

Vorlesung Software-Management

Sommersemester 2010

Grundlagen

Prof. Dr. K.-P. Fährnich / Thomas Riechert

06.04.2010

- **Termin: dienstags 9:15–10:45, Johannisgasse 26; Raum 1-22**
- Diplomstudiengang
 - Schwerpunktfach (Komplexprüfung am Ende des Studiums)
- Bachelor Studiengang alte Studienordnung
 - PL bzw. Alternative-PL mündlich. Prüfung am Ende
- **Aktuelle Studienordnung**
 - **Modul: 10–202-2319; Software Management**
- Ansprechpartner
 - Hr. Riechert <riechert@informatik.uni-leipzig.de>
 - Johannisgasse 16; Raum 5-09, dienstags 11:00 Uhr – 12:00 Uhr
 - Informationen und Folien zur Vorlesung
<http://bis.informatik.uni-leipzig.de/de/Lehre/0910/SS/SWM>

Sommersemester

- Modul: Betriebliche Informationssystem (Klaus-Peter Fähnrich)
- Modul: Software Management (Hans-Gert Gräbe)
- Modul: Semantic Web (Sören Auer)
- Modul: Softwaretechnik Praktikum (Bachelor Studiengang; Hans-Gert Gräbe)

Wintersemester

- Modul: Software aus Komponenten (Hans-Gert Gräbe)
- Modul: Softwaretechnik (Bachelor Studiengang; Klaus-Peter Fähnrich)

- Modulnummer: 10–202–2308
- Lehrveranstaltungen des Moduls
 - Vorlesungen
 - Betriebliche Informationssysteme (do 17-19 Uhr Jo26 Felix-Klein HS)
 - Einführung in XML (E-Learning, do 19-21 Uhr Jo26 Felix-Klein HS)
 - Seminare/Praktika
 - Praktikum BIS
 - Seminar/Praktikum Semantic Web
 - Prüfung
 - am Ende des Semesters

- Modulnummer: 10–202-2319
- Lehrveranstaltungen des Moduls
 - Vorlesungen
 - Software Management (di 9-11 Uhr Jo26; Raum 1-22)
 - Engineering IT-basierter Dienstleistungen (di 17-19 Uhr Jo26; Raum 1-22)
 - Software Qualitätsmanagement (mi 15-17 Uhr HS 14)
- Prüfung
 - mündlich 30 min; am Ende des Semesters

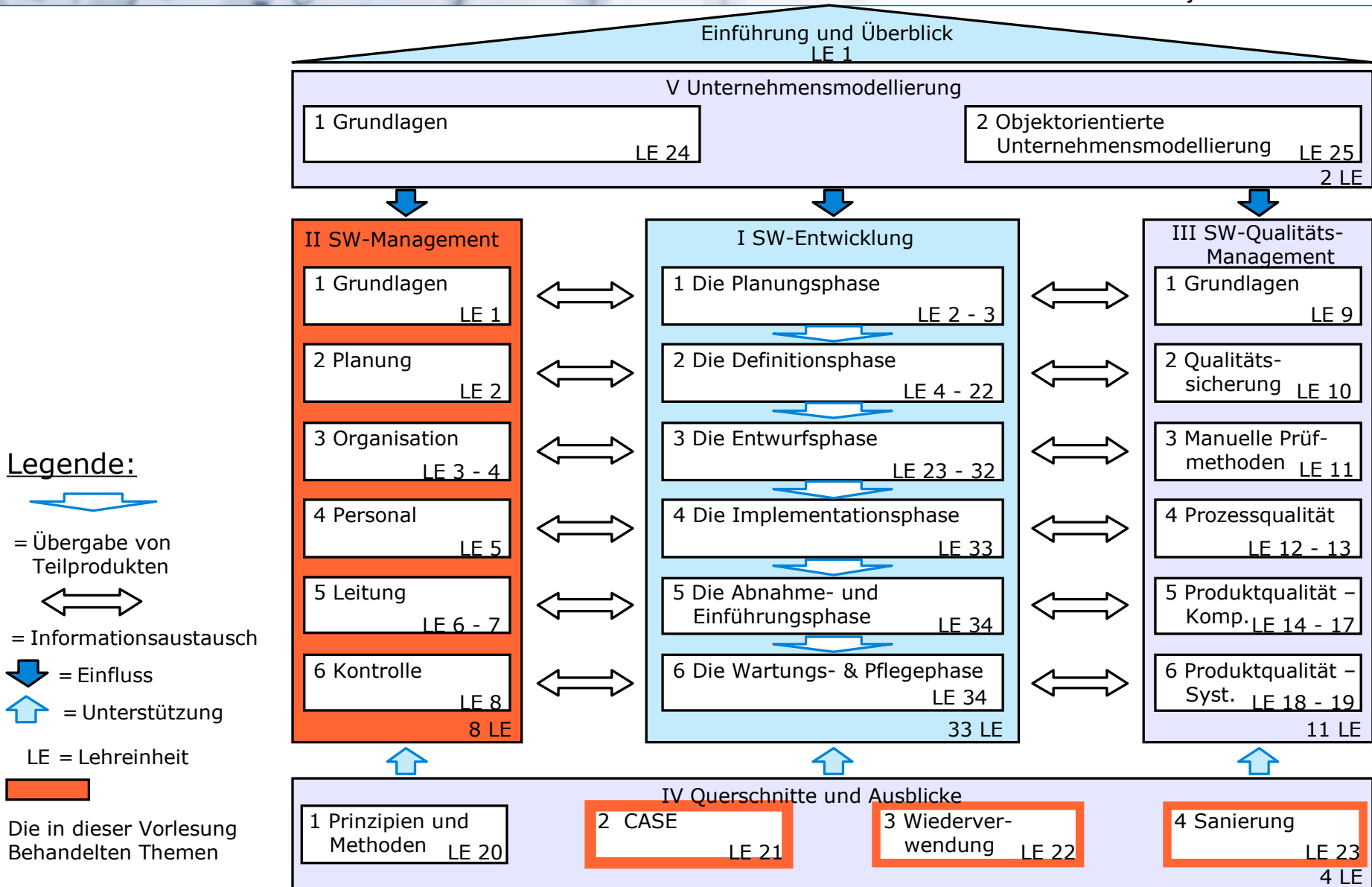
- Modulnummer: 10–202–2309
- Lehrveranstaltungen des Moduls
 - Vorlesungen
 - Semantic Web (di 13-15 Uhr Jo26 Felix-Klein HS)
 - Seminare/Praktika
 - Praktikum Semantic Web (mi 9-13 Uhr Jo26, Raum 1-22)
 - Seminar Semantic Web (do 11-13 Uhr, SG 1-10)
 - Prüfung
 - Klausur am Ende des Semesters

- Bewerbungsbogen unter <http://bis.informatik.uni-leipzig.de/>
- Ansprechpartner:
 - Hr. Riechert <rieichert@informatik.uni-leipzig.de>
 - Sprechzeiten: dienstags, 11:00-12:00, Jo26, Raum 5-09

