

# Vorlesung "Software-Engineering"

---

Prof. Ralf Möller, TUHH, Arbeitsbereich STS

## ■ Vorige Vorlesung

- Analytische Qualitätssicherung
  - | Testverfahren
- Konstruktive Qualitätssicherung
  - | ISO-9000

## ■ Heute

- TQM
- Projektmanagement
- Versionierung und Konfiguration von
  - | Software
  - | Entwicklungsunterlagen

# Konstruktive Qualitätssicherung

---

- TQM (Total Quality Management)
- CMM (Capability Maturity Model)
- SPICE (Software Process Improvement and Capability dEtermination)
  
- Siehe: Balzert, Lehrbuch der Softwaretechnik Band 2

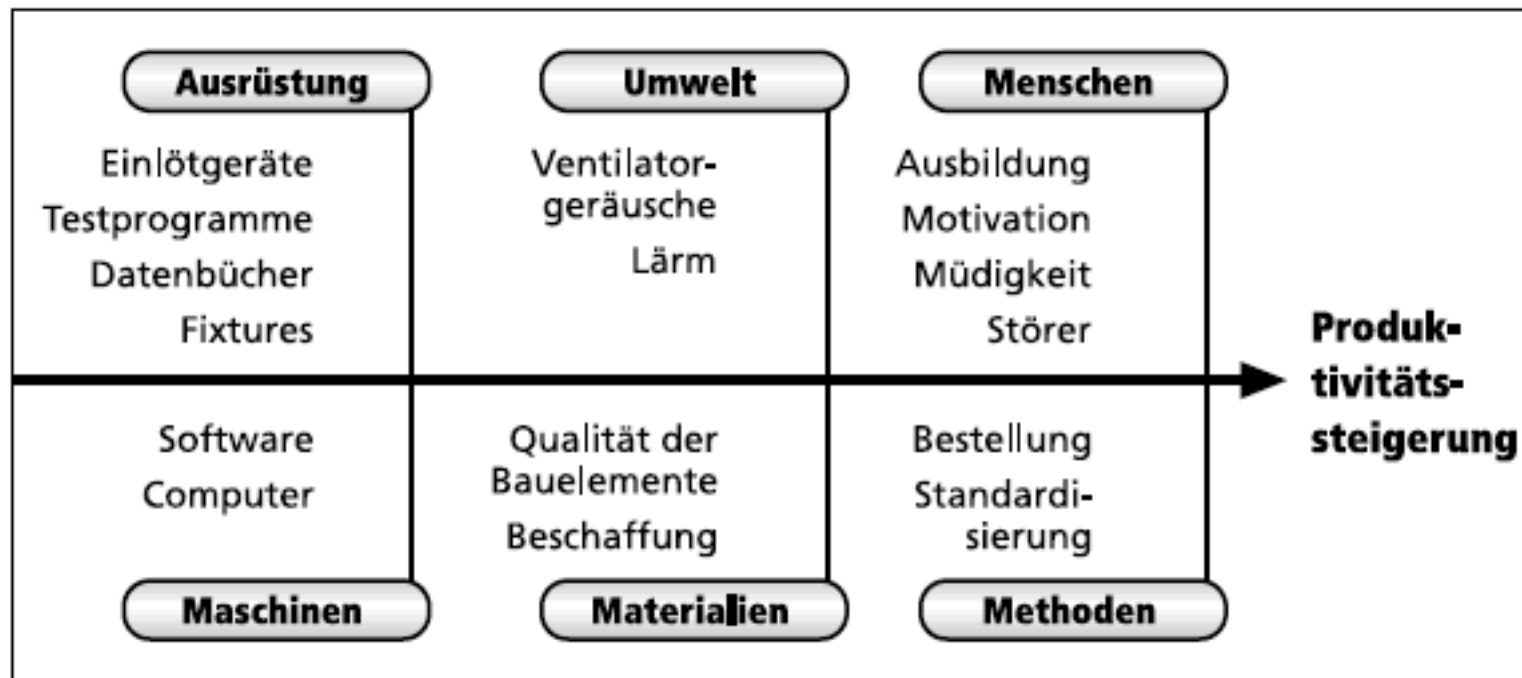
# Totales Qualitätsmanagement

---

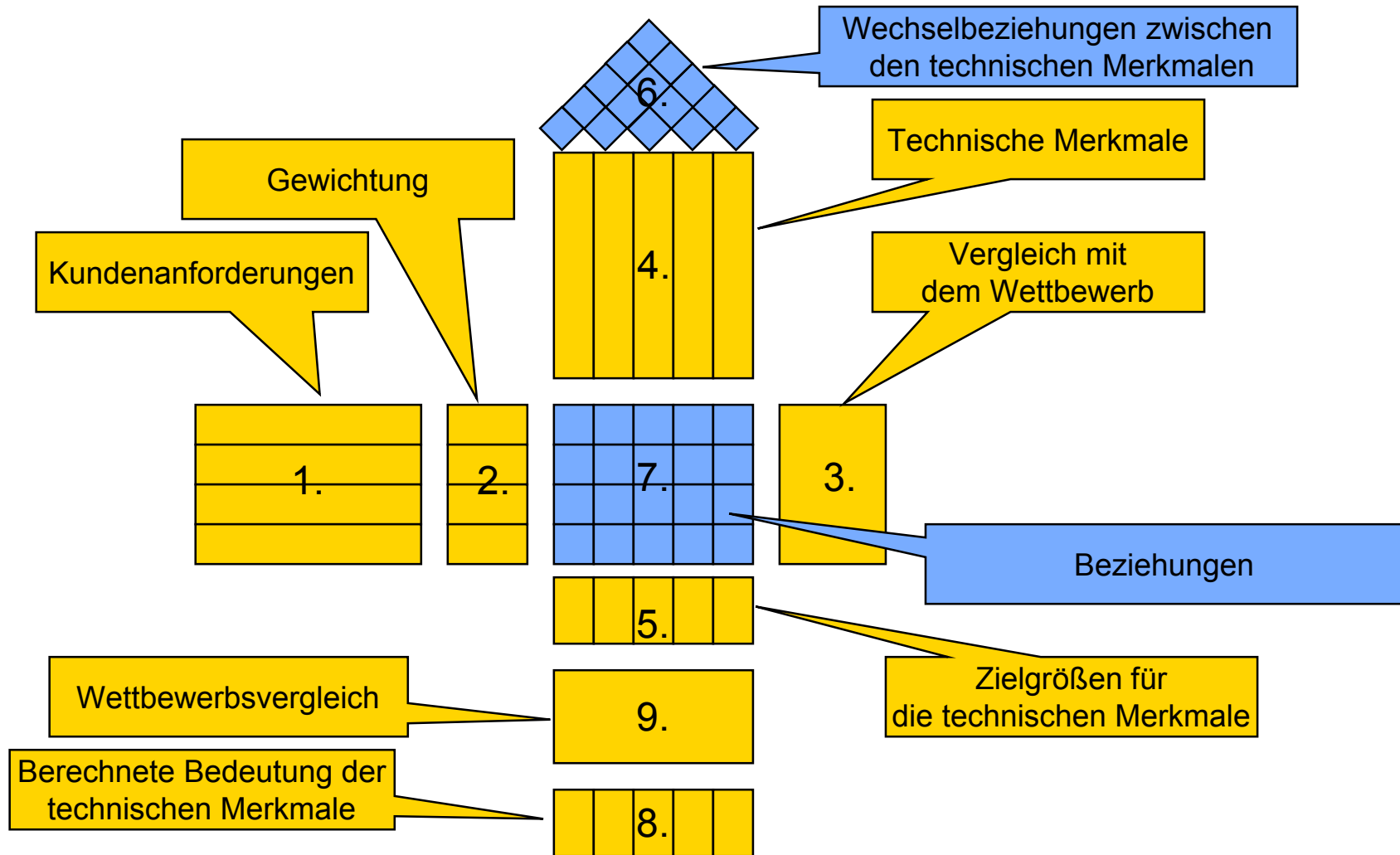
- **Totales Qualitätsmanagement (TQM)** ist eine auf der Mitwirkung aller ihrer Mitglieder basierende Führungsmethode einer Organisation, die Qualität in den Mittelpunkt stellt und durch Zufriedenheit der Kunden auf langfristigen Geschäftserfolg sowie auf Nutzen für die Mitglieder der Organisation und für die Gesellschaft zielt.
- TQM umfaßt gleichermaßen produktorientiertes und prozeßorientiertes Qualitätsmanagement.
- Konzepte des TQM sind Qualitätszirkel und das **Quality Function Deployment** (QFD) mit Hilfe der QFD-Matrix („Haus der Qualität“).
- **Qualitätszirkel** sind regelmäßige Treffen weniger Mitarbeiter mit dem Ziel, Qualitätsprobleme zu lösen bzw. aktiv Verbesserungen einzuführen.
  - Identifikation von Problemen
  - Pareto-Prinzip: 20:80
  - Analyse von Ursache-Wirkungs-Zusammenhängen z.B. mit Grätendiagrammen im *Brainstorming*
  - Zieldefinition, Lösungssuche, Maßnahmendefinition
  - Ergebniskontrolle

