

# PROJEKT ENGINEERING

## Informationsaustausch

Herwig Mayr

Fakultät für Informatik, Kommunikation und Medien  
Fachhochschule OÖ, Hagenberg

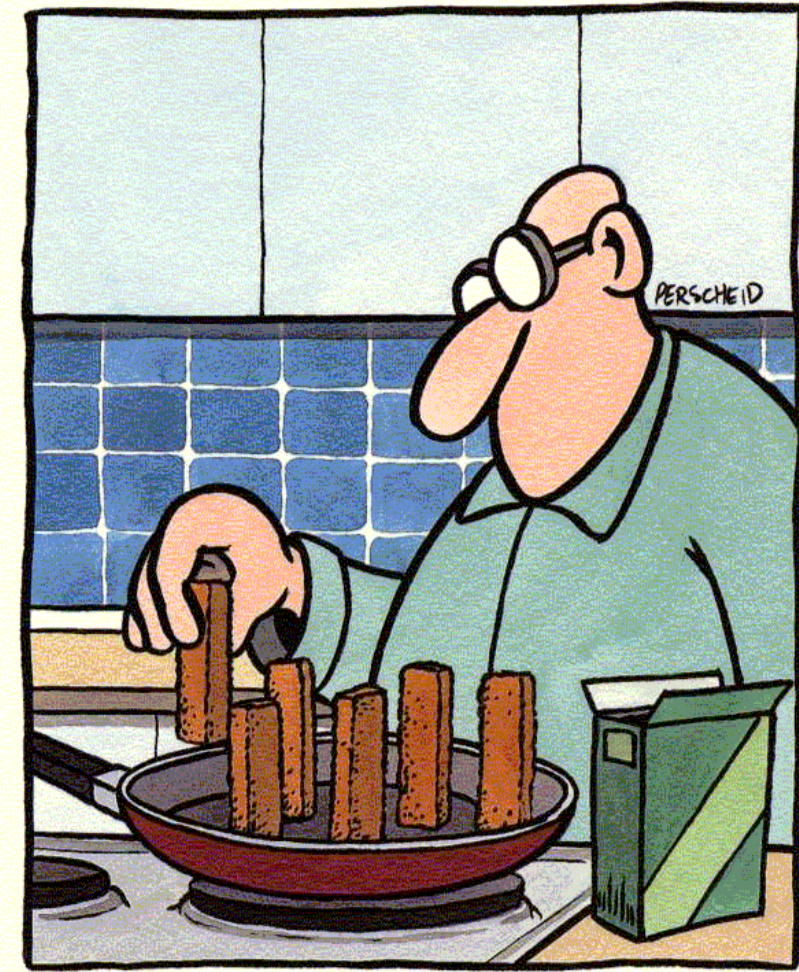
# Kommunikation

# Kommunikation in Softwareprojekten

mündlich wie auch schriftlich –  
alles Wichtige schriftlich festhalten!

Trends:

- Kundeneinbindung steigt (agiles Vorgehen)
- räumliche Distanz der Mitarbeiter steigt
- Komplexität von Werkzeugen und Ergebnissen steigt
- Entwicklungszeit sinkt



DIE FISCHSTÄBCHEN UNAUFGETAUT DER PACKUNG ENT-  
NEHMEN UND 5-7 MIN. VON ALLEN SEITEN BRATEN.