- (1) **Grundlagen**
- (2) Planung
- (3) Organisation: Gestaltung
- (4) Organisation: Prozess-Modelle
- (5) Personal
- (6) Leitung
- (7) Innovationsmanagement
- (8) Kontrolle: Metriken, Konfigurations- und Änderungsmanagement
- (9) CASE
- (10) Wiederverwendung
- (11) Sanierung

Begleitliteratur: Helmut Balzert, Lehrbuch der Software-Technik

Quelle der Grafiken und Tabellen: Helmut Balzert, Lehrbuch der Software-Technik,
wenn nicht anders angegeben



- (1) **Einführung**
- (2) **Aufgaben**
- (3) **Produktivität**
- (4) **Einflussfaktoren der Produktivität**
- (5) **Produktivität und Qualität**
- (6) **Maßnahmen zur Produktivitätssteigerung**

- Erfolgreiche Software-Erstellung ist von der Güte des Software-Managements abhängig.
- **Ziele des Software-Managements:**
 - **Erhöhung der Produktivität,**
 - **Sicherstellung einer definierter Qualitätssicherung und**
 - **Senkung der Kosten.**
- Drei primäre Managementstrategien (nach [Grady92])
 - Maximierung der Kundenzufriedenheit,
 - Minimierung des Aufwands und der Zeit der Software-Erstellung,
 - Minimierung von Fehlern.

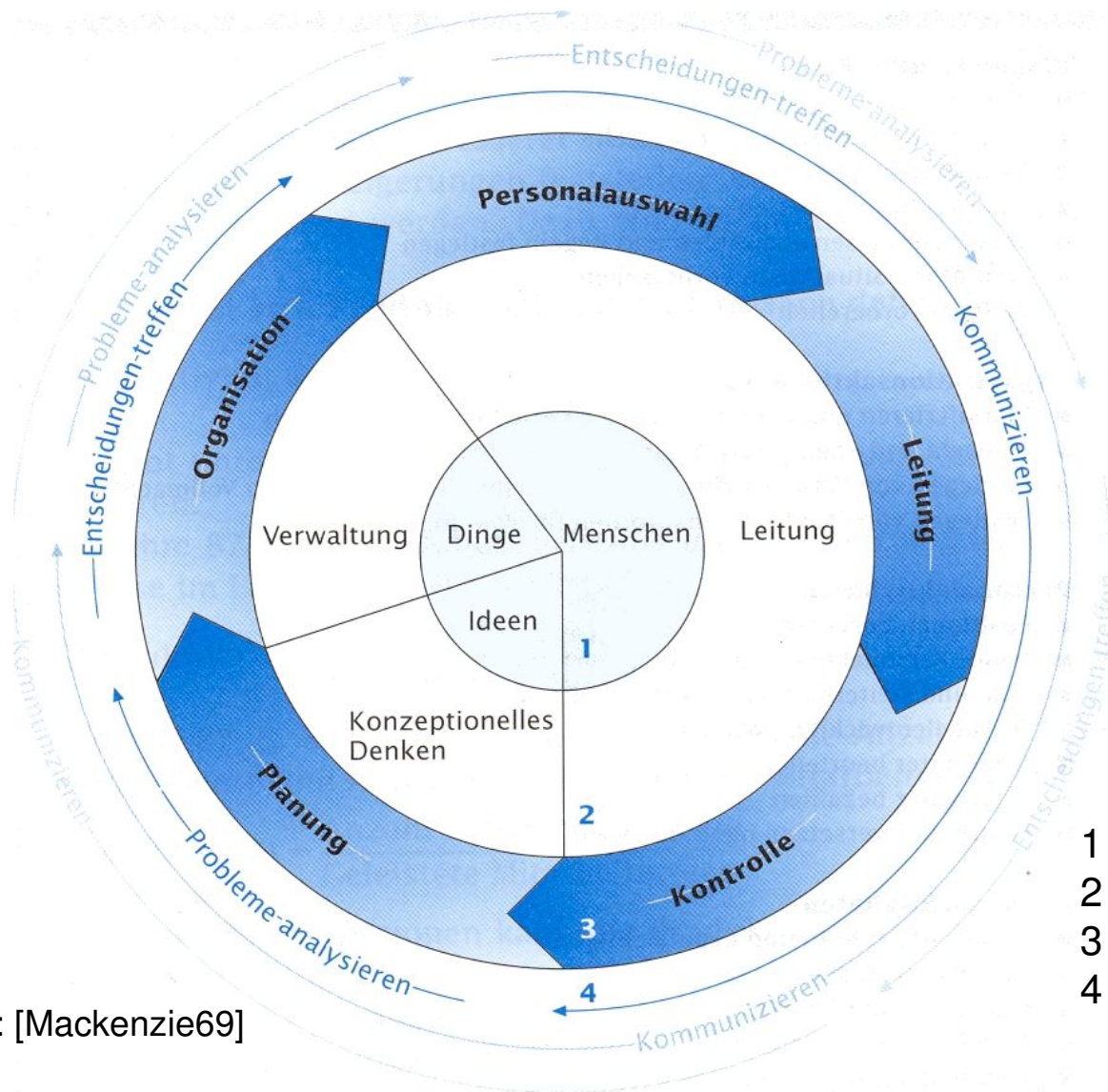
Hauptcharakteristika	Maximale Kundenzufriedenheit	Minimierung von Aufwand und Zeit	Minimierung von Fehlern
Hauptgeschäftsstrategie Effizient ...	Marktanteile erlangen ... beim ersten Markteinstieg	Konkurrenz erfordert neue Produkte oder Kostenkontrolle ... bei mehreren Konkurrenzprodukten oder wenn man profitablere Produkte verkauft	Halten oder Vergrößern des Marktanteils ... bei Konkurrenzfähigen Eigenschaften und falls ein adäquater Marktanteil gehalten wird
Charakteristische Eigenschaften	Kommunikation mit dem Kunden, schnelle Reaktion	Fokus auf Auslieferungsdatum und Aufwand	Analyse und Entfernen von Fehlerursachen

Quelle: [Grady92, S.24]

- Unterschied zwischen Software-Management und Management anderer Ingenieurbereiche [Sneed87]:
 - Das Produkt ist immateriell.
 - Der Entwicklungsfortschritt ist objektiv nicht zu ermitteln.
 - Eine Software-Entwicklung verläuft nicht-deterministisch.
 - Es gibt noch kein klares Verständnis vom Entwicklungsprozess.
 - Große Software-Systeme tendieren dazu, einmalige Entwicklungen zu sein.
 - Unteilbarkeit der Arbeit.
 - Die Software-Technik ist keine Naturwissenschaft.
 - Hoher Grad an Abstraktion, bei gleichzeitig niedrigem Grad an Normierung.