# Grätendiagramm / Ishikawa-Diagramm

## ■ Ursache-Wirkungs-Diagramm



# QFD-Matrix (Haus der Qualität)

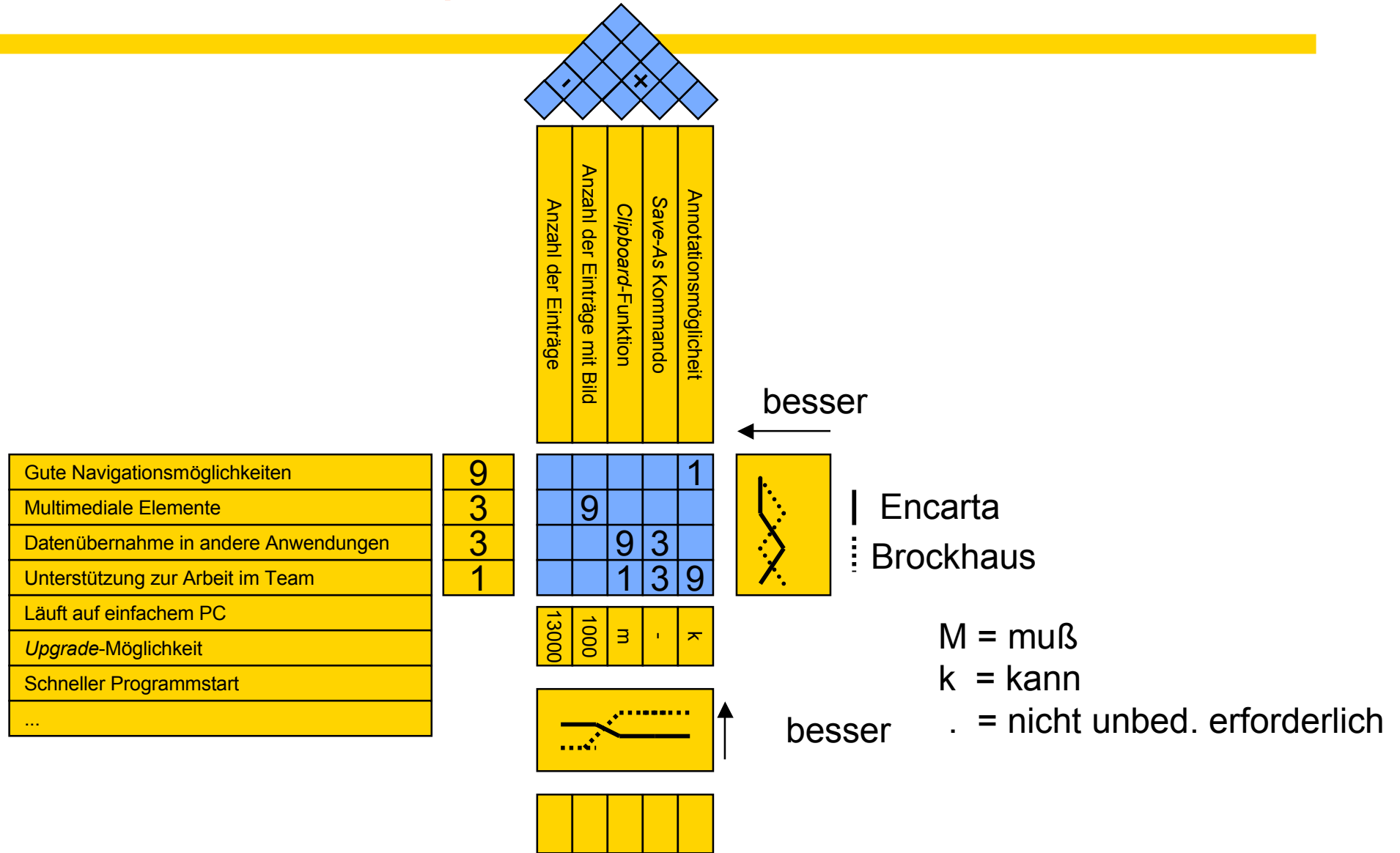


# Methodisches Vorgehen

---

1. Sammle Kundenanforderungen und liste sie auf.
2. Priorisiere die Kundenanforderungen (durch paarweisen Vergleich) und definiere Gewichte.
3. Nimm einen Wettbewerbsvergleich vor, um Ziele für die Positionierung am Markt vorzugeben.
4. Ermittle technische Merkmale zur Realisierung der Kundenanforderungen.
5. Spezifiziere Zielgröße für die technischen Merkmale.
6. Ermittle Abhängigkeiten zwischen technischen Merkmalen (+ / -)
7. Trage Beziehungen in der Matrix ein:  
Welche Kundenanforderung wird durch welche technischen Merkmale realisiert?  
Überprüfe, ob es "irrelevante" technische Merkmale gibt.
8. Berechne lokale Priorität eines Merkmals
9. Erstelle einen Wettbewerbsvergleich in den technischen Merkmalen  
(→ Handlungsspielraum)

# QFD-Matrix: Beispiel Online-Lexikon



# Projektmanagement

---

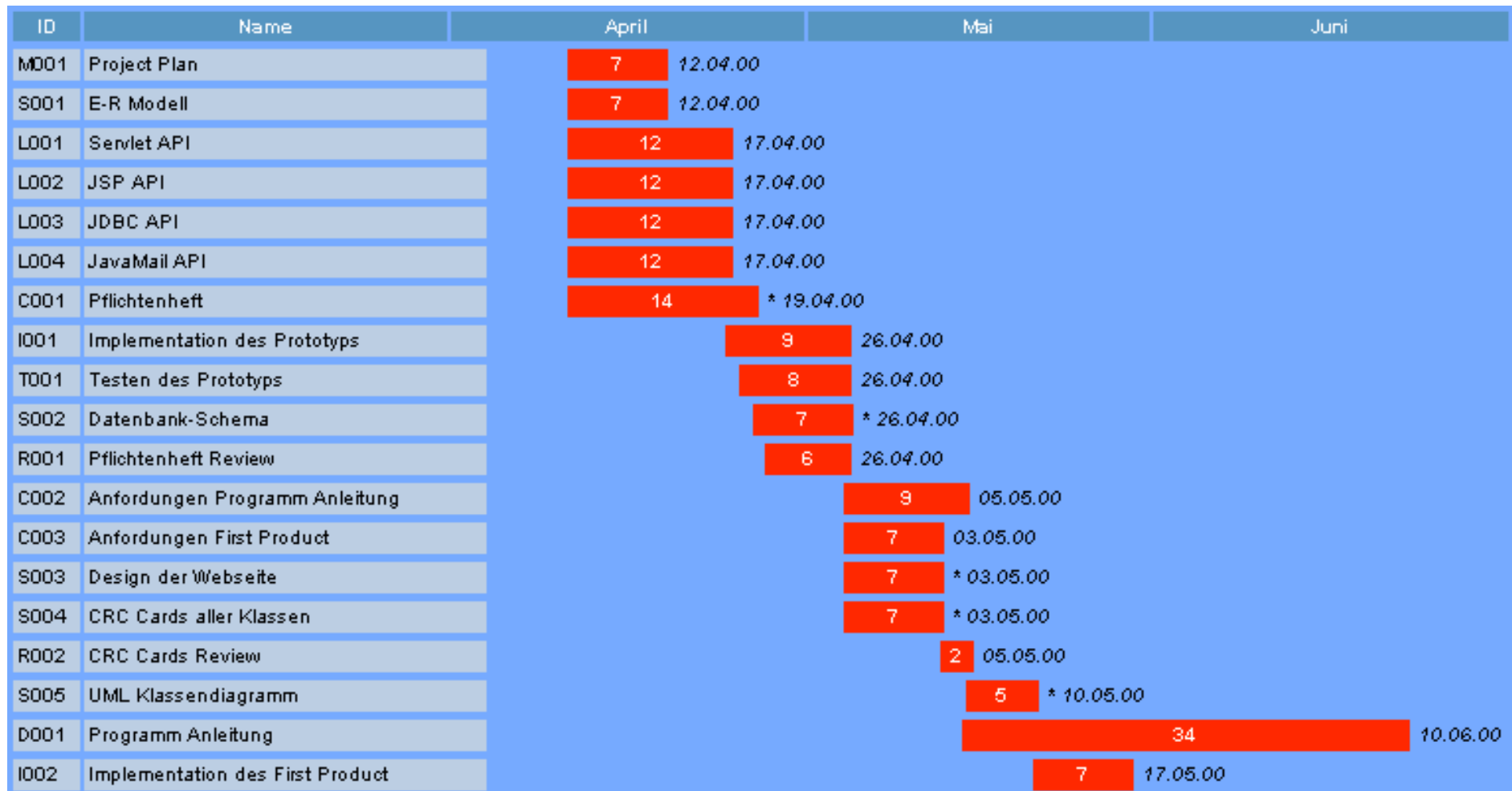
- **Wirtschaftliche Kontrolle**
  - Termine (z.B. Gantt-Diagramme, Netzplantechniken)
  - Kosten (z.B. Function Point-Methode, Prozentsatzmethode, ...)
  - Personaleinsatz
- **Fachliche/technische Kontrolle**
- **Führen des Projektteams, Organisation**

