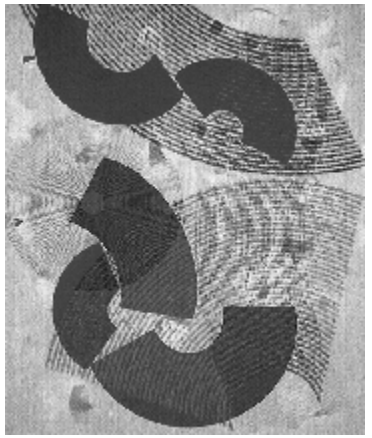
Software-Management

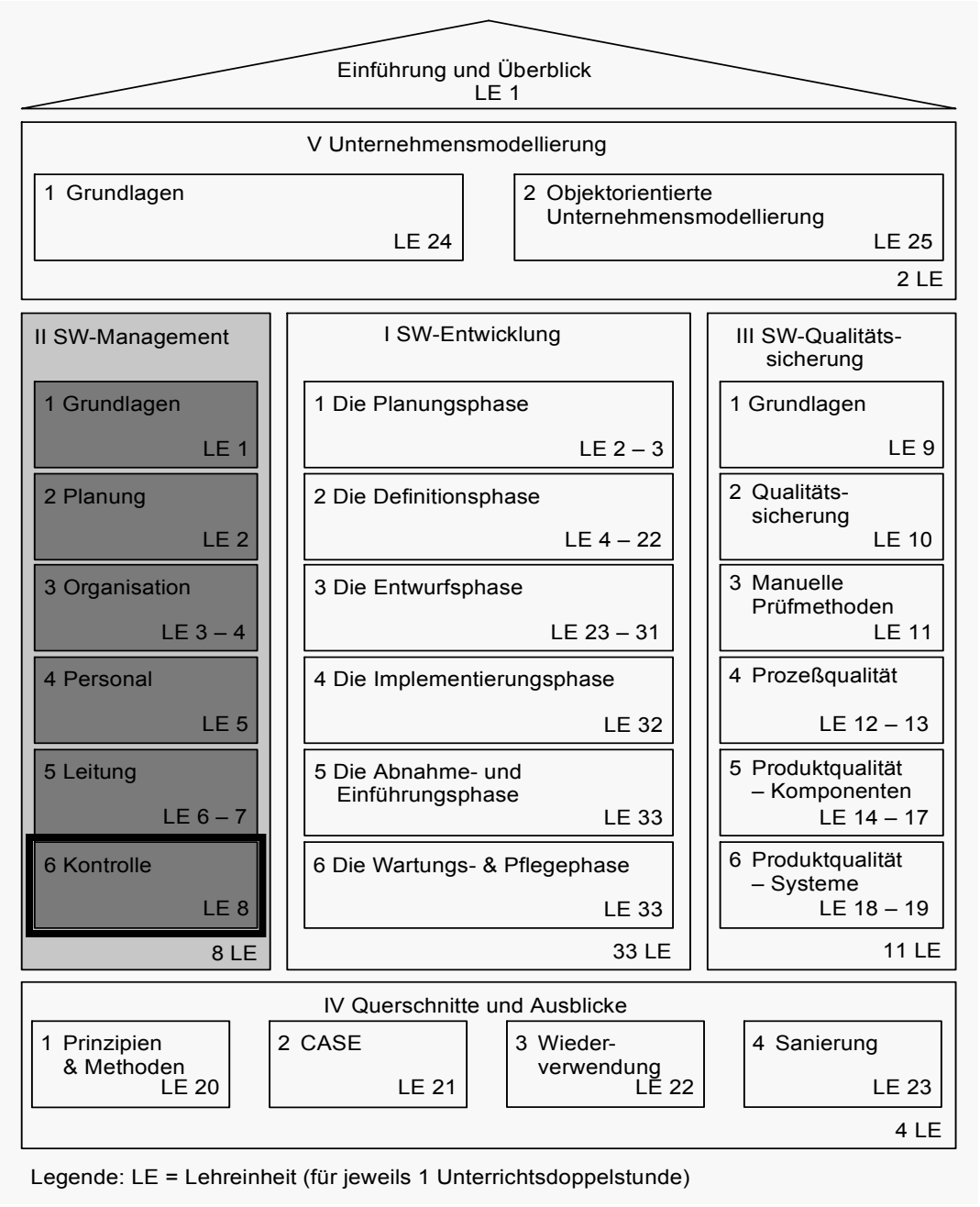
6 Kontrolle

Prof. Dr. Joachim Hertel



II Software-Management - Kontrolle

LE 8
2



Legende: LE = Lehreinheit (für jeweils 1 Unterrichtsdoppelstunde)

Lernziele

- Anhand von Beispielen Konfigurationen beschreiben und Versionszählungen durchführen können
- Anhand von Szenarien Konfigurations- und Änderungsmanagement exemplarisch durchführen können
- Gegebene Metriken klassifizieren und auf Gütekriterien hin überprüfen können
- Das verwendete Projektplanungssystem für die Projektkontrolle einsetzen können
- Das *Checkin/Checkout*-Modell auf Beispiele anwenden können.