- (1) **Einführung**
- (2) **Aufgaben**
- (3) **Produktivität**
- (4) **Einflussfaktoren der Produktivität**
- (5) **Produktivität und Qualität**
- (6) **Maßnahmen zur Produktivitätssteigerung**

- Manager befasst sich mit drei grundlegenden Elementen:
 - Ideen, Dinge und Menschen
- Organisation benötigt einen
 - Planer, einen Verwalter und einen Leiter.
- *„Management umfasst alle Aktivitäten und Aufgaben, die von einem oder mehreren Managern, durchgeführt werden, um die Aktivitäten von Mitarbeitern zu planen und zu kontrollieren damit ein Ziel oder der Abschluss einer Aktivität erreicht wird, die durch die Mitarbeiter alleine nicht erreicht werden können “*
[Mackenzie69]



- 1 Element
- 2 Aufgaben
- 3 Sequentielle Funktionen
- 4 Kontinuierliche Funktionen

Quelle: [Mackenzie69]

- Ziele setzen,
- Strategien und Taktiken entwickeln,
- Termine festlegen,
- Entscheidungen treffen,
- Vorgehensweisen auswählen und Regeln festlegen,
- Zukünftige Situationen vorhersehen

Planungsaktivitäten

Organisationsaktivitäten

- Identifizieren und Gruppieren der zu erledigenden Aufgaben,
- Auswahl und Etablierung organisatorischer Strukturen,
- Festlegen von Verantwortungsbereichen und disziplinarischen Vollmachten,
- Festlegen von Qualifikationsprofilen für Positionen

- Positionen besetzen,
- Neues Personal einstellen und integrieren,
- Aus- und Weiterbildung von Mitarbeitern,
- Personalentwicklung planen,
- Mitarbeiter beurteilen, bezahlen,
- Mitarbeiter versetzen und entlassen

Personalaktivitäten

Quelle: [Thayer90, S.17]

Leitungsaktivitäten

- Mitarbeiter führen und beaufsichtigen,
- Kompetenzen delegieren,
- Mitarbeiter motivieren,
- Aktivitäten koordinieren,
- Kommunikation unterstützen,
- Konflikte lösen,
- Innovationen einführen

- Prozess- und Produktstandards entwickeln,
- Berichts- und Kontrollwesen etablieren,
- Prozesse und Produkte vermessen,
- Korrekturaktivitäten vermessen,
- Loben und Tadeln

Kontrollaktivitäten

Quelle: [Thayer90, S.17]

(1) Einführung

(2) Aufgaben

(3) **Produktivität**

(4) Einflussfaktoren der Produktivität

(5) Produktivität und Qualität

(6) Maßnahmen zur Produktivitätssteigerung

- **Produktivitätserhöhung der Software-Erstellung ist das zentrale Ziel des Software-Managements.**
- Produktivitätssteigerung bedeutet [Wallmüller90, S.57f.]:
 - Software-Produkte in kürzeren Zeitintervallen entwickeln;
 - Software-Produkte so entwickeln, dass sie einen höheren **Return on Investment** liefern;
 - Software-Produkte mit höherer Qualität entwickeln.

Produktivität lässt sich definieren:

1) **Produktivität = Leistung/Aufwand**

Unterschiedliche Ansätze um Leistung und Aufwand zu definieren,

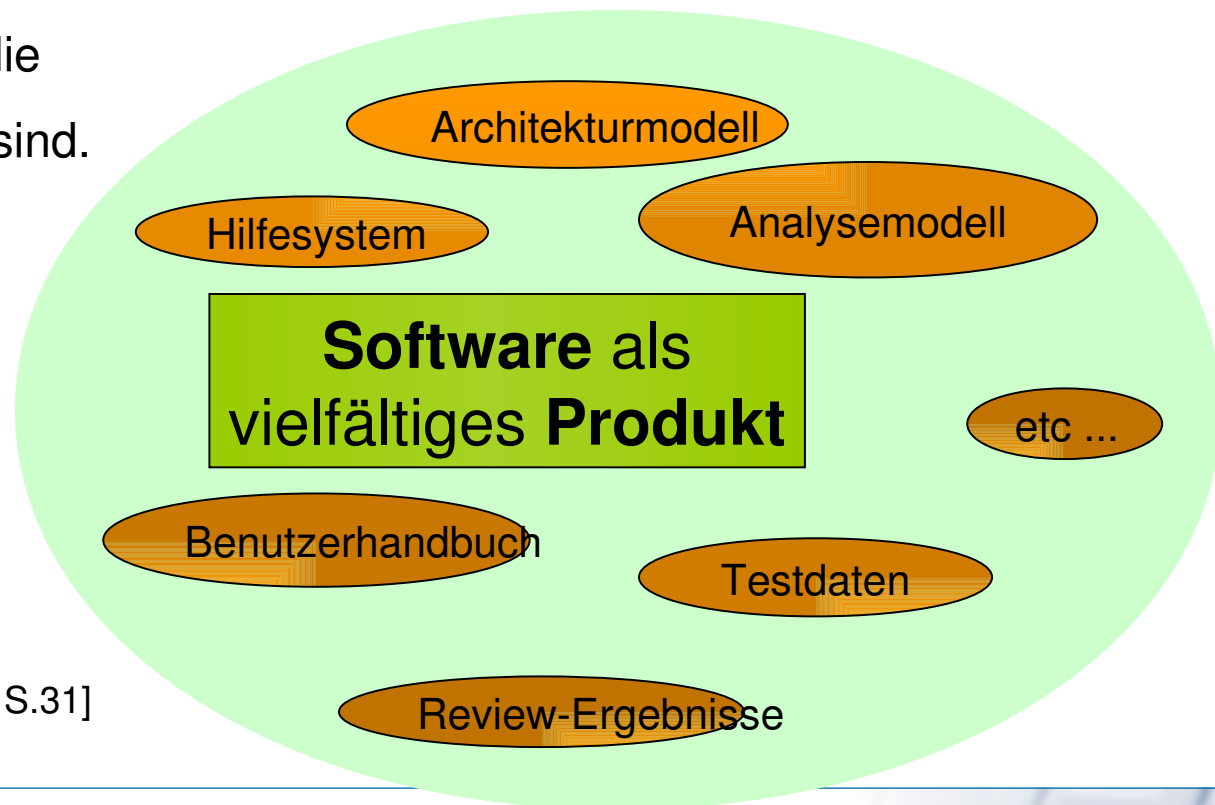
- [Boehm87, S.44]:

2) **Produktivität = Produzierte Ergebnisse/Eingesetzter Aufwand**

- [Sneed87, S.31]:

3) **Produktivität = Anzahl SW-Elemente/Geleistete Mitarbeitertage**

- Verbesserung der Produktivität, wenn:
 - Vermehrung der Ergebnisse,
 - Verringerung des Aufwands oder
 - Vermehrung der Ergebnisse und Verringerung des Aufwands
- Problem bei Definitionen 2) und 3) besteht darin, geeignete bzw. Elemente zu definieren, die quantifizierbar sind.