# Gantt-Diagramme

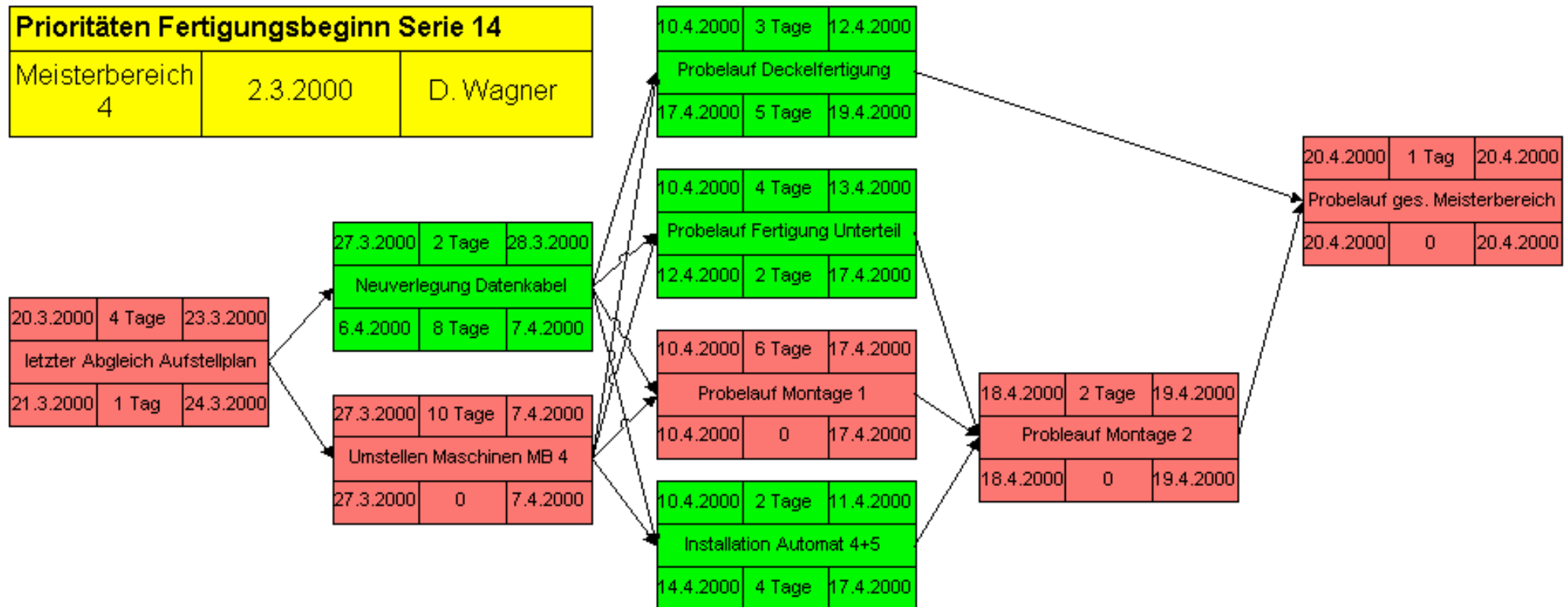
---

- In einem Gantt-Diagramm wird jeder Vorgang in einer Zeile dargestellt. Die Zeitachse verläuft x-Richtung. Start, Ende und Dauer jedes Vorgangs werden durch einen Balken (je Vorgang) dargestellt. Der Bearbeitungsstand des Vorgangs kann dargestellt werden (zweifarbige Balken).
- Vorteile: einfach, eingängig, schnelle Orientierung durch graphische Visualisierung
- Nachteile: Abhängigkeiten können nur bedingt dargestellt werden, Balkendiagramme für Projekte werden unhandlich gross (DIN A3 - DIN A0).

# Beispiel: Gantt-Diagramme



# Netzplantechnik: PERT-Diagramme



Die rot eingezeichneten Blöcke stellen den kritischen Pfad dar.  
 Verzögerungen in diesen Tätigkeiten führen automatisch zu  
 Verzögerungen des Gesamtprojekts.  
 Die grün dargestellten Blöcke verfügen über Pufferzeiten.

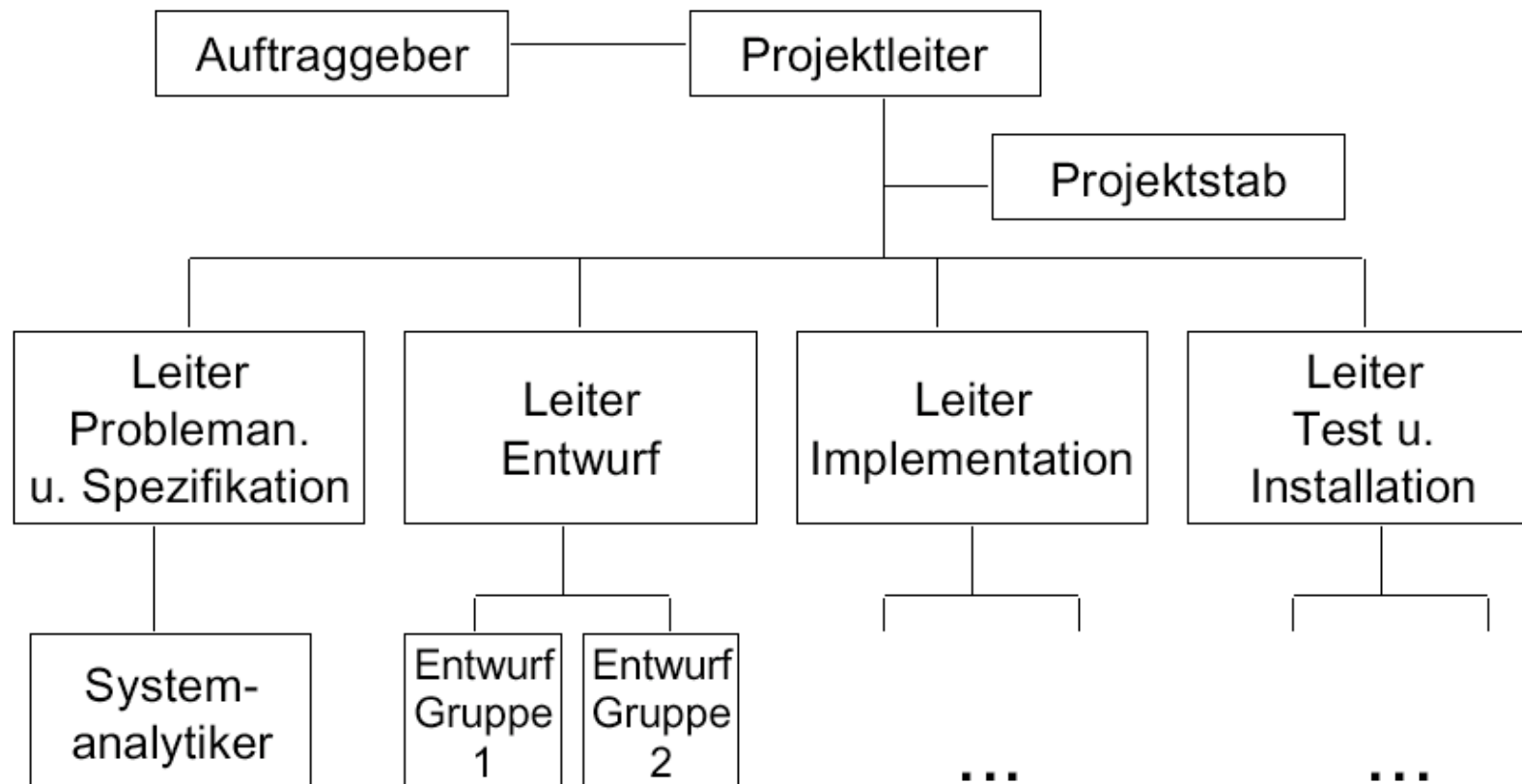
# Projektorganisation

---

- Zusammenstellung von Teams
- Bereitstellung von benötigten Ressourcen
- Aufbau der Organisationsstruktur
  - Zuordnung von Teilaufgaben/Phasen zu Teams oder einzelnen Personen
  - Genaue Festlegung der Aufgaben, Rechte und Pflichten aller am Projekt beteiligten Personen (Zuständigkeiten)
- Organisationsstrukturen
  - Klassische hierarchische Struktur
  - Chef-Programmierer-Struktur