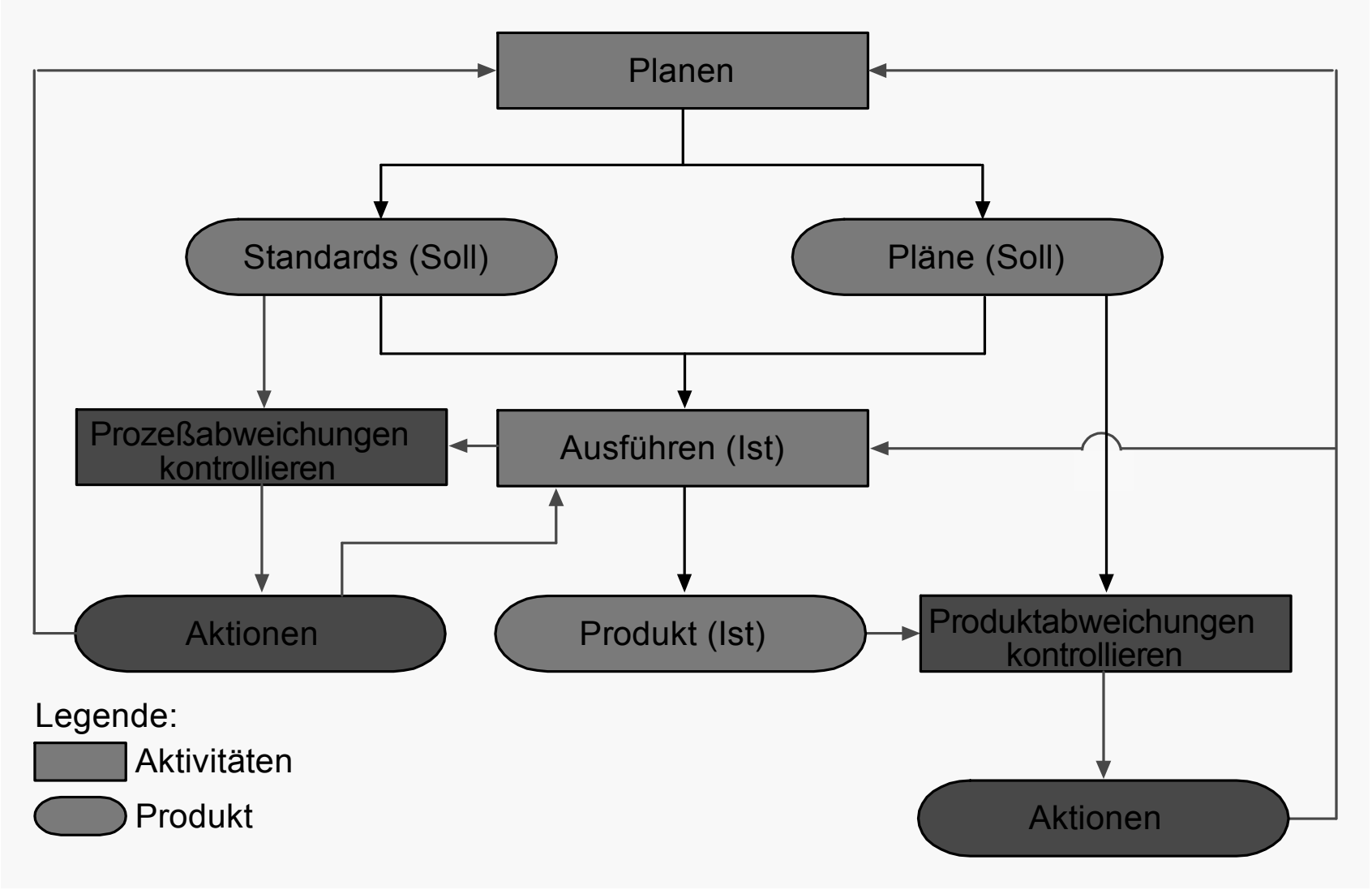
Inhalt

6.1 Grundlagen

- **Kontrolle**
  - **Managementaktivitäten, die sicherstellen, daß die laufenden Tätigkeiten mit dem Plan übereinstimmen**
  - **Rückkopplungssystem (Soll – Ist)**
- **Kontrollprozeß**
  - **Einrichten von Plänen und Standards**
  - **Messen der Ausführung gegen diese Pläne und Standards**
  - **Korrektur der Abweichungen.**