Quelle: [Sneed87, S.31]

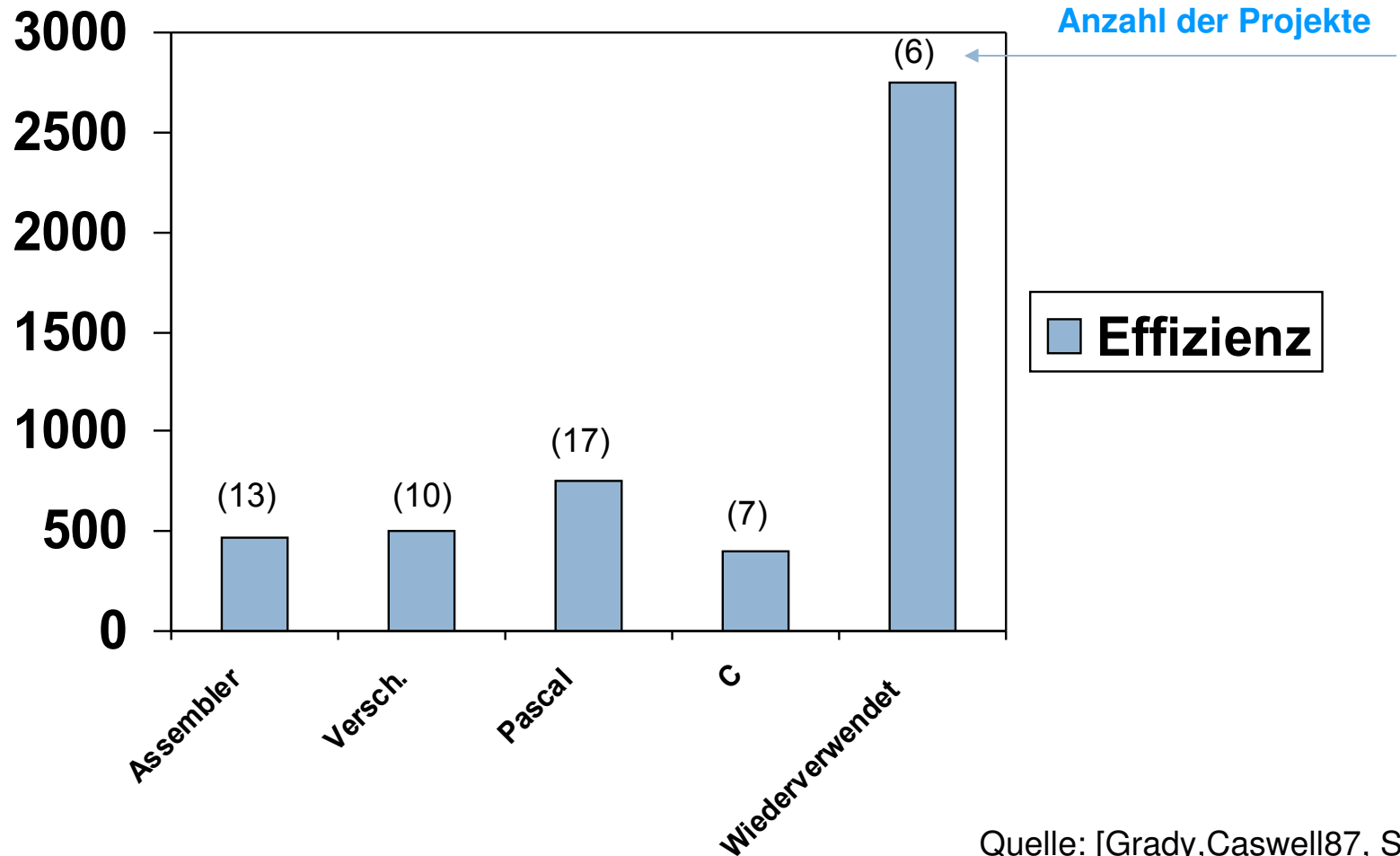
- Verbesserung der Produktivität, wenn:
 - Vermehrung der Ergebnisse,
 - Verringerung des Aufwands oder
 - Vermehrung der Ergebnisse und Verringerung des Aufwands
- Problem bei Definitionen 2) und 3) besteht darin, geeignete bzw. Elemente zu definieren, die quantifizierbar sind.

2) Produktivität= Produzierte Ergebnisse/Eingesetzter Aufwand

3) Produktivität= Anzahl SW-Elemente/Geleistete Mitarbeitertage

- Grobe Messungen:
 - Konventionell: Anzahl der Quellprogrammzeilen (LOC, Lines of Code)
 - Firma HP als Teil ihres Produktivitätsmaßes: non commented source statements (NCSS)
- Function Points

- Durchschnittliche NCSS / Ingenieurmonate / Sprache



Quelle: [Grady,Caswell87, S.22]

Aufwand für die Software-Entwicklung setzt sich zusammen aus:

- Personalkosten (enorm hoch),
- Kosten für Computerressourcen,
- Kosten für Hilfsmittel

Metrik der geistigen Anwesenheit [DeMarco,Lister91]:

Umweltfaktor = ungestörte Stunden / Stunden körperlicher Anwesenheit

Nach Basili ist Software-Produktion ein Prozess der Wertschöpfung:

Produktivität = Produktwert / Kosten

[Grady,Caswell87, S.3]

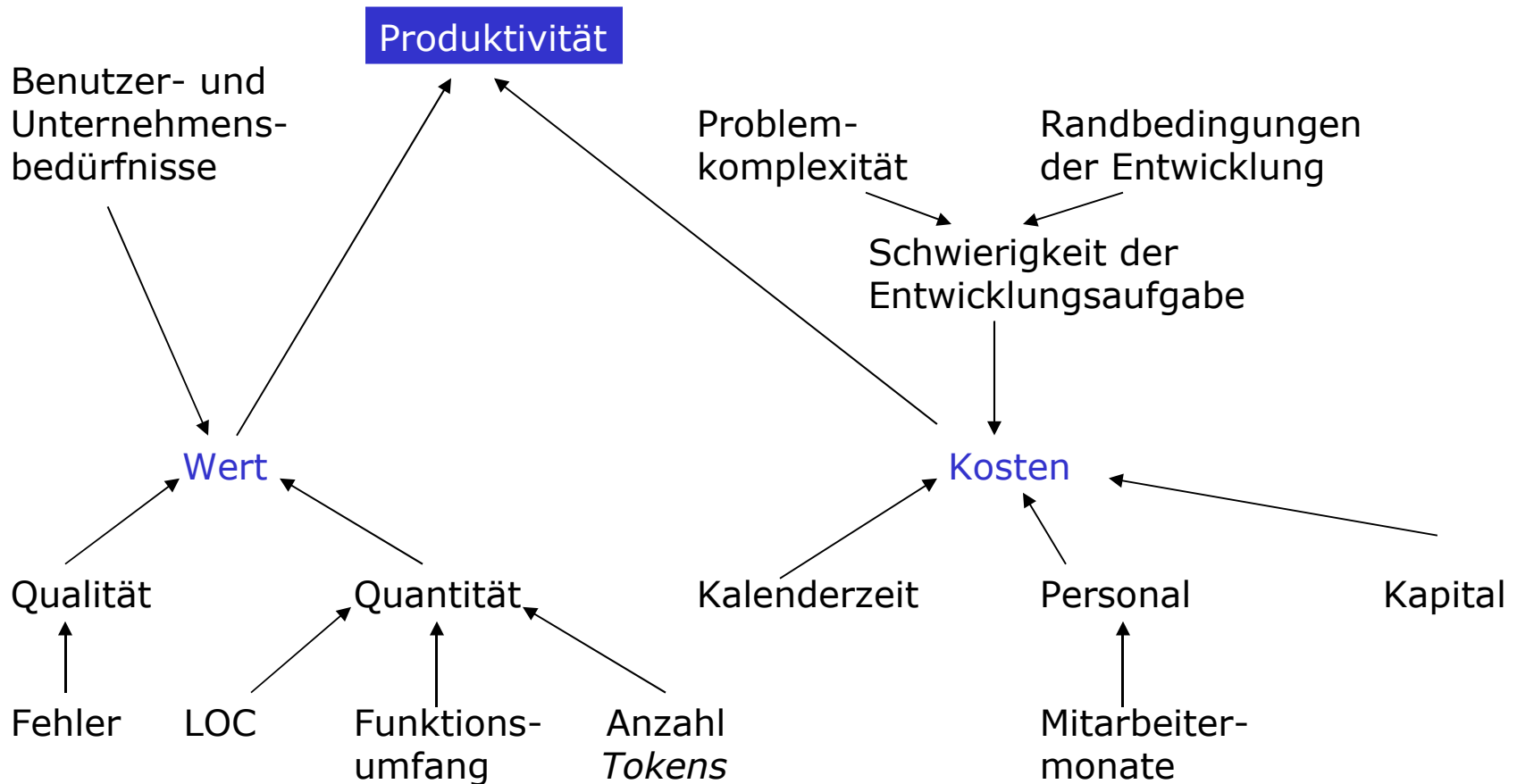
Problem: Wie kann man den Produktwert ermitteln bzw. frühzeitig schätzen.