# Hierarchische Struktur

- Hierarchie geprägt durch Phasen und Teilaufgaben



# Hierarchische Struktur: Schwächen

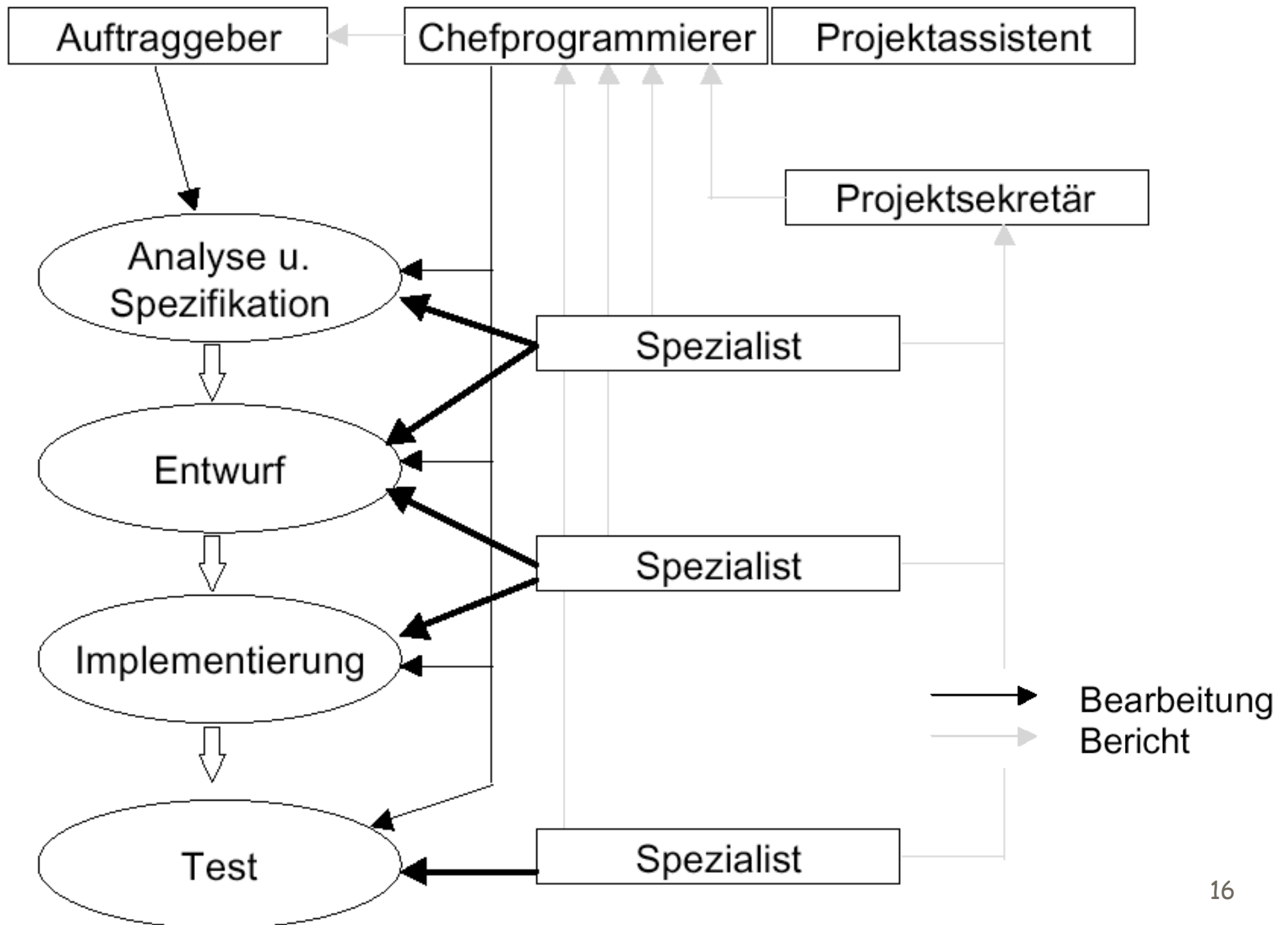
---

- Projektleiter zu weit von Programmierung entfernt
- Mehrstufigkeit behindert Kommunikation
- Aufstieg in Hierarchie bis zur Inkompetenz

# Chef-Programmierer-Team

---

- Verzicht auf Projektleiter, der nicht an Systementwicklung beteiligt ist
- Einsatz von sehr guten Spezialisten, die mit hoher Eigenverantwortung arbeiten
- Beschränkung der Teamgröße
- Zusammensetzung der Teams:
  - Chef-Programmierer
  - Projektassistent
  - Projektsekretär
  - Spezialisten (Systemanalytiker, Programmierer, Testspezialisten)



# Chef-Programmierer-Team: Bewertung

---

## ■ Vorzüge

- Chefprogrammierer kann durch direkte Einbindung Kontrollfunktion besser wahrnehmen
- Geringere Kommunikationsschwierigkeiten
- Kleinere (Spezialisten-)Teams sind produktiver

## ■ Nachteile

- Beschränkung auf kleine Teams
- Anforderungen an Chefprogrammierer nahezu unerfüllbar
- Stellung des Projektsekretärs problematisch

# Koordination der Arbeit im Team

---

- **Verschiedenste Dokumente**

- Versionskontrolle
- Konfiguration

-> Dokumentenmanagementsysteme für die Softwareentwicklung

- **Lokale Umgebung**

- Z.B. CVS

- **Globale, heterogene Umgebung**

- Z.B. WebDAV

# Danksagung



---

**Nachfolgende Präsentationen wurden entnommen aus:**

**Dokumentenmanagementsysteme  
- *CVS und WebDAV im Vergleich* -**

# Was ist CVS?

---

## ■ Versionskontrollsystem

- Entwicklung seit 1986 [aus RCS]
- Inkrementelle Entwicklung von Software
- Aufzeichnung und Nachverfolgung

## ■ CVS – Funktionalitäten

- Repository
- Revisionen
- Merging
- Branching
- Diffing
- Tagging

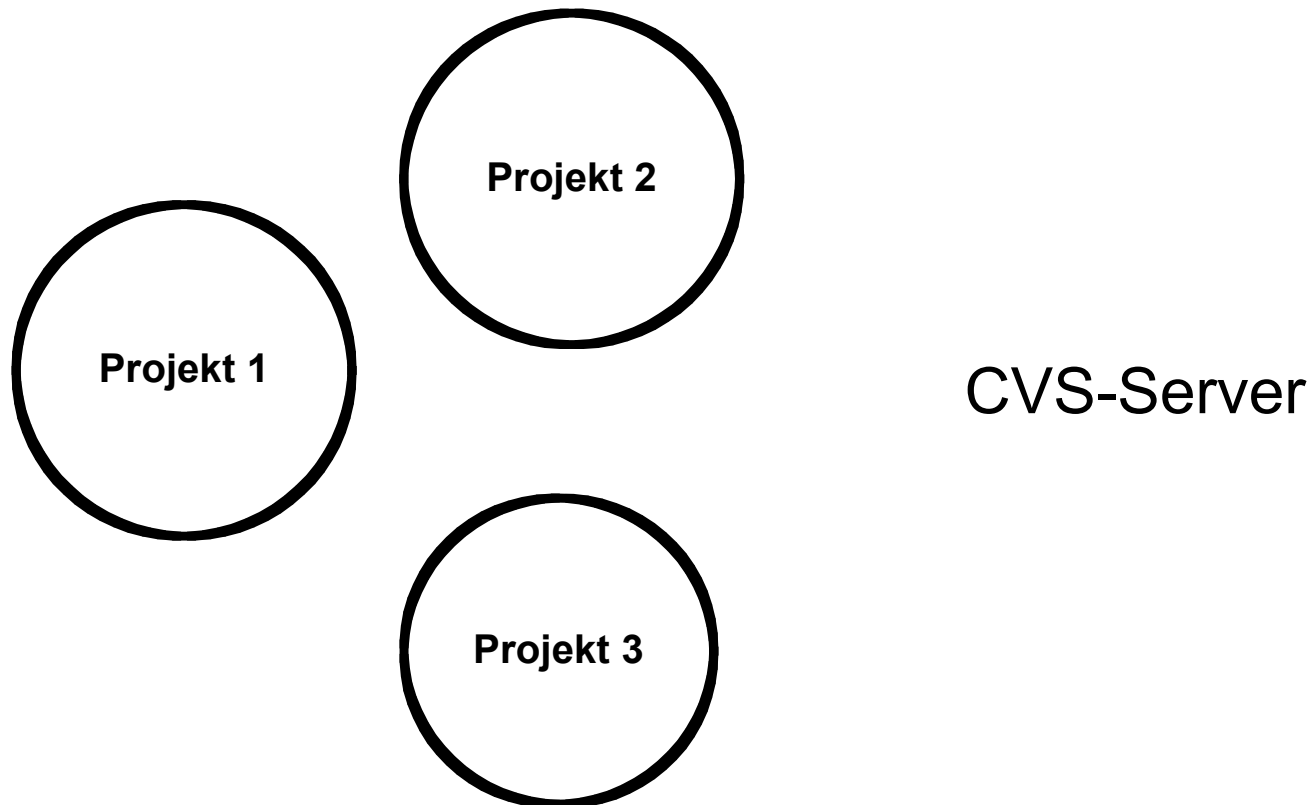
# Repository

---

- Engl.: Archiv
- Zentralisierte Verzeichnisstruktur
  - Zentraler Ablageort für alle Dateien
  - Angaben von früheren Versionen  
[zur evtl. Wiederherstellung]