6.1 Grundlagen

♦ Der prinzipielle Kontrollprozeß



6.1 Grundlagen

- **Probleme**
  - **Kontrollmethoden nehmen die bisher benötigte Zeit und die entstandenen Kosten als Maßstab**
  - **Standards für Entwicklungsaktivitäten**
    - **nicht vorhanden**
    - **nicht schriftlich fixiert**
    - **nicht durchgesetzt**
  - **Software-Meßtechnik nicht ausreichend**
    - **Fehlende Software-Maße (Metriken) über den Entwicklungsprozeß und das Produkt.**

6.1 Grundlagen

- **Aufgaben des Software-Managements**
  1. Standards entwickeln und festlegen
  2. Kontroll- und Berichtssystem etablieren
  3. Prozesse und Produkte vermessen
  4. Korrigierende Aktionen initiieren
  5. Loben und Tadeln.

6.1 Grundlagen

1 Standards entwickeln und festlegen

- Entwickeln von Quantitäts- und Qualitätsstandards
- Festlegen des Prozeßmodells
- Festlegen von Qualitätssicherungsmethoden
- Entwickeln von
  - Produktivitätsmetriken
  - Qualitätsmetriken
  - Prozeßmetriken.

6.1 Grundlagen

- **Vorteile von Standards**
  - + **Einarbeitungs- / Umschulungskosten sinken**
  - + **Kommunikation zwischen Teammitgliedern wird verbessert**
  - + **Personalaustausch zwischen Projekten wird erleichtert**
  - + **Erfahrungen können besser weitergegeben werden**
  - + **Erfahrungen erfolgreicher Projekte können einheitlich angewandt werden**
  - + **Wartung und Pflege werden vereinfacht**
  - + **Standards können kontrolliert werden.**

6.1 Grundlagen

2 Kontroll- und Berichtssystem etablieren

- Kontroll- und Berichtssysteme auswählen...
  - um den Entwicklungsprozeß zu überwachen
  - um den Entwicklungsstatus zu bestimmen
- Parameter für Art / Umfang der Systeme hängen ab von ...
  - verwendetem Führungsstil
  - verwendeter Aufbau- und Ablauforganisation (Prozeßmodelle)
  - dem Umfang der Entwicklung (Zeit, Anzahl, Mitarbeiter)
  - der eingesetzten CASE-Umgebung.

6.1 Grundlagen

- **Kontroll- und Berichtssysteme**
  - **Kontrolle des Prozesses**
    - **Budgetüberprüfungen**
    - **Meilensteinüberprüfungen**
    - **Verfolgung der Top 10-Risiken**
    - **Qualitätssicherung**
  - **Kontrolle des Produkts**
    - **Konfigurationsmanagement**
    - **Qualitätssicherung.**

Berichtstypen

- **Budgetbericht**
  - Vergleicht das Budget mit den Ausgaben und hilft neue Budgetschätzungen vorzunehmen
- **Terminübersicht**
  - Vergleicht Terminstatus mit fertiggestellten Meilensteinen
- **Mitarbeiterstunden / Aktivitäten-Bericht**
  - Anzahl der Stunden, die benötigt wurden, um eine Aktivität durchzuführen
- **Mitarbeitertage / Aufgabenbericht**
  - Zeigt die für eine Aufgabe benötigten Tage.

Berichtstypen

- **Meilensteinfälligkeitsbericht**
  - **Status über die erreichten und überfälligen Meilensteine (mit Gründen)**
- **Projektfortschrittsbericht**
  - **Projektfortschritt (nicht formalisiert) oder eine Liste der erledigten Aktivitäten**
- **Aktivitätenbericht**
  - **in der Periode erledigte Aktivitäten.**

Berichtstypen

- **Trenddiagramm**
  - Trends wie Budgettrend, Krankenstand usw.
  - Dient dazu, die Zukunft vorherzusagen
- **Änderungsbericht**
  - Ausnahmen vom Plan und signifikante positive und negative Veränderungen
  - Entwicklungsrückstand
- **Top 10-Risikoelementliste**
  - Periodischer Bericht, der pro Periode die 10 Risikoelemente mit den größten Risiken angibt.