Beste Produktivitätsmetrik [Maxwell,Wassenhove,Dutta96]:

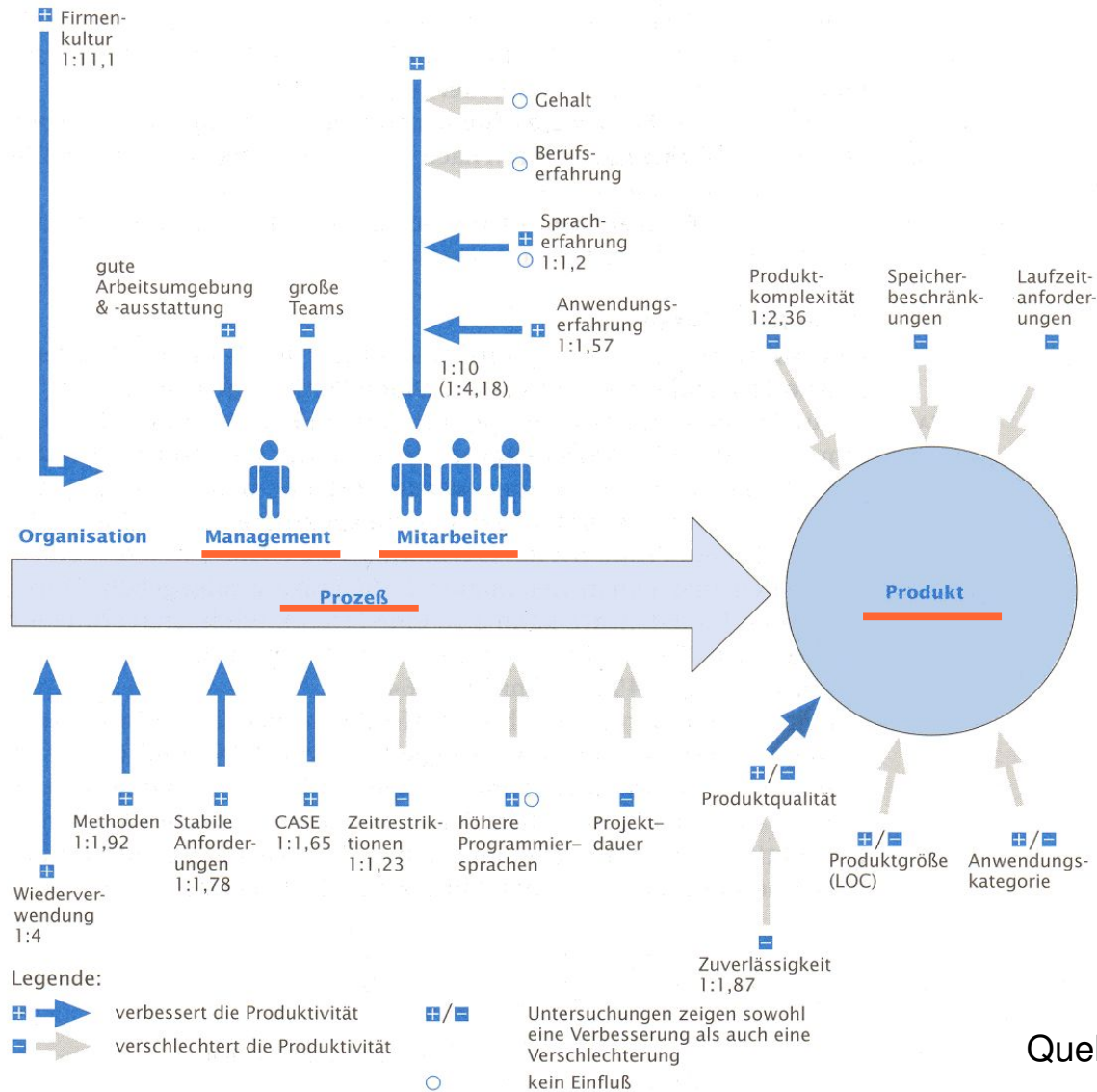
Produktivität = LOC / Aufwand in Mitarbeitermonate

Produktivität und ihre Einflussfaktoren nach Basili

Quelle: [Grady,Caswell87, S.3]



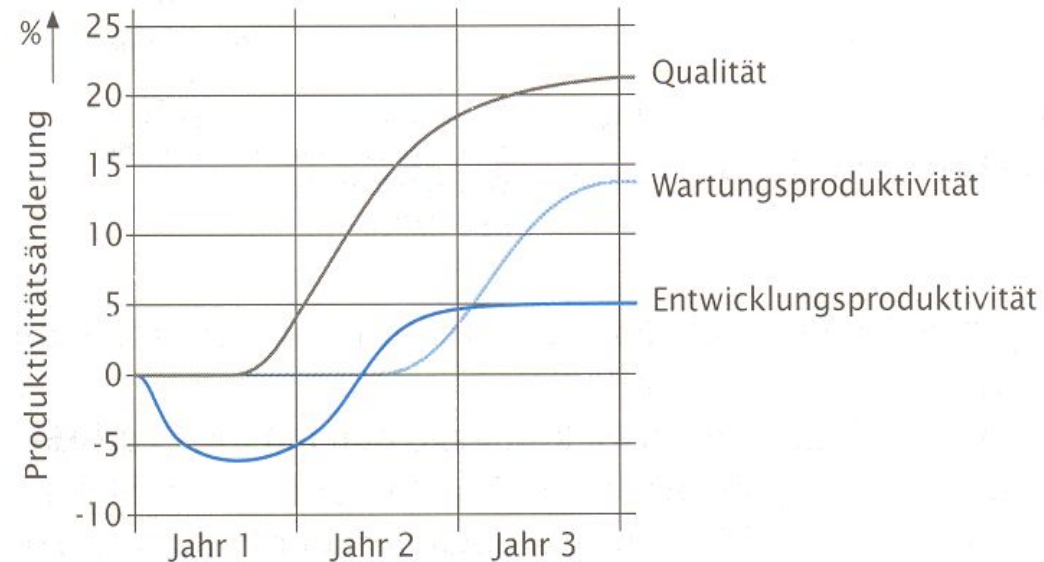
- (1) Einführung**
- (2) Aufgaben**
- (3) Produktivität**
- (4) Einflussfaktoren der Produktivität**
- (5) Produktivität und Qualität**
- (6) Maßnahmen zur Produktivitätssteigerung**