# Repository (2)

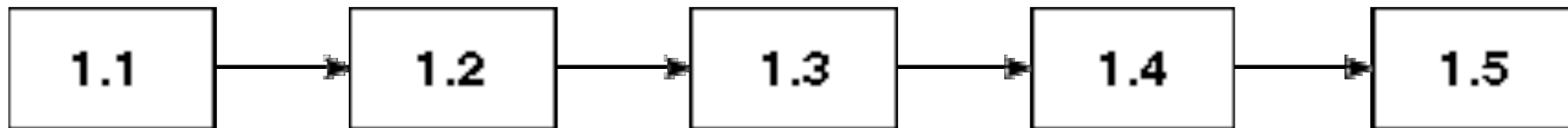
---



# Revisionen

---

- Jede Datei erhält eine Revisionsnummer [durch den CVS-Server zugewiesen]
- Versionsnummer [gerade Anzahl an Ziffern]

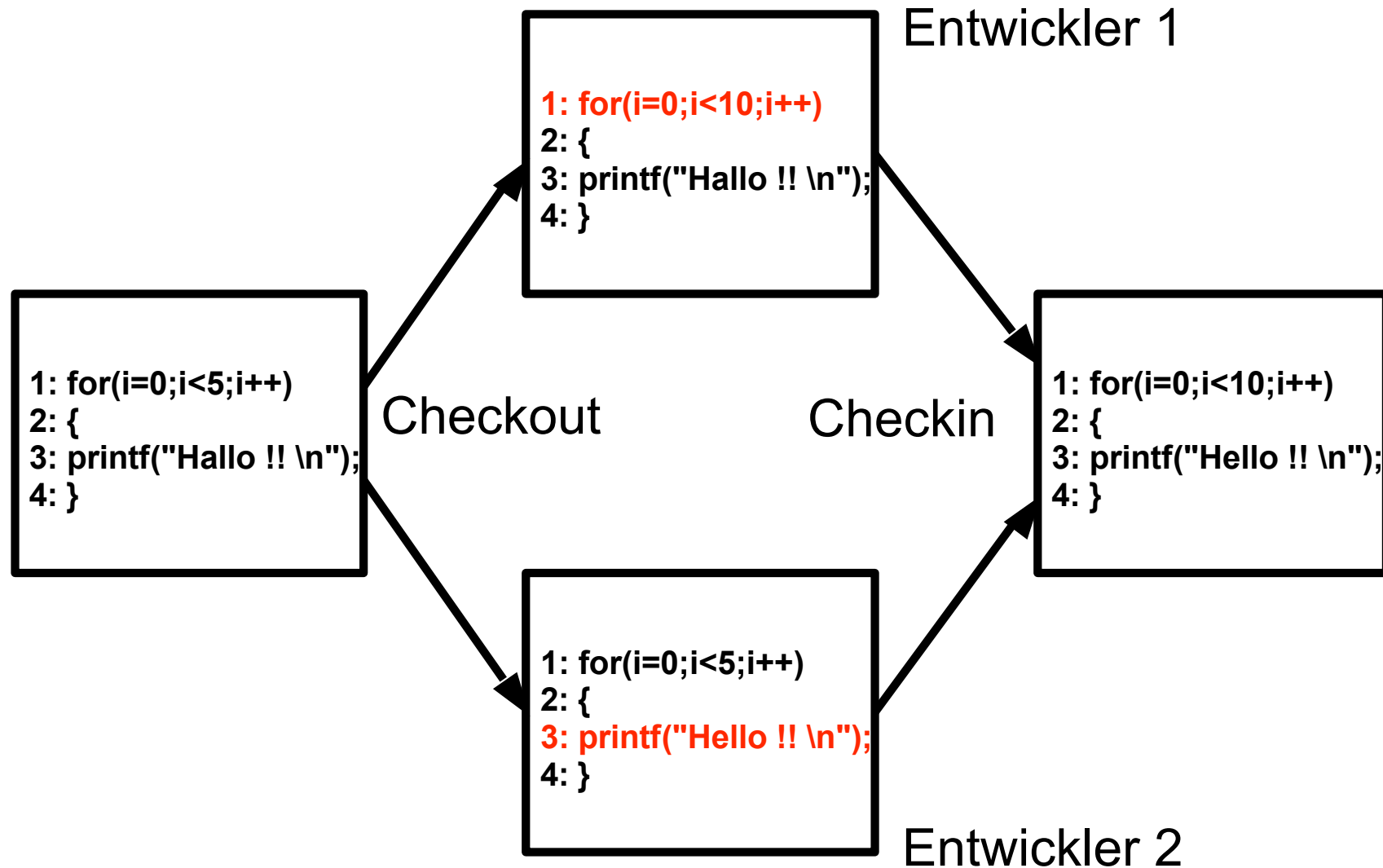


# Merging

---

- Vereinen zweier verschiedener Versionen einer Datei
- Ermöglicht gleichzeitiges Arbeiten an einer Datei [ohne Aussperren]
- Bei Konflikt manuelle Korrektur nötig [siehe Beispiel]

# Merging



# Branching

---

- gleichzeitiges Arbeiten an einer Datei durch mehrere Entwickler
- Gewährleistung der Lauffähigkeit von Versionen

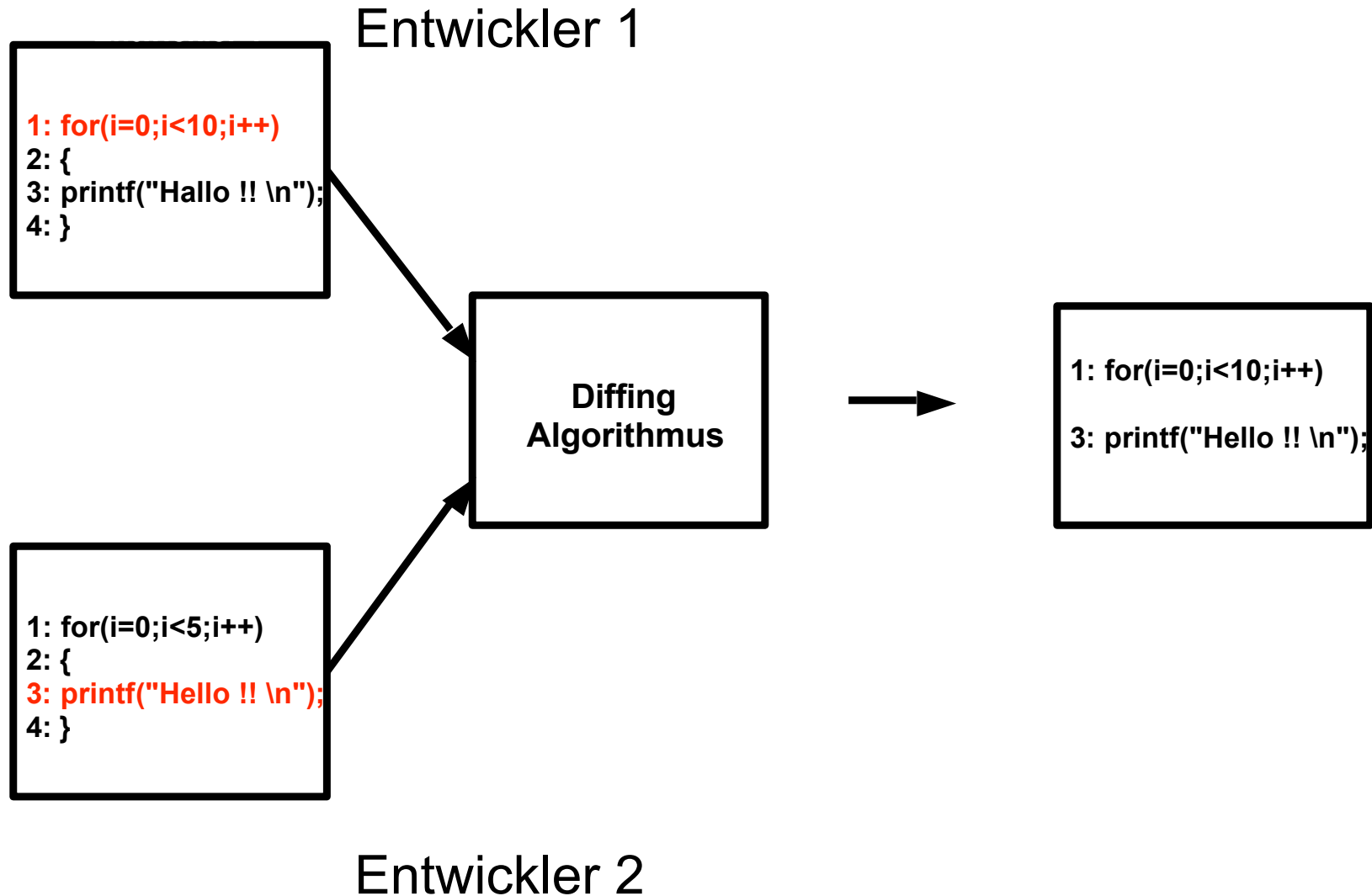
# Diffing

---

- Erkennen und Mitteilen von Unterschieden von verschiedenen Dateirevisionen
- Algorithmus zur Unterschiedserkennung funktioniert zeilenweise