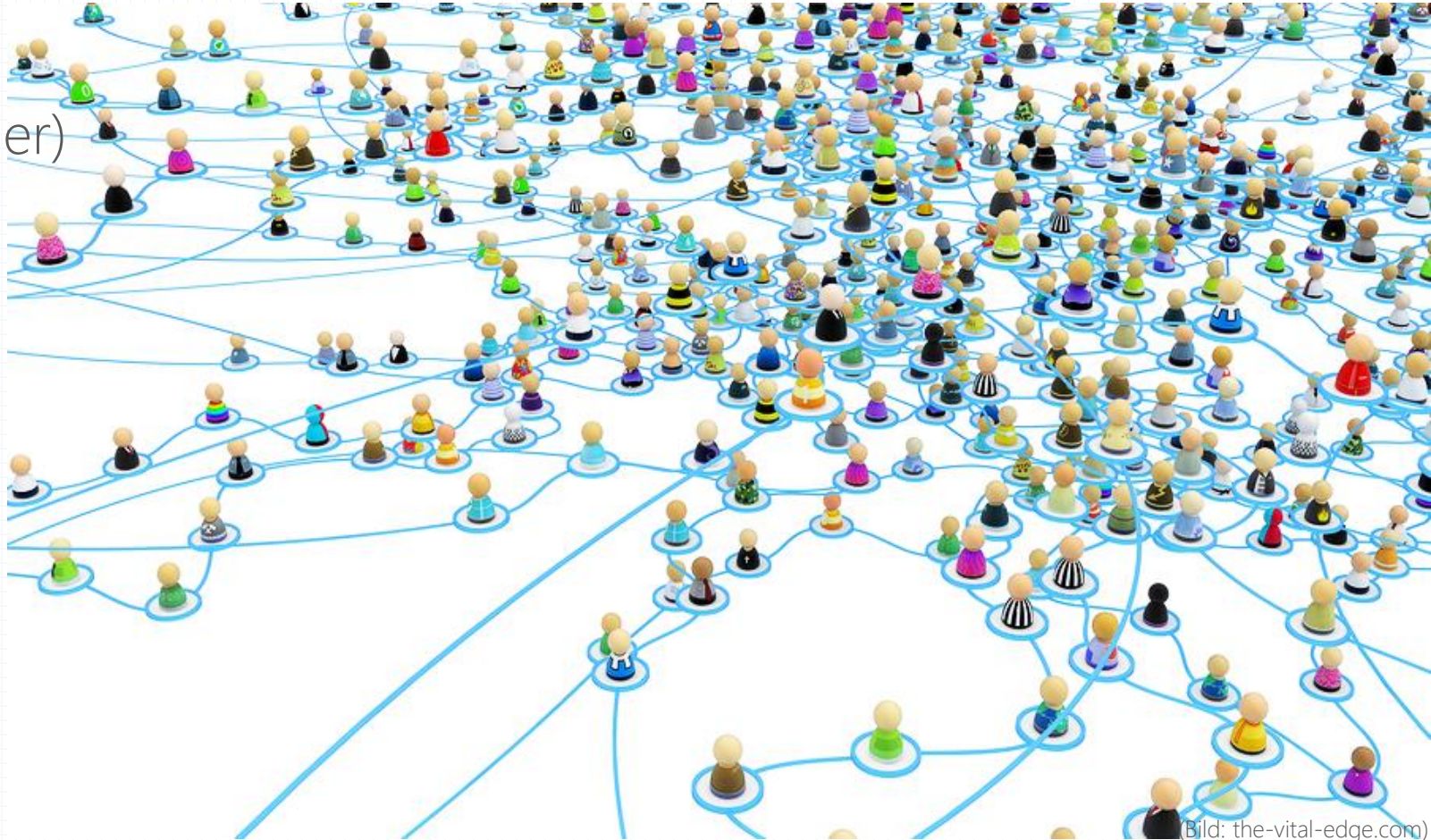
(Bild: [www.Martin-Perscheid.de](http://www.Martin-Perscheid.de))



# Kundeneinbindung (I)

**Stakeholder** = alle Beteiligten und Betroffenen eines Projekts

- Auftraggeber / Kunde(n)
- Auftragnehmer (Eigentümer)
- Anwender
- Entwickler, Projektleiter
- Sub-Auftragnehmer
- Zulieferer
- Investoren
- Fördergeber
- ...



(Bild: the-vital-edge.com)

# Beispiel: Wer sind Stakeholder Ihres Studiums?

Praktikumsgeber

Land (OÖ)

StudentInnen

Eltern/Familie

Lehrende

Projektpartner

FH-Bedienstete

Haustiere (FH-Katze)

z.B. auch Wirtschaftskammer, daran interessiert, in Zukunft gute Leute zu bekommen

# Kundeneinbindung (II)

## Value-Based Software Engineering (VBSE)



Bewertung der Anforderungen führt zu höherer Kundeneinbindung.

Dies bedingt:

- kompetenten und entscheidungsbefugten Auftraggeber
- gute Werkzeugunterstützung der Anforderungsanalyse (traceability)

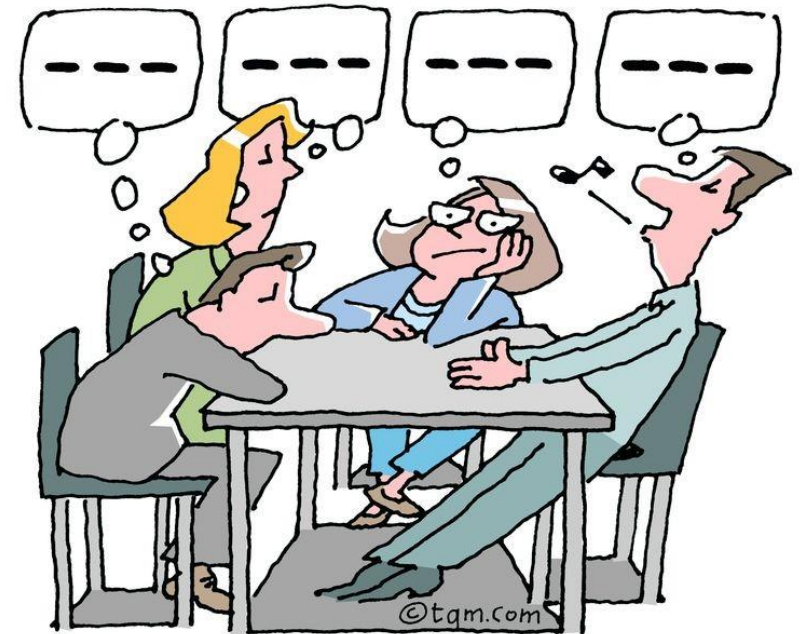
# Besprechungen (I)