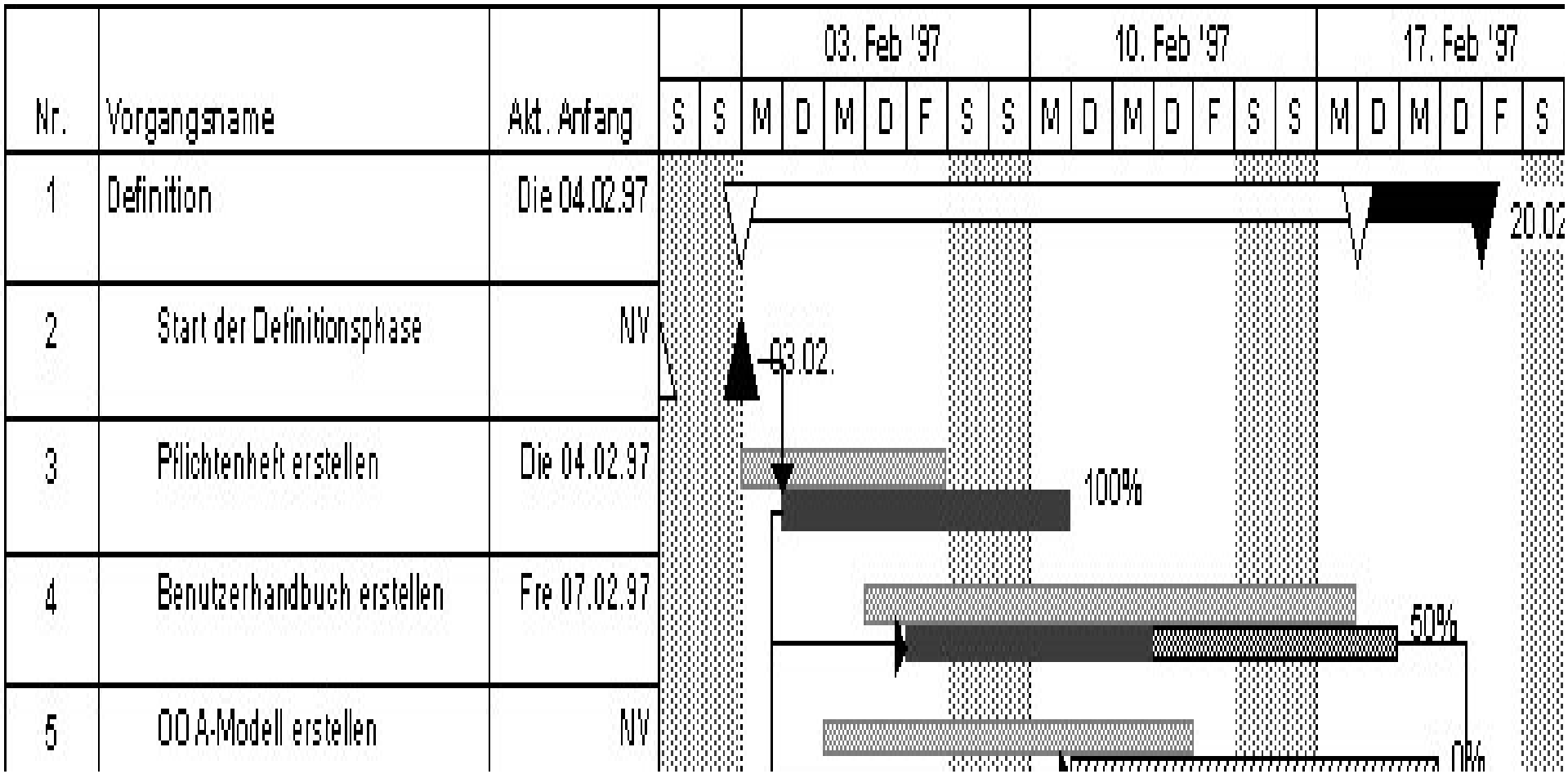
Projektstatistik		Monat/Jahr:	
Projekt:		Mitarbeiter:	
Phase	Phasenplanung	Phasenrealisierung	Phasenüberprüfung
Planung			
Definition			
Entwurf			
Implemen–tierung			
Abnahme & Einführung			
Wartung & Pflege			
Projektleitung			
Schulung, Einarbeitung			
Dienstreisen, Tagungen,			