Quelle: [Balzert198, S.13]

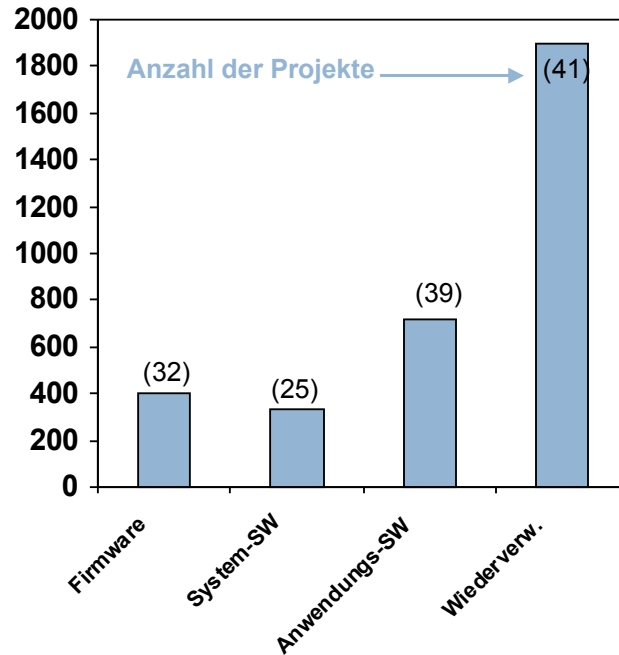
- Die Produktkomplexität beeinflusst wesentlich die Produktivität (Verhältnis 1:2,36 [Boehm87]).
- Zur Zeit existiert keine befriedigende Metrik, die es erlaubt, die Komplexität eines Software-Produktes in eine Zahl zu fassen.
- Produktkomplexität ist nur bedingt vom Management zu beeinflussen. Wesentlich erscheint die Verwendung der passenden Methoden.
- Produktgröße (LOC) hat einen großen Einfluss auf die Entwicklungskosten (Aufwand nimmt überproportional mit der Produktgröße zu).
- Geforderte Produktqualität hat hohen einen Einfluss auf die Produktivität (Verhältnis 1:1,87 [Boehm87]).
- Zunehmende Langzeitanforderungen und zunehmende Speicherungsbeschränkungen führen zu sinkender Produktivität [Maxwell,Wassenhove,Dutta96].

- Produktivitätsverhältnis von 1:1,65 zwischen einem minimalen und einem maximalen CASE-Einsatz [Boehm87].
- Unterscheidung zwischen Entwicklungs- und Wartungsproduktivität ist sinnvoll (siehe Grafik).
- Kritischer Erfolgsfaktor ist nicht CASE-Einführung, sondern die Methoden-Einführung (neue Denkweise der MAs).
- Durch intensiven Einsatz von Werkzeugen und Methoden wird hohe Produktivität erreicht.



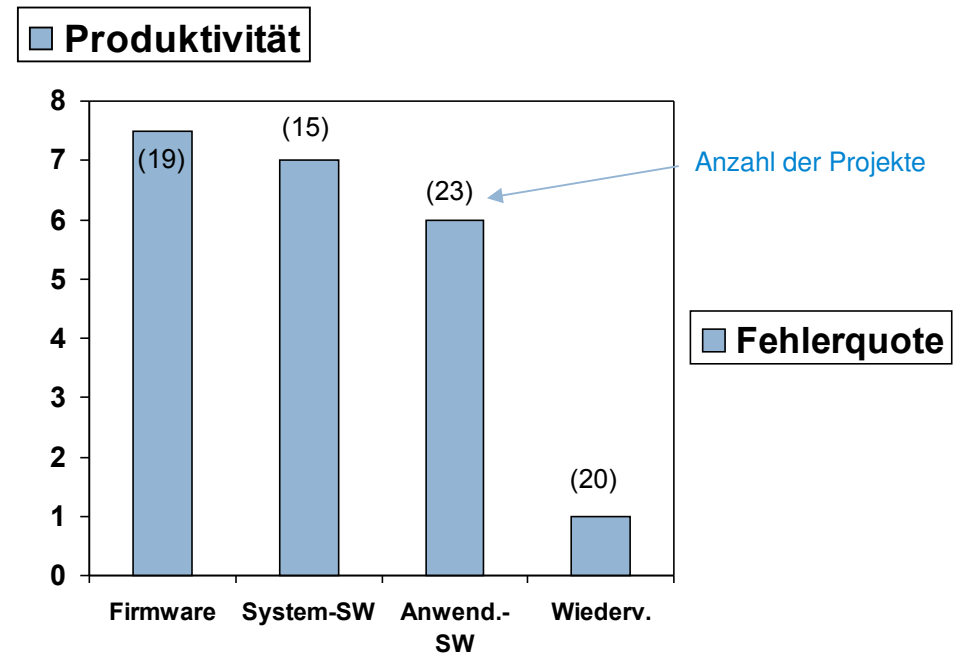
Quelle: [Jones92, S.40]

- Durchschnittliche NCSS / Ingenieurmonate



Quelle: [Grady92, S.43]

- Durchschnittliche Fehler / kNCSS



Quelle: [Grady,Caswell87, S.112]

Drei Grundregeln:

- Die besten Mitarbeiter sind um Faktor 10 besser als die schlechtesten;
- Die besten Mitarbeiter sind 2,5 mal besser als der Durchschnitt;
- Die überdurchschnittlichen MA übertreffen die unterdurchschnittlichen im

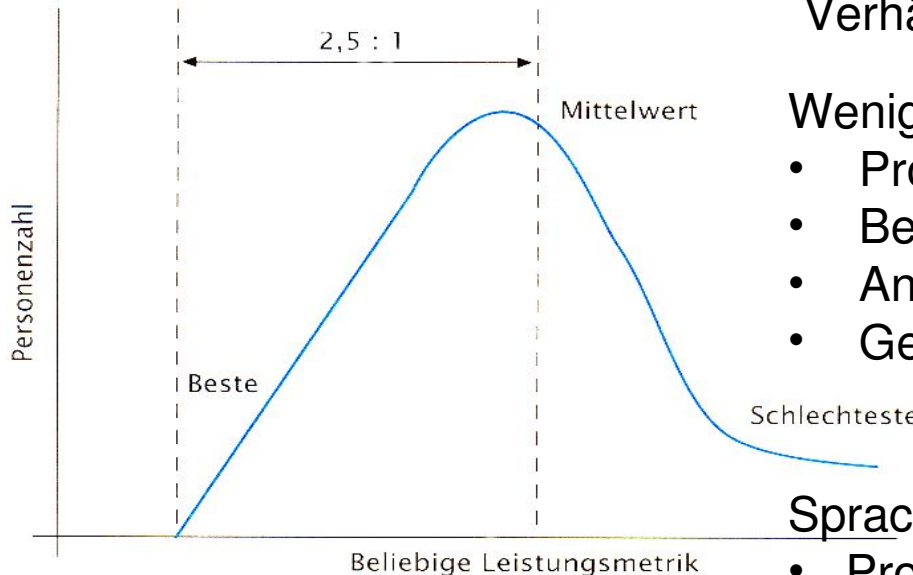
Verhältnis 2:1.