Interne Projektbesprechungen: unmittelbare Fortschrittskontrolle

Externe Projektbesprechungen: Informationsaustausch mit Auftraggeber

Organisation der externen Besprechung:

- interne Vorbesprechung
- externe Besprechung
- interne Nachbesprechung



(Bild: [www.quality-web.de](http://www.quality-web.de))

# Besprechungen (II)

Aufgaben des Besprechungsleiters:

- Gesprächsgliederung gemäß Tagesordnung
- Moderatorfunktion
- Gespräch vorhersehen – Auswege und Alternativen anbieten

Besprechungen sind zu **protokollieren** (Stichwortprotokoll).

Für jede Besprechung ist die **Beginn- und Endzeit** vorher **bekannt** zu **geben**.



# Reviews (I)

formale Überprüfungen von Zwischenergebnissen oder Ergebnissen

- in Gruppen durchgeführt
- breit einsetzbar (Dokumente, Benutzerschnittstelle, Quellcode, ...)
- auch für halb fertige Produkte anwendbar
- Informationsaustausch in der Gruppe wird gefördert
- verschiedene Teamzusammensetzungen fürs Review möglich

Bei Reviews ist der Erfolg stark von der Akzeptanz bei den Gruppenmitgliedern abhängig!

# Reviews (II)

## Teamzusammensetzung

Teamzusammensetzung für Reviews		
bei Fagan	bei Kaner	bei Grady
Moderator Objektdesigner Objektcodierer Objekttester	Meeting-Manager Recorder Experten (bis 5)	Chief Moderator Moderator Scribe Owner Inspectors

# Werkzeuge (I)

„**Werkzeuge** sind Programme, die die Herstellung, Prüfung, Wartung und Dokumentation von Programmen vereinfachen, beschleunigen oder in ihrer Qualität verbessern.“ [P. Rechenberg]

**Computer-Aided-Software-Engineering-Systeme** (CASE-Systeme) sind breit einsetzbare Werkzeuge oder eine koordinierte Sammlung von Werkzeugen zur Entwicklung von Softwareprodukten oder -services.

**Application-Lifecycle-Management-Systeme** (ALM-Systeme) sind Werkzeuge, die neben der Entwicklung von Software auch den Betrieb (Wartung, Pflege) dieser Software unterstützen. ALM-Systeme lösen zunehmend CASE-Systeme ab.



Visual Studio ALM +  
Team Foundation Server



## Werkzeuge (II)

Softwareentwicklungsumgebungen sind integrierte Entwicklungssysteme. Sie umfassen:

- Planung („Projektmanagement“)
- Prototyping
- Analyse, Design („Roundtrip-Engineering“)
- Implementierung
- Testfallgenerierung, -durchführung
- Konfigurations- und Versionsverwaltung
- Dokumentation



Beispiel: Mit welchen Softwareentwicklungsumgebungen haben Sie bereits gearbeitet?



# Werkzeuge (III)

Empfehlungen für eine sinnvolle Werkzeugnutzung:

- Entwicklungsprozess von Werkzeugen, Modellen und Sprachen trennen
- durchgehend nutzbare Werkzeuge verwenden
- frühzeitig mit der Werkzeugnutzung beginnen („von Anfang an“)
- iterativ-inkrementelles Vorgehen muss unterstützt werden
- Schulungen einplanen
- Releasewechsel geordnet durchführen (nur wenn notwendig)
- Projektfortschritt und Grad der Zielerreichung prüfen

Werkzeugnutzung ist gerade bei agilem Vorgehen essenziell!