# Diffing (2)

---



# Tagging

---

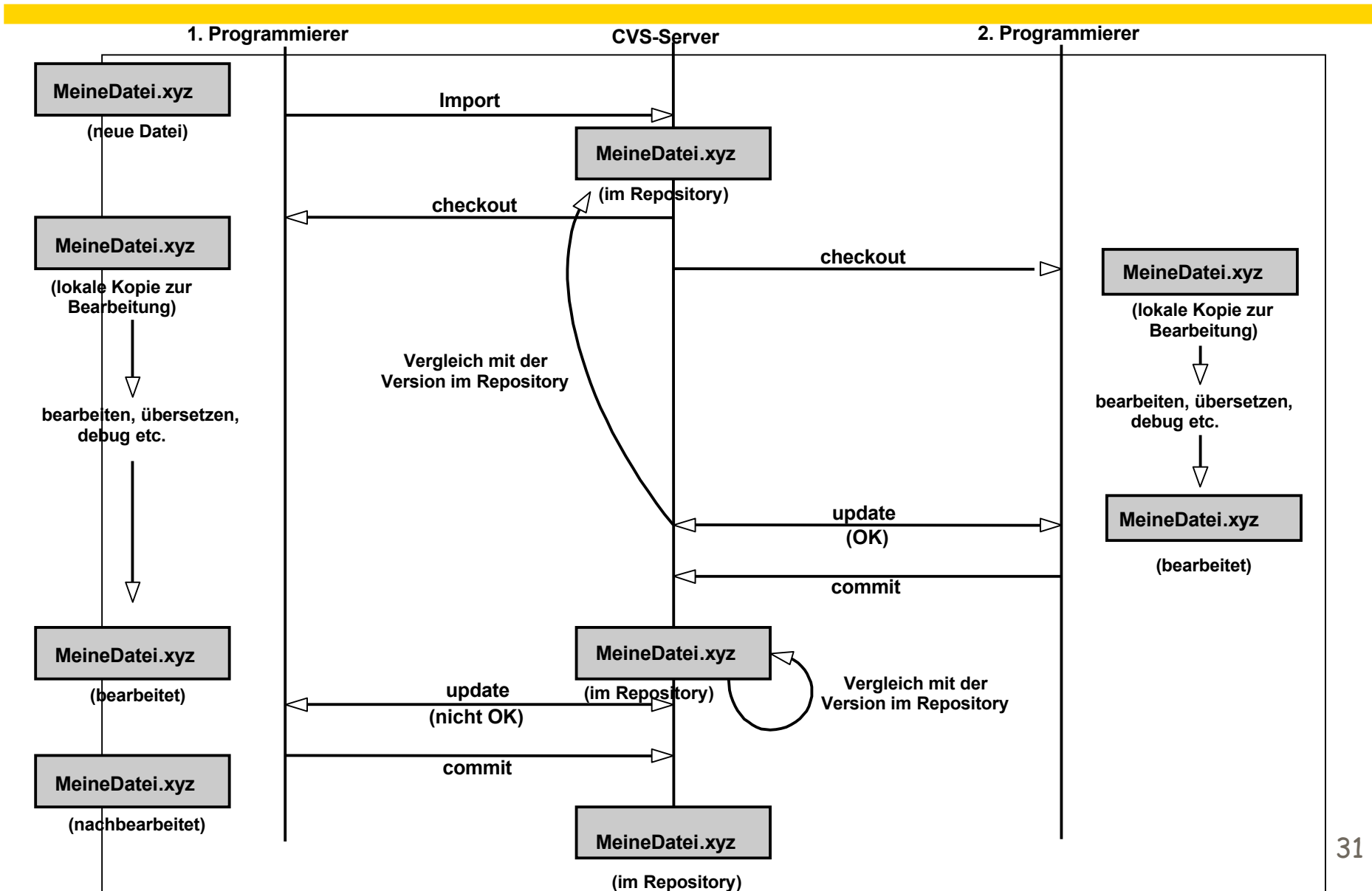
- Symbolische Verlinkung von Dateien eines Projekts
- Somit Erstellung eines Releases
- Nötig wegen nicht linearer Entwicklung

# Tagging (2)

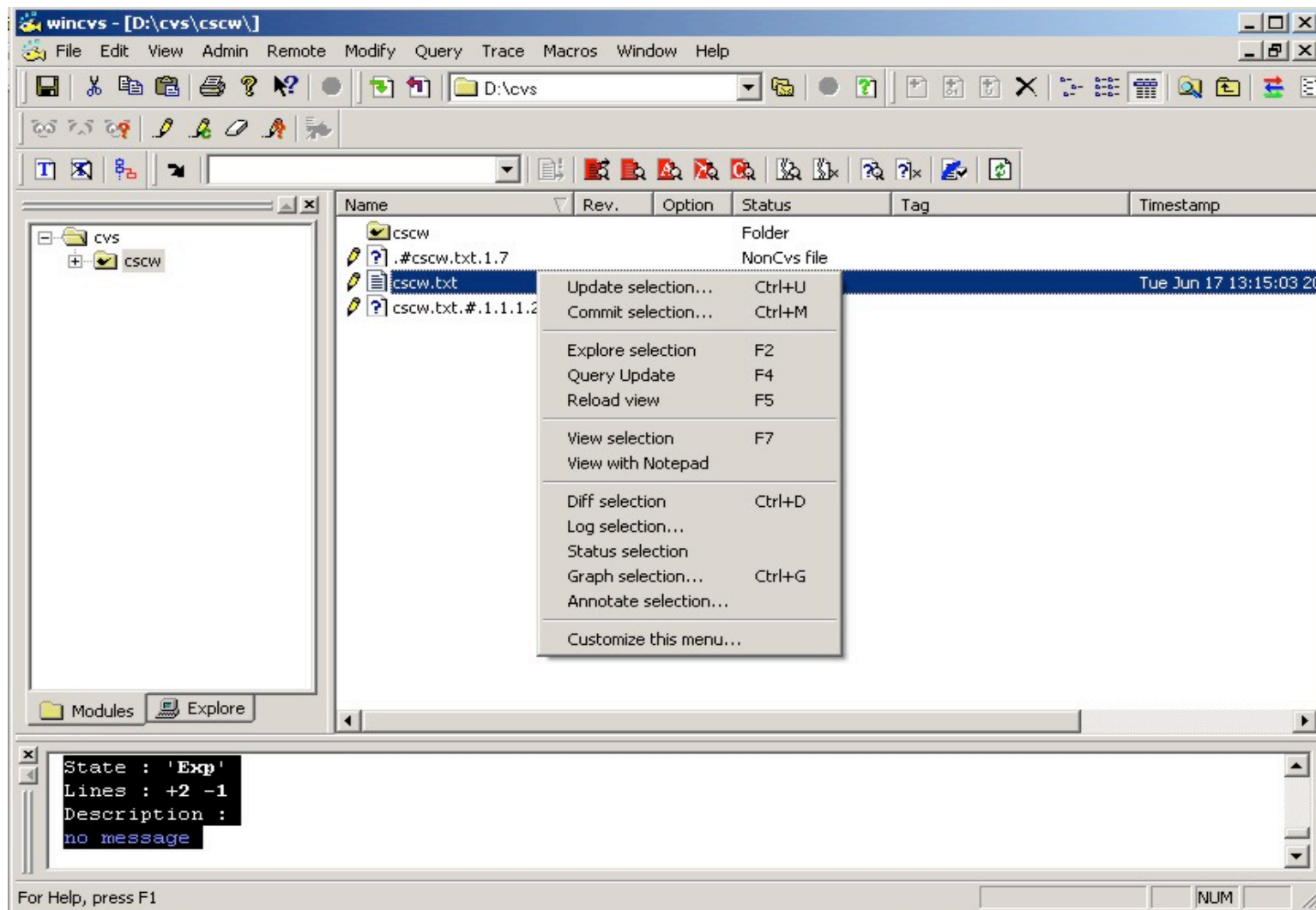
---



# Kooperatives Arbeiten in CVS



# Applikationen - WinCVS



# CVS - Zusammenfassung

---

## ■ Vorteile

- Gute Nachverfolgung der Arbeiten an den Quelldateien [bedingt durch die CVS-Funktionalitäten]

## ■ Nachteile

- Keine Unterstützung von Dateiattributen
- Keine Versionskontrolle von Binärdateien [aufgrund der Diff-Algorithmen]