LE 8
17

6.1 Grundlagen

• Berichtstypen eines Planungssystems (1)

Projektfortschrittbericht (siehe auch Abb. 2.5-6)



6.1 Grundlagen

• Berichtstypen eines Planungssystems (2)

erechnet:	376h	Verbleibend:	2
eplant:	0h	Aktuell:	1
bweichung:	376h	% Abgeschlossen:	4

osten

erechnet:	66,760.00 DM	Verbleibend:	51,160.00
eplant:	0.00 DM	Aktuell:	15,600.00
bweichung:	66,760.00 DM		

organgsstatus		Ressourcenstatus	
och nicht begonnene Vorgänge:	4	Ressourcen:	
organge in Arbeit:	2	Überlastete Ressourcen:	
organge abgeschlossen:	1		
esamtzahl Vorgänge:	7	Gesamtzahl Ressourcen:	

6.1 Grundlagen

3 Prozesse und Produkte vermessen

- Meß- und Überprüfungsverfahren auswählen und etablieren

4 Korrigierende Aktionen initiieren

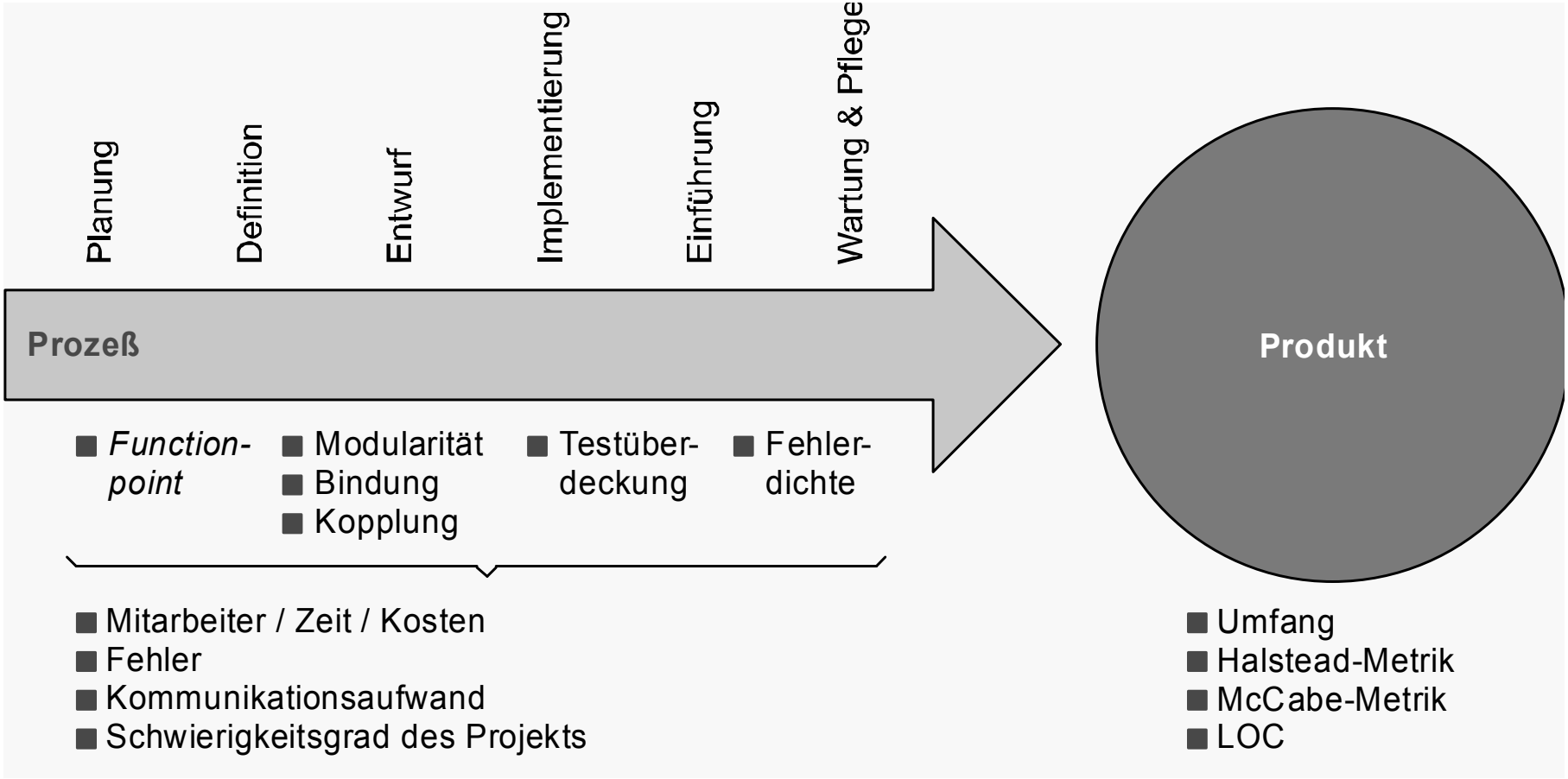
- Überstunden anordnen usw.
- Projektplan ändern