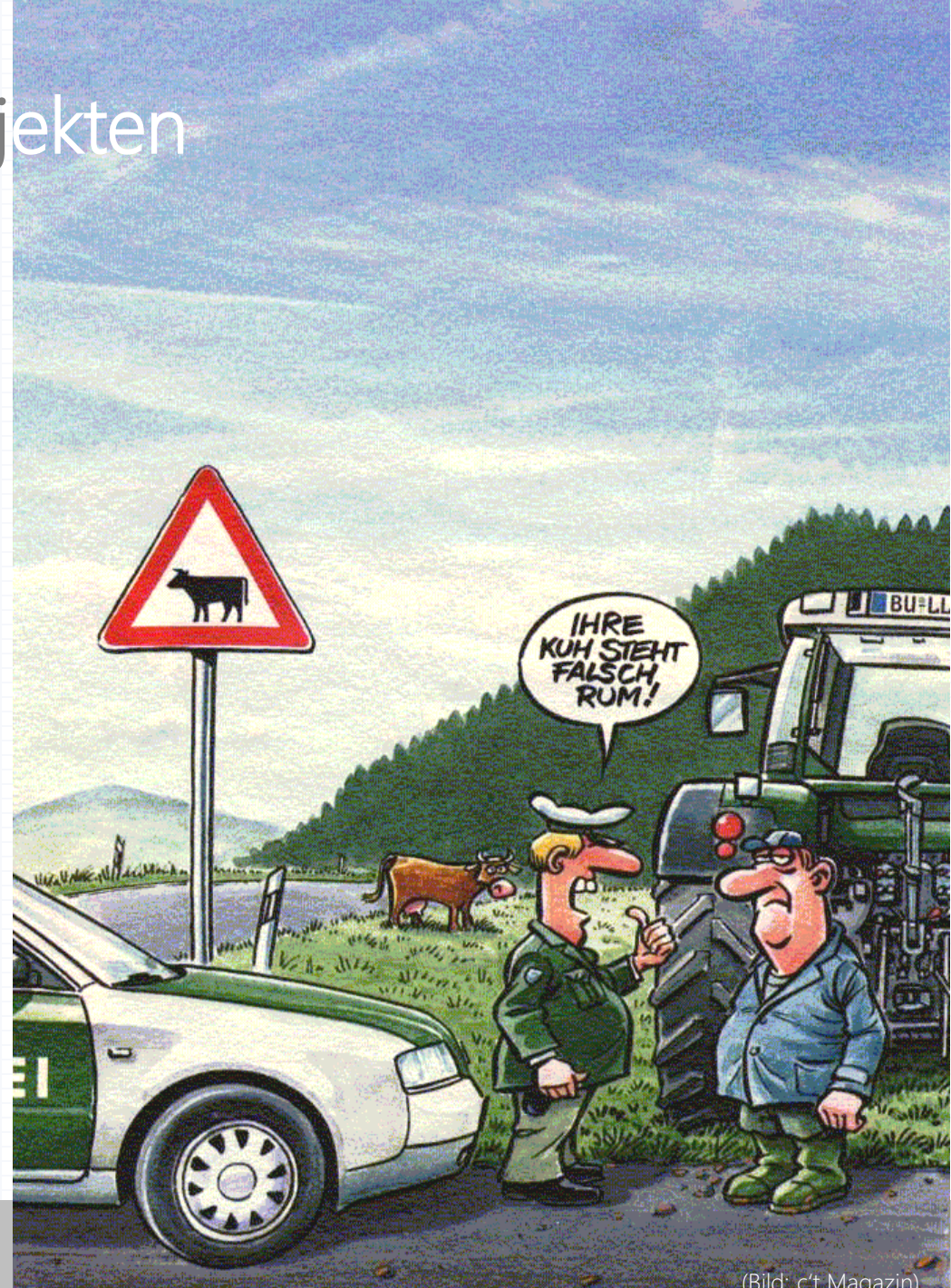
# Dokumentation

# Dokumentation von Softwareprojekten

umfasst Projektdokumentation  
(„organisatorische Dokumentation“)

Meeting: Wer war dabei, was wurde besprochen

und Produktdokumentation  
(„technische Dokumentation“)



# Dokumentationsplanung

## Ziele:

- einheitliche Dokumentstruktur
- rasche Erstellung neuer Dokumente
- einfache Zusammenführung von Dokumenten (deshalb braucht man Standards --> font size, font etc.)
- aktuelle Information über Projektstand (Projektkontrolle)
  - Wer arbeitet aktuell an einem Projekt, wieviele Stunden bereits gearbeitet

## Aufgabe:

- Erstellung von Standards, Strukturen und Vorgaben

# Berichtswesenplanung

Die Berichtswesenplanung legt den **Kontrollfluss für Informationen** fest, d. h. wer von wem wann welche Informationen erhält).

Folgende Entscheidungen sind für jedes Dokument zu treffen:

- Wann ist es zu erstellen?
- Wer erstellt es?
- Wer empfängt es?
- Wie oft ist ein Update notwendig?
- Sollen Versionen gehalten werden?
- Welches Format ist zu wählen?



# Essenzielle Dokumente (I)

Effizienter **Werkzeugeinsatz** ist bei agilen Methoden essenziell!

Wichtiger bei traditionellem Vorgehen	Gleich wichtig	Wichtiger bei agilem Vorgehen
	Organigramm, Stellenbeschreibungen	
	Projektbibliothek, Projekttagbuch, Protokolle	
Fortschrittsberichte	Zwischenprodukte, Prototypen (Protokolle)	
	Konfigurationsmanagement	(Werkzeug!)
	Qualitätsmerkmale	
	Zielbeschreibung	
Lastenheft	Auftrag, Anforderungskatalog	(Werkzeug!)
Gesamtplan	Grobplan, kurzfristiger Feinplan	(Werkzeug!)
		Risiko-Vorsorgeplan

Werkzeug: tool

# Essenzielle Dokumente (II)

Wichtiger bei traditionellem Vorgehen	Gleich wichtig	Wichtiger bei agilem Vorgehen
Benutzerdokumentation	Zwischenprodukte, Prototypen	
Pflichtenheft	Designmodell (Werkzeug!), durchg. Modellsprache	
	Quellcode	
Systemdokumentation		(Anwendungs-)Daten
	Lauffähiges Programm	
		Testplan, Testsuite
Fehlerliste, Fehlerbericht		Fehlerbehandlung (Werkzeug!)
	Fehlerlogbuch	
	Betriebsversion	
	Installations-, Inbetriebnahmeprotokoll	
	Abnahmeprotokoll	
Abschlussbericht	Wartungs-, Pflegeplan	
	Projektarchiv	

# Essenzielle Dokumente (III)

Alle „e“ sind eigene  
(externe)  
*Anforderungen*, die zu  
*Aufgaben* führen!