# WebDAV - Definition

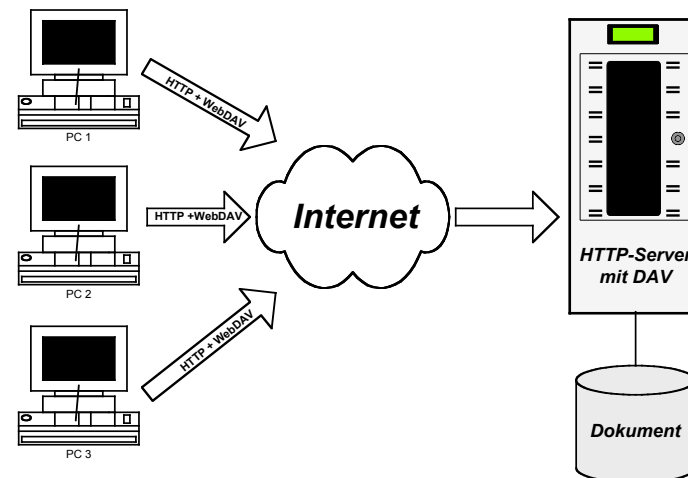
---

- WebDAV = **Web Distributed Authoring and Versioning**
  - ☒ (überwiegend) offenes Netzwerkprotokoll zum verteilten Dokumentenhandling auf Web-Servern
  - Ergänzung zu HTTP 1.1
  - Entwicklung seit 1996 durch die WebDAV Working Group (Teil der IETF)
  - Standardisierung 1999 als RFC 2518

# Zielsetzung

*„WebDAV will enable the original vision for the Web as a writeable, collaborative Medium.“*

- ⌘ Aufweitung des Internets vom „lesbaren“ Medium zu einem System zum Austauschen und Teilen von Daten
- ⌘ Ermöglichung eines direkten Zugriffs auf Dokumente im Web
- ⌘ Schaffung einer Steuerungsmöglichkeit zur kollektiven Veränderung von Dateien auf einem Server



# Grundfunktionen von HTTP

---

- **GET, PUT, DELETE**      Anfordern, Schreiben oder Löschen einer Ressource
- **POST**                      Übermittlung von Informationen an den Server
- **HEAD**                      Ähnlich wie GET, nur wird kein Message-Body zurückgegeben
- **OPTIONS**                      Anfordern von Informationen über eine Kommunikation
- **TRACE**                      Überprüfung der Request-/Responsekette
- **CONNECT**                      Reserviert für das Tunneling bei Proxies

## Grundfunktionalität von WebDAV

---

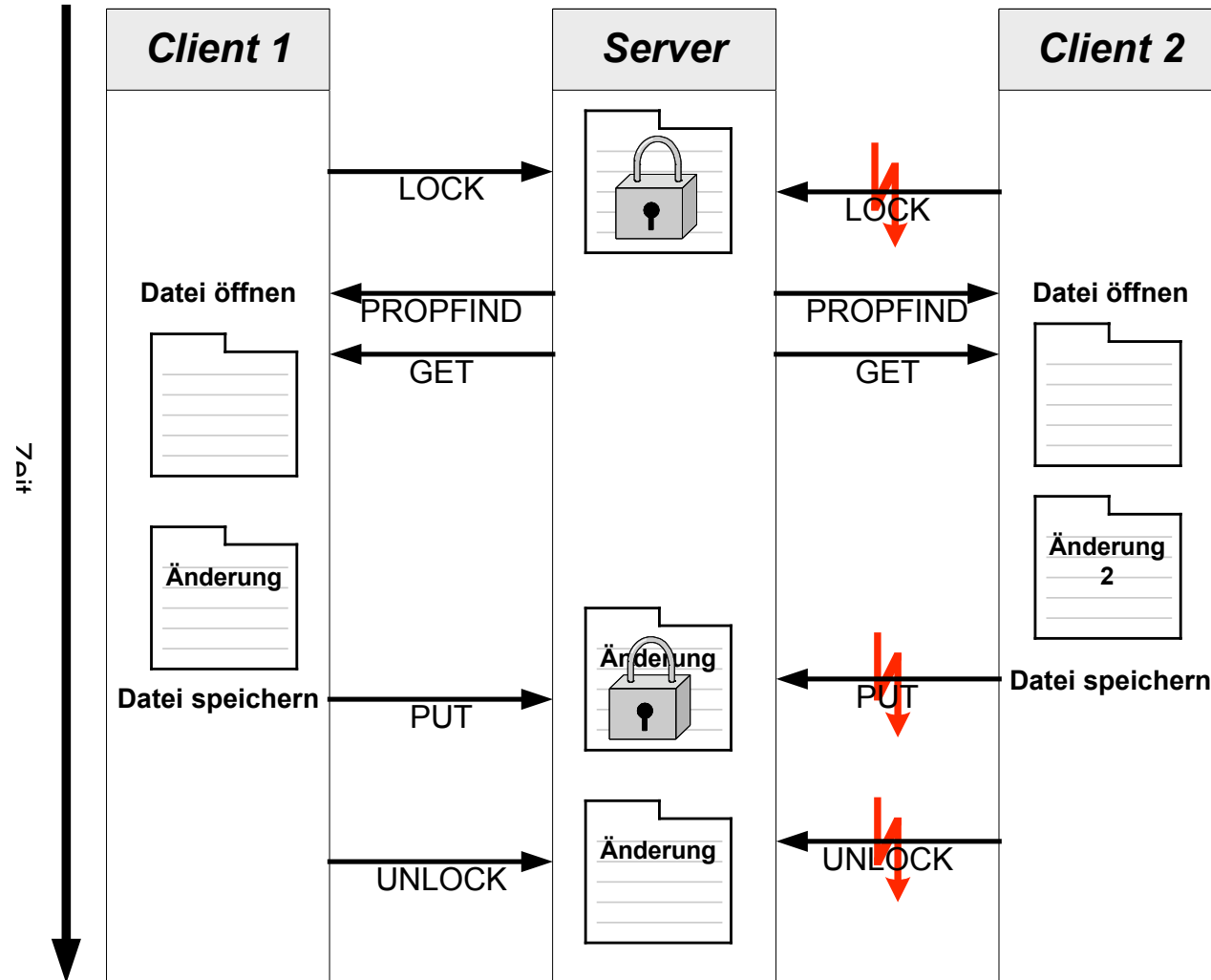
- Einführung von 7 neuen Methoden in WebDAV zur Realisierung von 3 Konzepten:
  - **Locking**  
Überschreibschutz von Dokumenten
  - **Properties**  
Zuweisung von Metadaten
  - **Namespace Management**  
Namensraumverwaltung

# Locking – Überschreibschutz (1)

---

- Sicherung der Datenkonsistenz durch einen Überschreibschutz (Lock) für Dokumente, die sich in Bearbeitung befinden
- Neue Methoden:
  - **LOCK**        Setzen eines Überschreibschutzes auf eine Ressource
  - **UNLOCK**     Freigeben einer geschützten Ressource inklusive aller durch dasselbe Lock gesperrter Dateien
- Schutzmöglichkeiten: exclusive / shared lock

# Locking – Überschreiberschutz (2)



# Properties (1)

---

- Möglichkeit des Zuweisens von **Metadaten** auf Dokumente:

Form: (**name**, **value**)

- Vordefinierte Eigenschaften:

· <b>creationdate</b>	Erstellungsdatum der Ressource
· <b>displayname</b>	Angezeigter Name der Ressource für den
	User
· <b>getcontentlanguage</b>	Sprache des Dokumentinhalts
· <b>getcontentlength</b>	Ressourcenlänge
· <b>getlastmodified</b>	Datum der letzten Änderung
· <b>lockdiscovery</b>	Aktiver Schreibschutz der Ressource
· <b>resourcetype</b>	Typ der Ressource
· <b>source</b>	Ressourcenquelladresse
· <b>supportedlock</b>	Unterstützte Lockfähigkeiten

## Properties (2)

---

- Einführung von 2 neuen Methoden:
  - **PROPFIND**  
liest Metadaten von Ressourcen oder Collections
  - **PROPPATCH**  
schreibt Metadaten von Ressourcen oder Collections
  
- Unterscheidung zwischen:
  - „live“-Properties
  - „dead“-Properties

# Namespace Management (1)

---

- Erstellung einer hierarchischen Ressourcenansammlung in Form von Namensräumen
  - Organisation erfolgt nur „virtuell“
  - Namensräume = „**Collections**“
- Drei weitere Methoden:
  - **MKCOL** Erzeugen einer neuen Collection an dem durch die URI (*Uniform Resource Identifier*) gegebenen Ort unter Berücksichtigung der bestehenden Hierarchie
  - **COPY** Kopieren von Collections oder Ressourcen
  - **MOVE** Verschieben von Collections oder Ressourcen

