personenzahl
  - Das Aufstocken des Personalbestands in einem verspäteten Projekt führt zu noch mehr Verspätung (Gesetz von Brooks)

Gesetzmäßigkeiten über Software-Leute – 2

- Arbeitsverteilung
  - Programmierer schreiben nicht nur Programme



- Gruppengröße
  - Nicht produktiver Aufwand wächst überproportional mit der Gruppengröße



Emotionales vs. Rationales im Software Engineering

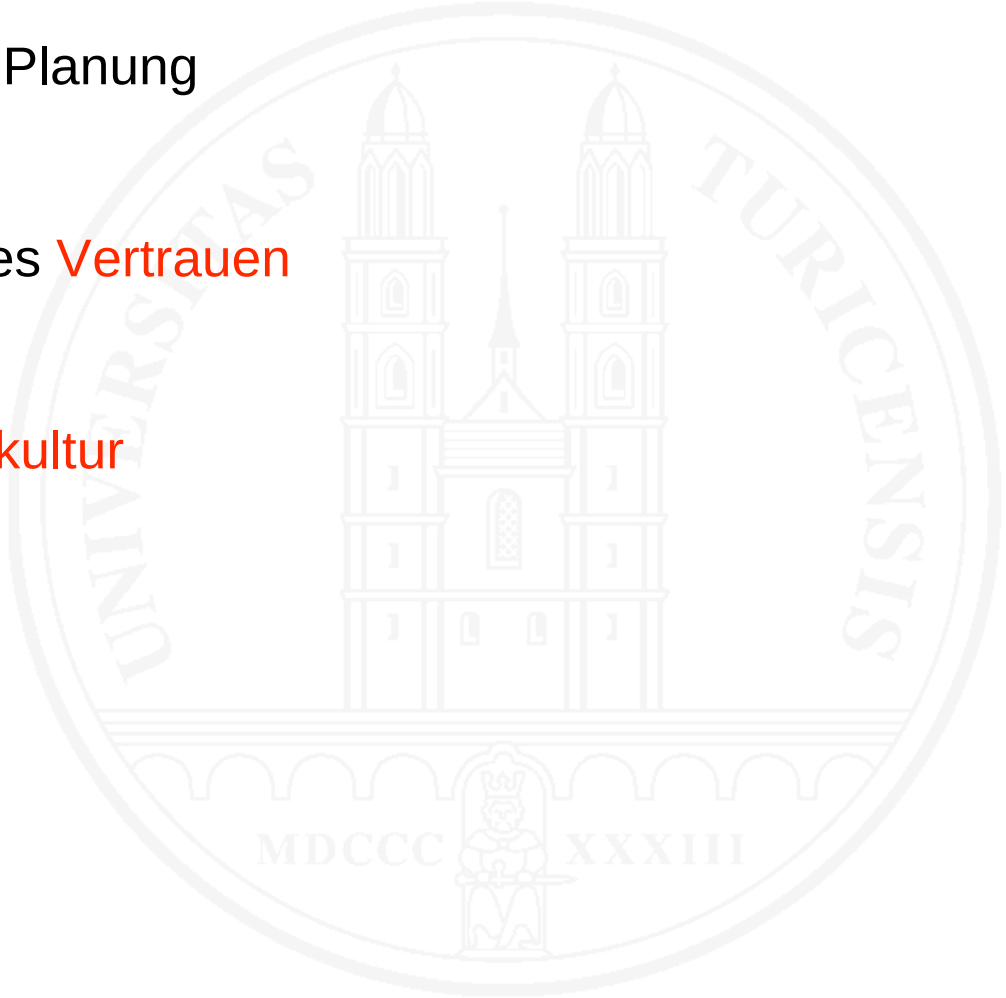
- Fühlen im Kleinen – Arbeiten im Großen
- Kreativität wollen – Disziplin benötigen
- Lustbetonte Arbeiten ≠ Notwendige Arbeiten
- Trägheitseffekte behindern Innovation
- Kurzfristiges Denken führt zu Fehlsteuerungen

Der Einfluss der Arbeitsumgebung

- Ausbildung
  - Gute Leute einstellen
  - Leute besser machen: Fortbildung, geeignete Gruppenprozesse
- Arbeitsplätze, an denen gearbeitet werden kann
  - Genug Platz
  - Technisch adäquat ausgerüstet
  - Wenig Störungen

Software Management – 1

- Realistische Planung
- Gegenseitiges Vertrauen
- Informationskultur



Software-Kultur

- Schaffung einer Software-Kultur
  - Wir sind kompetent, aber wir können nicht alles.
  - Wir sind schnell, aber wir versuchen nicht, die Lichtgeschwindigkeit zu ändern.
  - Wir tun die Dinge von Anfang an richtig.
  - Wir haben Spaß am professionellen Arbeiten.