Dokument	Sichtbarkeit	Werkzeug
Projektbibliothek	i	CM-Tool, QM-Tool / Tab.kalk.
Organigramm	i	Textverarbeitung
Stellenbeschreibungen	i	Textverarbeitung
Agenden, Protokolle	i / e	Textverarbeitung
Prototypen	i / e	div. Prototyp.-Tools, Entw.umd.
Risikoplan	i	Risiko-Mgmt.-Tool / Tab.kalk.
Zielbeschreibung	e	Textverarbeitung
Tagebuch	i	Textverarbeitung / Tab.k. / DB
Anforderungen	e	Textverarbeitung / Tab.k. / DB
Projektplan	i	Projektmanagement-Tool
Aufgaben	i / e	Textverarbeitung / Tab.k. / DB
Fortschrittsdokumentation	i / e	Textverarbeitung
Benutzerdokumentation	e	Textverarbeitung
Quellcode	i / e	Entwicklungsumgebung
Systemdokumentation	i / e	Textverarbeitung
Testdokumentation	i / e	Test-Tool, Bug Tracker
Programm	e	Entwicklungsumgebung
Abschlussbericht	i	Textverarbeitung
Archiv	i	-

# Dokumentationsorganisation

Festlegung der zu erstellenden Dokumente

Für jedes Dokument sind festzulegen:

- Arten und Standards
- Erstellungs- und Freigabestatus
- Ablage, Archivierung

Wichtige Prinzipien:

- mitlaufende Dokumentation sichern
- Redundanz vermeiden
- einheitliche Bezeichnungen verwenden
- systematische Ablage einhalten



# Dokumentablage

für sicheres **Ablegen** und rasches **Wiederauffinden** von Dokumenten  
(Werkzeuge verwenden!)

Folgende Entscheidungen sind für jedes Dokument zu treffen:

- Ablageort? (lokal, Intranet, Cloud,...)
- Ablageformat? (Originalformat, strukturiert – XML, PDF/A,...)
- Indexierung für Suche (Beschlagwortung, Volltextzugriff, ...)
- Zugriffsschutz?
- Sekundärsicherung? (z.B. extern)
- Wartung der Ablage? (z.B. Transferieren auf neues Medium alle 5 Jahre)
- Ablagedauer? (Entsorgung bzw. sicheres Löschen danach)



(Bild: Computerwoche)



# Beispiel: Wie gestalten Sie Ihre persönliche Dokumentablage (im Studium)?

# Einschub: Konfigurationsmanagement

# Motivation

Programme gab es immer in Versionen und wird es immer in Versionen geben ...



# Konfigurationsmanagement (I)

## Definition Konfigurationsmanagement:

„...ist derjenige Aufgabenbereich in der Projektentwicklung, der für die **Verwaltung und Speicherung aller** innerhalb des Projekts anfallenden **Zwischen- und Endergebnisse** und deren **Versionen** verantwortlich ist.“ [IEEE]

## Erkenntnis:

*Jedes* Softwareprojekt braucht eine Konfigurationsverwaltung ... **händisch** oder **werkzeugunterstützt!**

-> Werkzeugunterstützung möglichst ab Projektbeginn!



# Konfigurationsmanagement (II)

## Aufgaben

- **Versionsverwaltung:** Versionen von Dokumenten und Code, Zugriffsrechte (Teamprojekte!)
- **Konfigurationsverwaltung:** Revisionen, Varianten, Releases
- **Build-Unterstützung:** Continuous Integration, Continuous Release
- **Workflow-Unterstützung**
- **Änderungsmanagement:** Anforderungen, Fehler, Erweiterungen
- **Auswertungen**