# Namespace Management (2)

---

## ■ Beispiel zum Erstellen einer Collection

### >>ANFRAGE

MKCOL http://www.sharemation.com:80/tilo/Bilder HTTP/1.1

Host: www.sharemation.com

Proxy-Connection: Keep-Alive

Connection: TE

TE: trailers, deflate, gzip, compress

User-Agent: UCI DAV Explorer/0.81 RPT-HTTPClient/0.3-3E

### >>SERVERANTWORT

HTTP/1.1 201 Created

# DeltaV – Versionskontrolle unter WebDAV (1)

---

- Standardisierung erst 2002 durch die **RFC 3253**
- Einführen von 11 neuen Methoden, davon v.a. zu erwähnen:
  - **VERSION-CONTROL**      Start der Versionskontrolle einer bislang nicht versionierten Datei
  - **CHECKOUT**              Start des Editierprozesses
  - **UNCHECKOUT**          Abbrechen des begonnenen Editierprozesses
  - **CHECKIN**              Beenden des Editierens, speichern
  - **REPORT**              Rückgabe der Versionshistorie

# Weitere (geplante) Funktionalitäten

---

- · **Zugriffskontrolle** (access control)
  - Setzen bzw. Löschen von Zugangskontrolllisten zur Vergabe der Nutzungsrechte für Ressourcen
  - Bearbeitung durch Subarbeitsgruppe „ACL“
  
- · **Ressourcen-Suche**
  - DAV Searching and Locating zur Durchführung von Suchoperationen auf dem Server
  - Dazu: Einführung der **SEARCH**-Methode

# Applikationen (1)

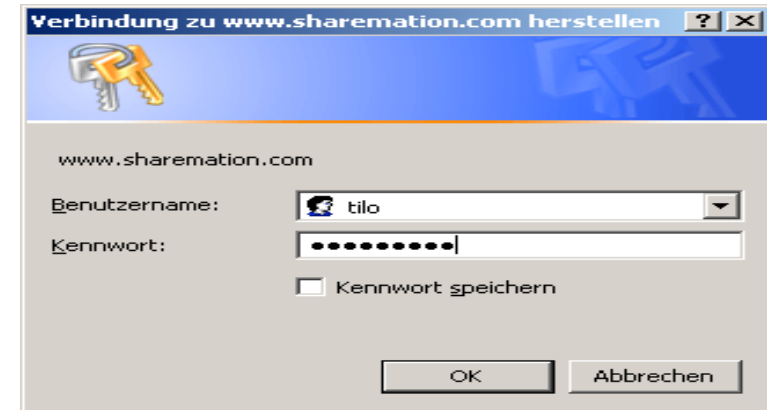
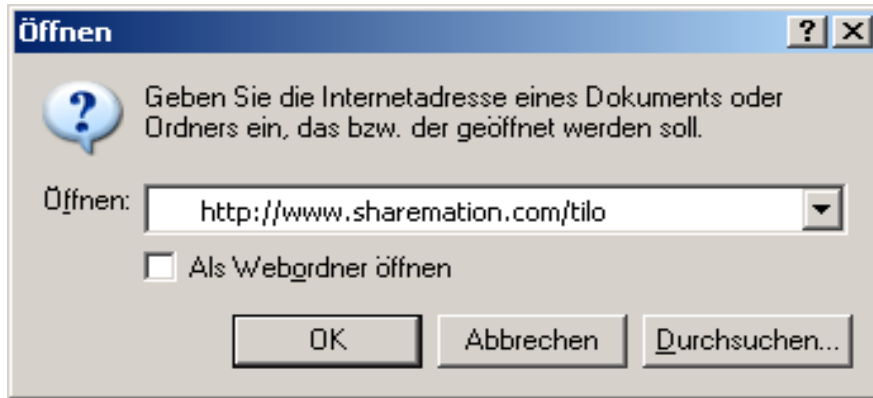
Office-Produkte	<ul style="list-style-type: none"><li>· Microsoft Word / Excel / Powerpoint / Publisher ab Version 2000</li><li>· Adobe Acrobat ab Version 5</li><li>· Adobe Photoshop ab Version 6</li></ul>
Web-Design-Tools	<ul style="list-style-type: none"><li>· Adobe Go Live ab Version 5</li><li>· Macromedia Dreamweaver ab Ver. 4</li></ul>
Internet-Browser	<ul style="list-style-type: none"><li>· Microsoft Internet-Explorer ab Version 5</li></ul>
Server-Betriebssysteme	<ul style="list-style-type: none"><li>· Novell Netware ab Version 5.1</li><li>· Microsoft IIS ab Version 5</li></ul>

## Open Source Varianten:

- mod\_dav-Modul für Apache
- KDE
- OpenOffice
- DAV-Explorer

# Applikationen (2)

## Internet-Explorer 6:



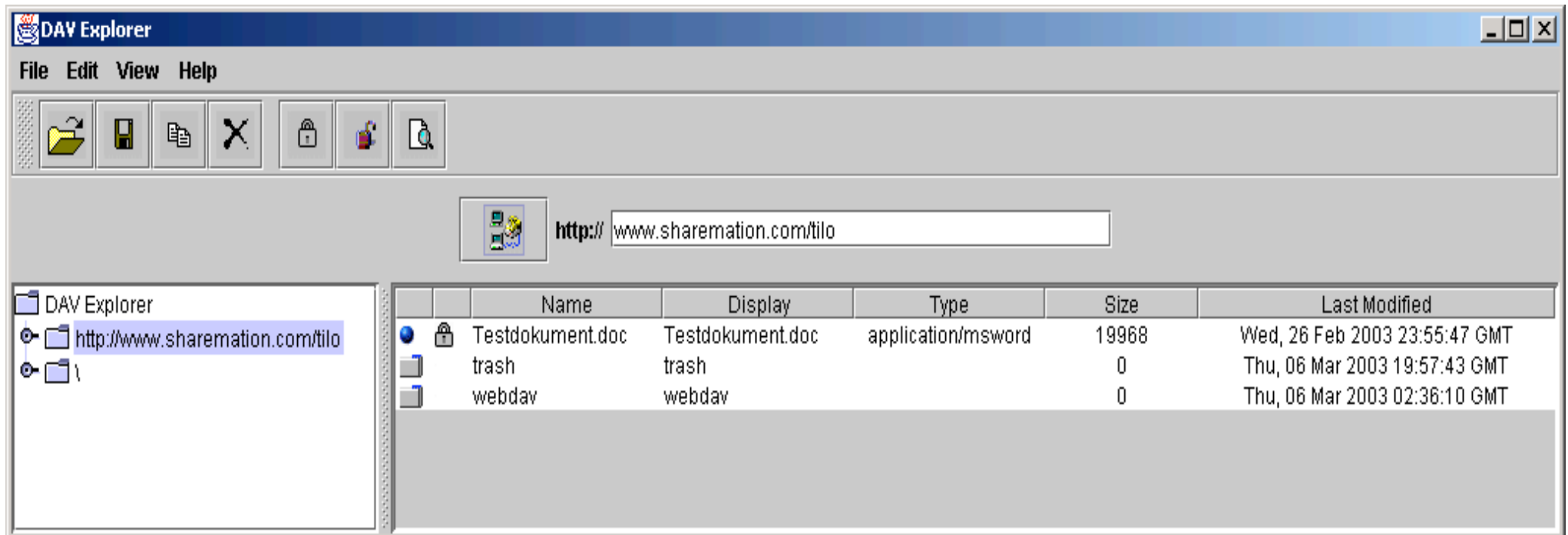
## Index of /tilo/

[Launch Web Folder](#)  
[Launch WFS WebUI](#)

Name	Modified	Size
<a href="#">trash/</a>	06.03.03 20:57:43 CET	
<a href="#">webdav/</a>	06.03.03 03:36:10 CET	
<a href="#">Testdokument.doc</a>	27.02.03 00:55:48 CET	19968

# Applikationen (3)

## DAVExplorer:



# Vergleich von CVS mit WebDAV

---

- Vorteile von CVS:
  - Open-source
  - Mittlerweile sehr stabil
  - Weite Verbreitung
  - Arbeitsunabhängigkeit für Teammitglieder
  - Versionshistorie und Durchsuchen aller alter Revisionen möglich
- Nachteile:
  - Keine Dateiattribute
  - Keine Versionskontrolle von Binärdateien
  - Kein Schreibschutz durch einen User realisierbar
  - Teilweise höherer Aufwand bei Basisdateioperationen nötig (z.B. Umbenennen)
  - CVS-Server ist nur für Linux/Unix verfügbar

# Vergleich von CVS mit WebDAV

---

## Vorteile von WebDAV:

- Direkter Aufsatz auf HTTP, dadurch Nutzung von Proxys als auch ssl-Verschlüsselung möglich
- Integration in viele schon vorhandene Anwendungen
- Metadaten über XML, Locking, Namespacemanagement
- Nicht nur Beschränkung auf reines Versionsmanagement

## Nachteile:

- Noch wenige Client-Implementierungen mit Versionisierungs-Unterstützung
- Nur teilweise open-source
- Mögliche „Kinderkrankheiten“
- Risiko des direkten Arbeitens auf einem zentralen Server