5 Loben und Tadeln

- Nichtmonetäre Belohnungen.

6.2 Metriken definieren, einführen und anwenden

- **Software-Metrik**
  - **Kenngröße eines Software-Produkts**
  - **Kenngröße eines Software-Prozesses**



6.2 Metriken definieren, einführen und anwenden

- **Meßtechnik**

  1 Definition der Meßziele

  2 Ableitung der Meßaufgaben aus den Meßzielen

  3 Bestimmung der Meßobjekte

  4 Festlegen der Meßgröße und Meßeinheit

  5 Zuordnung der Meßmethoden und Meßwerkzeuge zu den Meßobjekten und Meßgrößen

  6 Ermittlung der Meßwerte

  7 Interpretation der Meßwerte.

6.2 Metriken definieren, einführen und anwenden

- **Beispiel-Metrik Umfang**
  - **Meßziel**
    - **Bestimmung der Anzahl der nichtkommentierten Quellanweisungen**
  - **Meßaufgabe**
    - **Zählen der Anzahl der nichtkommentierten Quellanweisungen eines Programms**
  - **Meßobjekt**
    - **Auswahl eines zu vermessenden Programms.**

6.2 Metriken definieren, einführen und anwenden

- **Kenngröße**
  - **Anzahl Quellanweisungen einschl. Compiler-Anweisungen und Datendeklarationen, aber ohne Leerzeilen oder Kommentarzeilen**
- **Meßeinheit**
  - **KNCSS (1000 NCSS)**
- **Meßmethoden/Meßwerkzeuge**
  - **Automatischer Zeilenzähler**
- **Interpretation**
  - **Repräsentiert den Umfang der produzierten Software.**

Gütekriterien für Software-Metriken

1 Objektivität (Intersubjektivität)

- Objektiv, wenn keine subjektiven Einflüsse des Messenden auf die Messung möglich

2 Zuverlässigkeit (Meßgenauigkeit)

- Maß ist stabil und präzise (zuverlässig)

3 Validität (Gültigkeit, Meßtauglichkeit)

- Eindeutiger und unmittelbarer Rückschluß auf die Ausprägung der Kenngröße

4 Normierung

- Gibt es eine Vergleichbarkeitsskala, dann ist ein Maß normiert.

Gütekriterien für Software-Metriken

5 Vergleichbarkeit

- Kann ein Maß mit anderen Maßen in eine Relation gesetzt werden?

6 Ökonomie

- Möglichst geringe Kosten
- Abhängigkeiten
  - Automatisierungsgrad
  - Anzahl der Meßgrößen
  - Anzahl der Berechnungsschritte

7 Nützlichkeit

- Werden mit einer Messung praktische Bedürfnisse erfüllt?

Klassifikation von Metriken

- Objektiv/Subjektiv
  - Objektiv: leicht quantifizierbar und meßbar
  - Subjektiv: menschliche Einschätzung
- Absolut/Relativ
  - Absolut: invariant gegenüber Hinzufügen neuer Elemente
  - Relativ: Metriken ändern sich
- Explizit/Abgeleitet
  - Explizit: werden direkt ermittelt
  - Abgeleitet: werden von anderen expliziten oder abgeleiteten Metriken berechnet.