# Beispiel: Wie gestalten Sie Ihr persönliches Konfig.management?

Versionsverwaltung

Zugriffsschutz

Änderungsmanagement

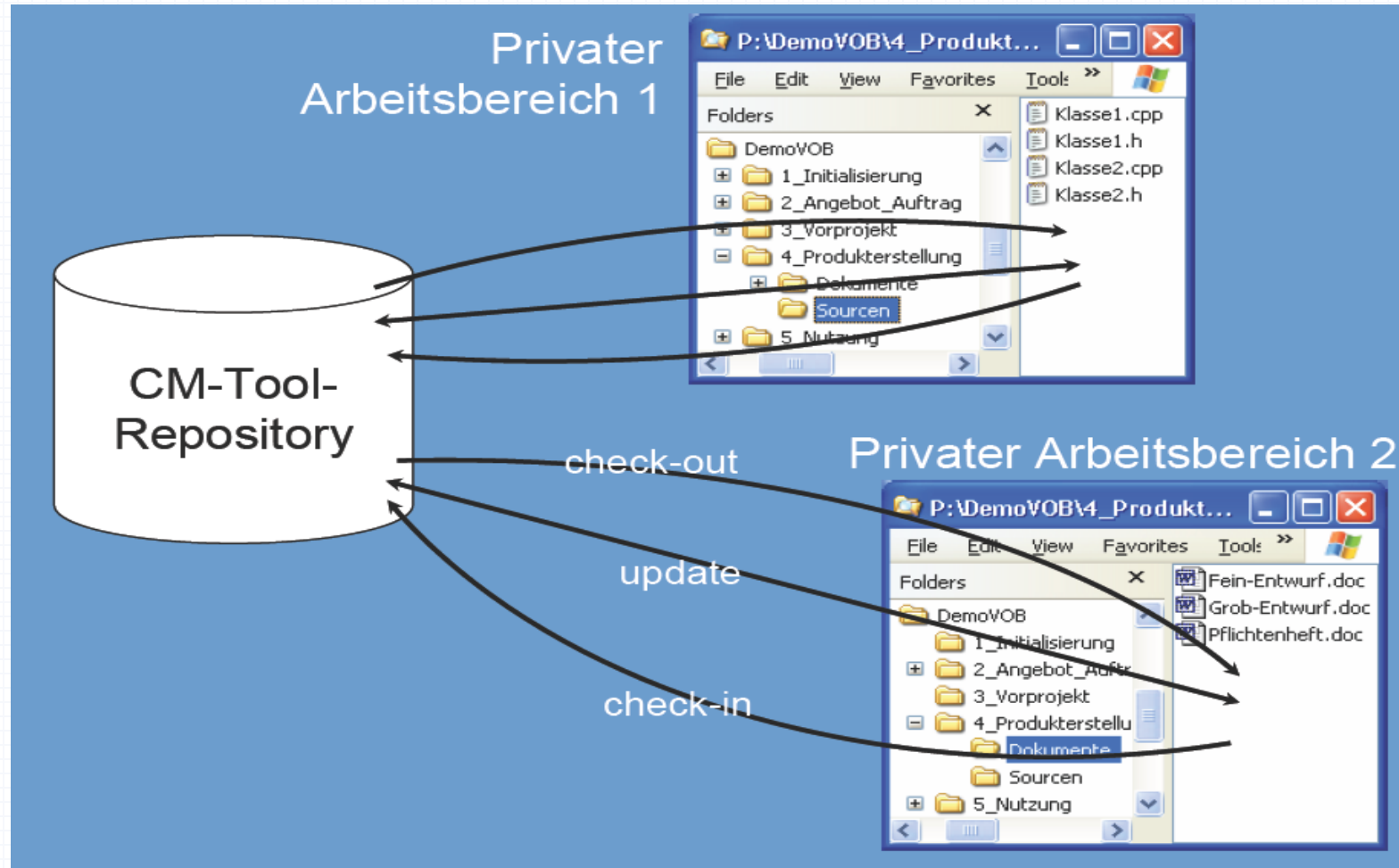


# Konfigurationsmanagement (III)

## Nutzen

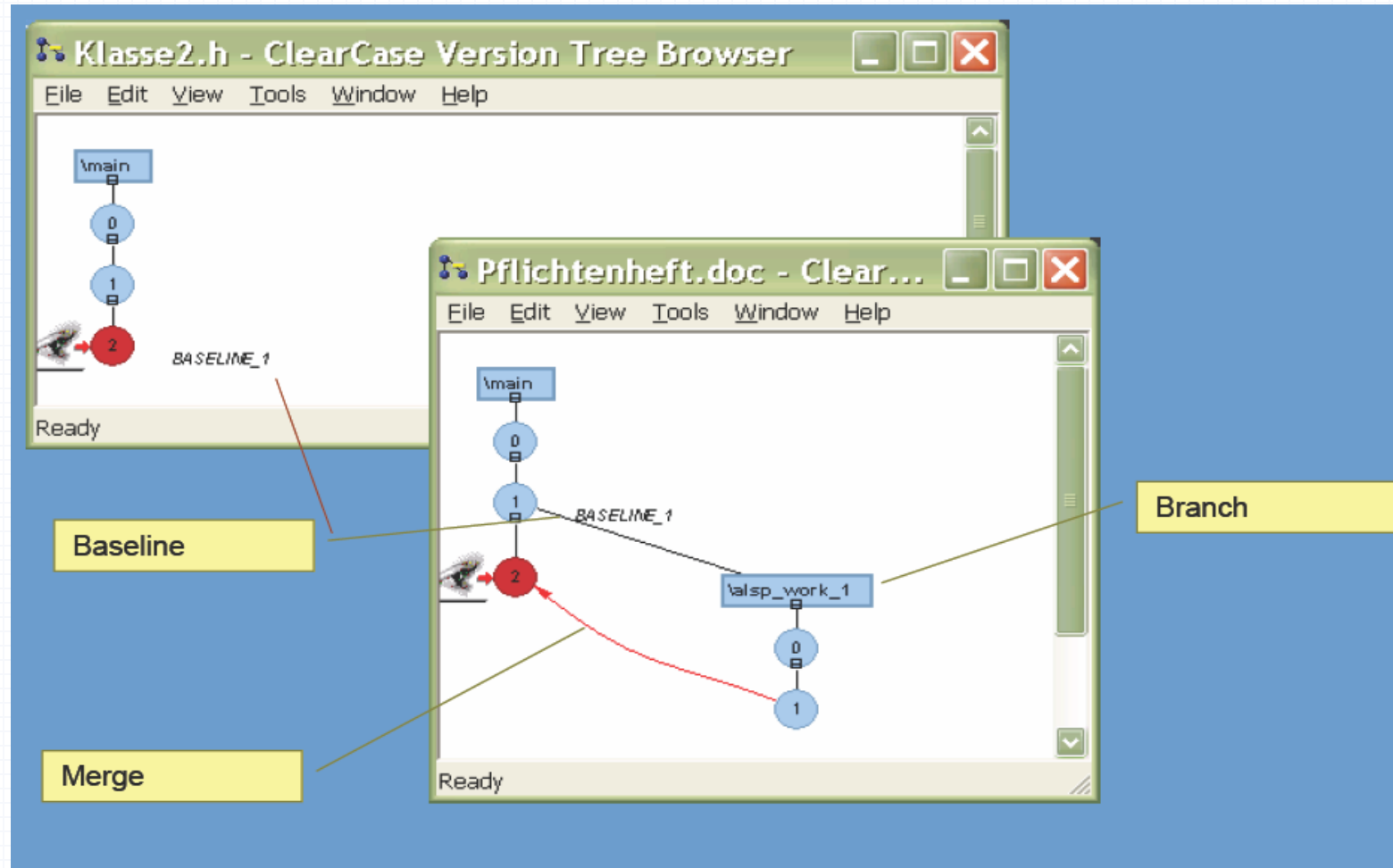
- **Entwickler**: koordiniertes tägliches Arbeiten im Team
- **Projektleiter**: tagesaktuelle Auswertungen, passende Dokumentversionen