Wenig oder keine Korrelation mit Leistung:

- Programmiersprachen,
- Berufserfahrung,
- Anzahl der Fehler,
- Gehalt

Spracherfahrung und Anwendungserfahrung:

- Programmiersprache im Verhältnis 1:1,2,
- Anwendungsgebiet im Verhältnis 1:1,57



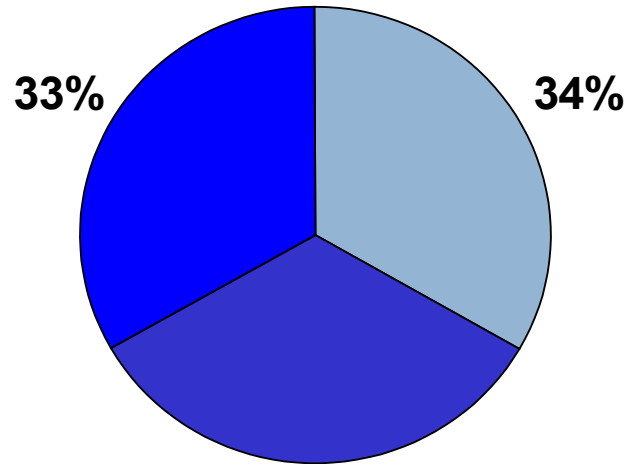
Grafik: Produktivitätsunterschiede zwischen den einzelnen MA;

Quelle: [DeMarco, Lister87, S.52]

- Softwaremanagement hat Einflüsse auf Mitarbeiter und Entwicklungsprozess.
- Einfluss auf die Produktivität haben auch physische Arbeitsumgebung und Arbeitsplatzausstattung/-gestaltung.
- Mitarbeiter umso produktiver je:
 - ruhiger sein Arbeitsplatz ist,
 - weniger er gestört wird,
 - besser die Privatsphäre gewahrt,
 - größer der Arbeitsplatz.
- Die Gestaltung der Firmenkultur beeinflusst indirekt die Produktivität und wesentlich die Einführung der Innovationen.
- Mit wachsender Teamgröße sinkt die Produktivität.

- Zeitaufteilung eines Software-Ingenieurs

Projektfremde Aktivitäten
(Training, Präsentationen,
Reisen)



Mündliche Aktivitäten
(Besprechungen,
Diskussionen, *Reviews*)

33%
Ruhige, konzentrierte Aktivitäten

Quelle: [Grady,Caswell87, S.37]

(1) Einführung

(2) Aufgaben

(3) Produktivität

(4) Einflussfaktoren der Produktivität

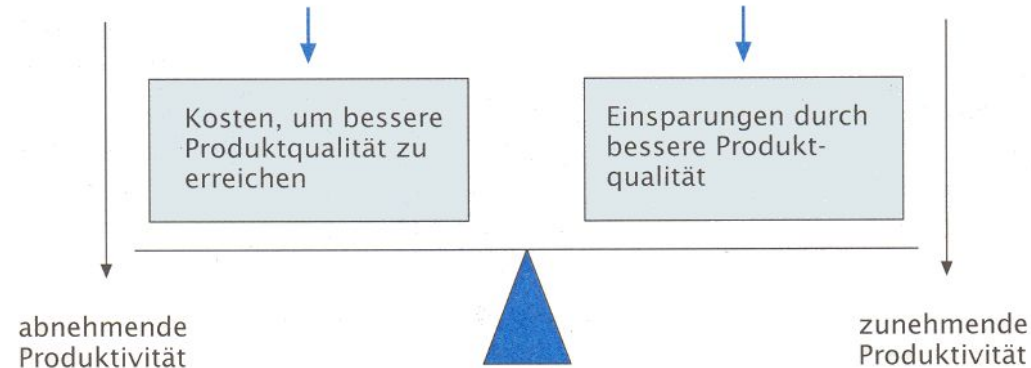
(5) **Produktivität und Qualität**

(6) Maßnahmen zur Produktivitätssteigerung

Produktivität und Qualität beeinflussen sich.

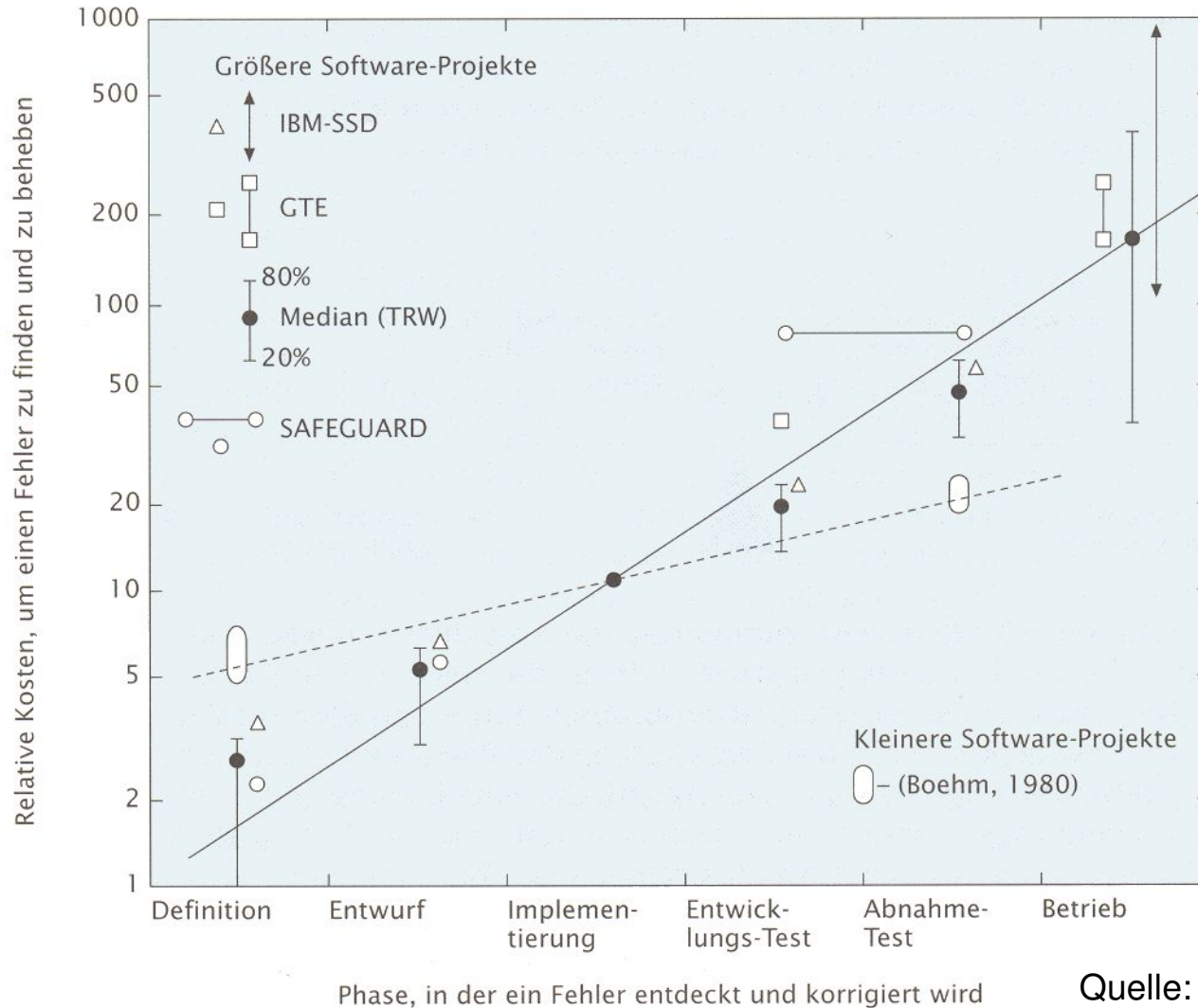
Zwei konträre Auffassungen:

- Hohe Qualitätsanforderungen verringern die Produktivität.
- Hohe Qualitätsanforderungen verbessern die Produktivität.



Quelle: [Balzert98, S.21]

- 2/3 aller Lebenszykluskosten entfallen auf die Wartung und Pflege und nur 1/3 auf die eigentliche Entwicklung.
- Probleme bei den Einsparungen:
 - Wie quantifiziert man Software-Qualität?
 - Wer bezahlt die Wartung?
 - Software-Qualität schlecht quantifizierbar und messbar.
- Als pragmatischer Ansatz werden die Wartungskosten ermittelt.



Quelle: [Boehm81, S.40]

(1) Einführung

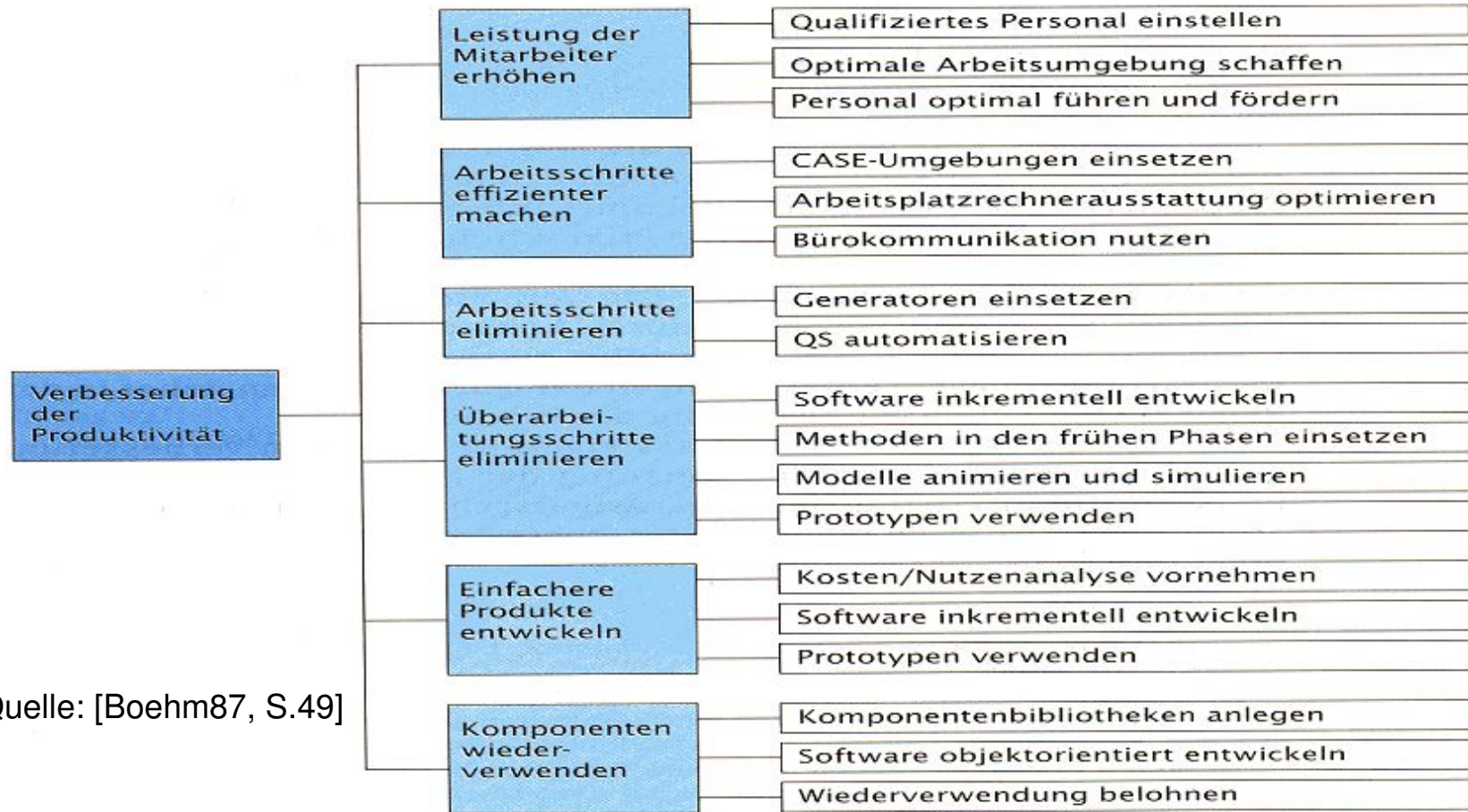
(2) Aufgaben

(3) Produktivität

(4) Einflussfaktoren der Produktivität

(5) Produktivität und Qualität

(6) **Maßnahmen zur Produktivitätssteigerung**



Quelle: [Boehm87, S.49]

1) **Produktivität = Leistung/Aufwand**

2) **Produktivität = Produzierte Ergebnisse/Eingesetzter Aufwand**

3) **Produktivität = Anzahl SW-Elemente/Geleistete Mitarbeitertage**

[DeMarco,Lister91]

DeMarco T., Lister T., Wien wartet auf Dich! Der Faktor Mensch im DV-Management, 1991

[Grady92]

Grady R.B., Practical Software Metrics for Management and Process Improvement, 1992

[Mackenzie69]

Mackenzie R.A., The management process in 3-D, 1969

[Maxwell,Wassenhove,Dutta96]

Maxwell K.D, Wassenhove L.V., Dutta S., Software Development Productivity of European SpaceA, Military, and Industrial Applications, 1996

[Sneed87]

Sneed H.M., Software-Management, 1987

[Wallmüller90]

Wallmüller E., Software-Qualitätssicherung in der Praxis, 1990

Begleitliteratur: Helmut Balzert, Lehrbuch der Software-Technik
Quelle der Grafiken und Tabellen: Helmut Balzert, Lehrbuch der Software-Technik,
wenn nicht anders angegeben