Klassifikation von Metriken

- **Dynamisch/Statisch**
  - **Dynamisch: Metriken besitzen Zeitdimension**
  - **Statisch: Metriken bleiben invariant**
- **Vorhersagend/Erklärend**
  - **Vorhersagend: Metriken können im voraus ermittelt oder generiert werden**
  - **Erklärend: Metriken werden hinterher ermittelt.**

Klassifikation von Metriken

- **Prozeßorientiert/Produktorientiert**
  - **Prozeßorientiert: Metrik ist ein Attribut des Entwicklungs- und Pflegeprozesses**
  - **Produktorientiert: Metrik wird am Produkt gemessen**
- **Global/Speziell**
  - **Global: für Software-Manager, auf einem hohen Abstraktionsniveau**
  - **Speziell: Indikatoren für jeweils eine spezielle Phase im Entwicklungsprozeß.**

6.2 Metriken definieren, einführen und anwenden

- **Metriken bei Hewlett-Packard**
  - **Mitarbeiter / Zeit / Kosten *(People / Time / Cost)***
  - **Umfang *(Size)***
  - **Fehler *(Defects)***
  - **Kommunikation *(Communications)***
  - **Schwierigkeit *(Difficulty)***
  - **Wartung *(Maintainance).***

Mitarbeiter/Zeit/Kosten

Projektname: Release-Nr. :

Aktivitäten	Lohnkostenaufwand in Ingenieurmonaten	Kalendermonate
Definition		
Entwurf		
Implementierung		
Test		
Summen		

% Überstunden (oder Minderarbeit) = _____ %

Gebrauchsanleitung

- Am Ende jeder Aktivität ist die entsprechende Zeile auszufüllen.
- Minderarbeit durch ein Minuszeichen kennzeichnen.
- Mit der Produktfreigabe ist das ausgefüllte Formular an die Metrikgruppe zu senden.

Definitionen

- Lohnkostenaufwand in Ingenieurmonaten
  Summe der Kalenderlohnkostenmonate, die jedem Projektingenieur zugeordnet wurden, einschl. der Mitarbeiter, die Tests durchführen. Längere Urlaube und Abwesenheiten werden nicht berücksichtigt. Zeiten, die Projektmanager für Managementaufgaben benötigt haben, werden nicht eingetragen.
- Überstunden/Minderarbeit
  Ingenieurzeit, die über/unter der 40 Stunden-Ingenieurwoche im Durchschnitt des Projekts lag. %-Über-/Unterzeit kann als Normalisierungsfaktor für den Lohnkostenmonat verwendet werden.
- Kalendermonate
  Die vergangene Zeit in Kalendermonaten zwischen speziellen Projekt-Kontrollpunkten.

Fehler vor der Auslieferung *(pre-release defects)*

Projektname: Release-Nr. :

Aktivitäten	Fehler eingebracht (optional)	Fehler gefunden	Fehlerbehebung abgeschlossen
Definition			
Entwurf			
Implementierung			
Test			
Summen			

Gebrauchsanleitung

- Am Ende jeder Aktivität sind die gefundenen Fehler und abgeschlossenen Fehlerbehebungen einzutragen. Die eingebrachten Fehler sind zu aktualisieren.
- Wurden Fehler während einer Aktivität nicht gezählt, dann ist nichts einzutragen; keine Null eintragen.
- Mit der Produktfreigabe ist das ausgefüllte Formular an die Metrikgruppe zu senden.

Definitionen

- Fehler:
  Ein Fehler ist eine Abweichung von der Produktspezifikation oder ein Fehler in der Spezifikation, wenn der Fehler nicht entdeckt und korrigiert wurde.
  Konnte der Fehler nicht entdeckt werden oder wurde er entdeckt aber nicht behoben, dann handelt es sich um eine Erweiterung und nicht um einen Fehler.
  Fehler schließen typographische oder grammatikalische Fehler in der Dokumentation *nicht* ein.
- Fehler eingebracht:
  Anzahl der Fehler, die dem Ergebnis einer Aktivität zugeordnet und nicht vor der letzten Aktivität gefunden werden konnten.
- Fehler gefunden:
  Anzahl der Fehler, die in einer Aktivität gefunden wurden.
- Fehlerbehebung abgeschlossen:
  Anzahl der Fehler, die in einer Aktivität korrigiert wurden.

6.2 Metriken definieren, einführen und anwenden

Ausgelieferter Umfang

Projektname:	Release-Nr. :
Sprache A:	Sprache B:

Zeilenzähler-Werkzeug (oder andere Technik):

	Sprache A:	Sprache B:
NCSS		
Kommentarzeilen		
Leerzeilen		
% wiederverwerteter Code		
# Prozeduren		

6.2 Metriken definieren, einführen und anwenden

♦ Erfassungsformular Umfang

- Inlinecodedokumentation

  Dokumentation, die nicht im Quellcode oder in der Endbenutzerdokumentation enthalten ist.