- **Tester**: bessere Kommunikation mit Entwicklern, systematisches Änderungsmanagement
- **Qualitätsbeauftragter**: Messwerte zur Qualitätsbeurteilung
- **Verkäufer**: geordnete Erstellung von Releases, überprüfbares Einbinden bestimmter Features

# Begriffe: Check-in – Check-out – Update



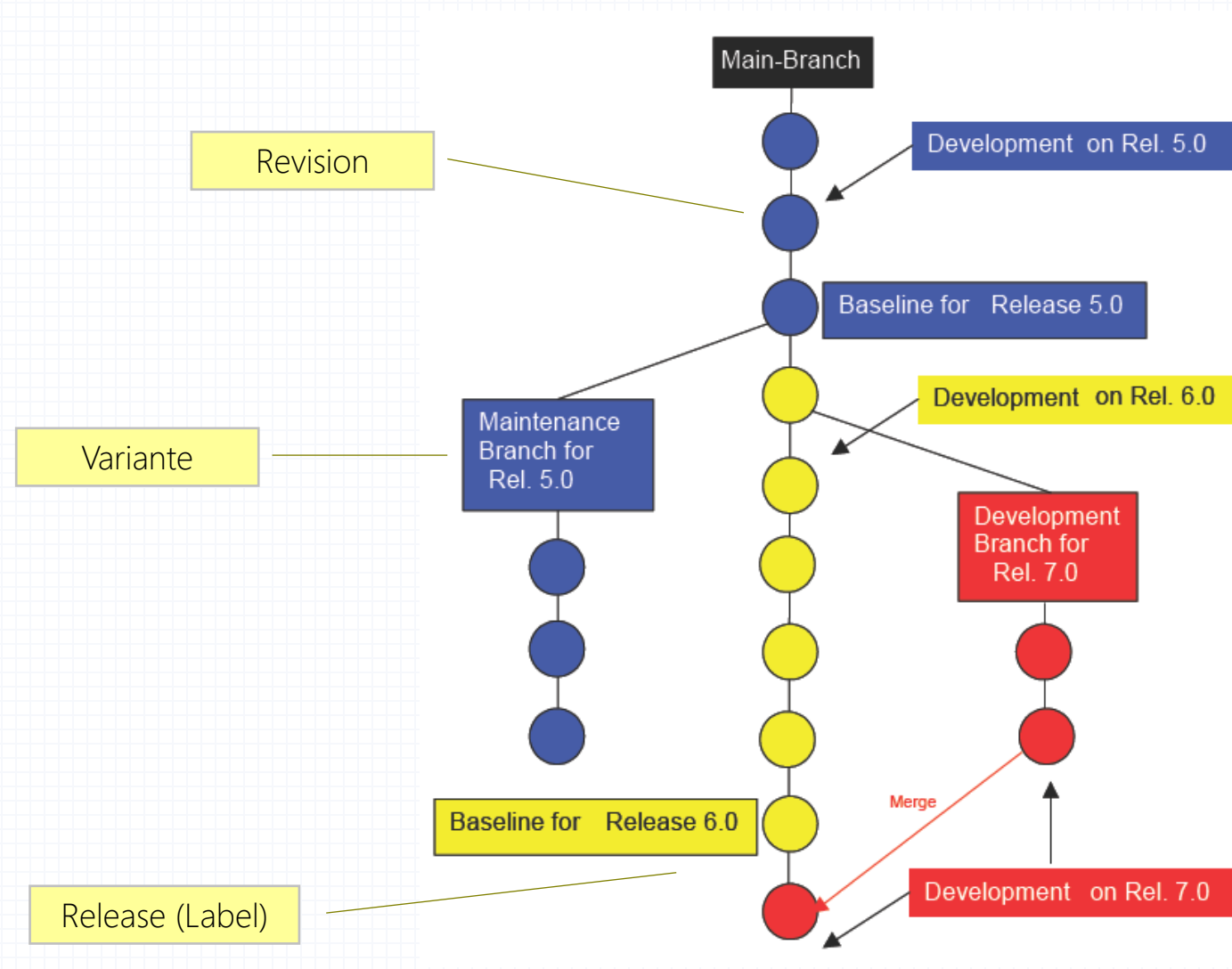
(Bild: Rational ClearCase; IBM)