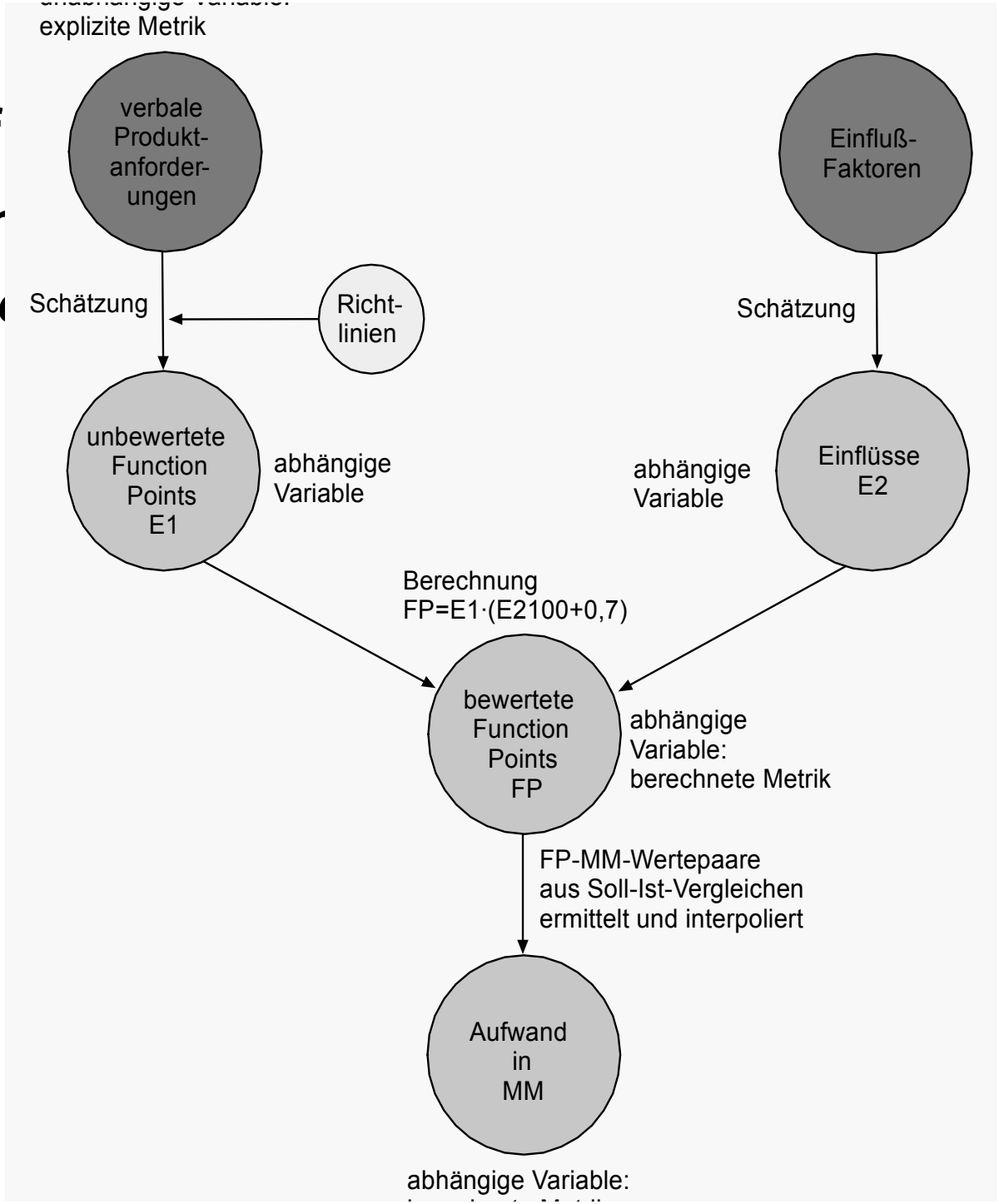
  Wenn die Zeilen geschätzt werden, dann ist von 54 Zeichen pro Zeile auszugehen.

- Wiederverwerteter Code

  Code, der in dieses Produkt eingegliedert wurde, und der vorher in einem anderen Produkt oder einem anderen Teil dieses Produktes einwandfrei arbeitete.

6.2 Metriken def

- **Meßbare vs. ir**
  - **Beispiel: *Fun***



- **Danke!**
- **Aufgaben**

- **Diese Präsentation bzw. Teile dieser Präsentation enthalten Inhalte und Grafiken des Lehrbuchs der Software-Technik (Band 2) von Helmut Balzert, Spektrum Akademischer Verlag, Heidelberg 1998**