# Begriffe: Baseline – Branch – Merge



(Bild: Rational ClearCase; IBM)

# Begriffe: Revision – Variante – Release



# Benennung von Versionen (Empfehlung)

Varianten werden textuell bezeichnet:

- z.B. Sprache (deutsch, englisch) oder Plattform (iOS, Android)

Revisionen werden in der Form MAJOR.MINOR nummeriert:

- MAJOR wird erhöht, wenn sich für den Kunden geplante Anpassungen oder Erweiterungen ergeben.
- MINOR wird erhöht, wenn (kleinere) Korrekturen durchgeführt werden.
- Vgl. – für Quellcode – „Compatible Versioning“: <https://gitlab.com/staltz/comver>.

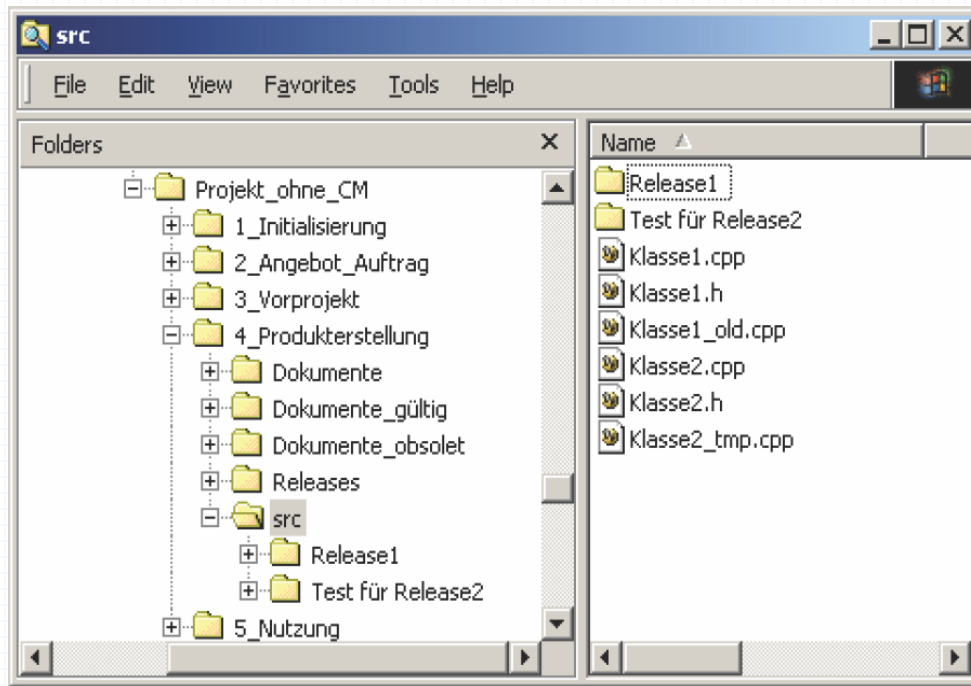
- Beispiele:

v. 0.1	v. 0.5
v. 0.9	v. 0.91
v. 1.0	v. 1.1
v. 2.0	

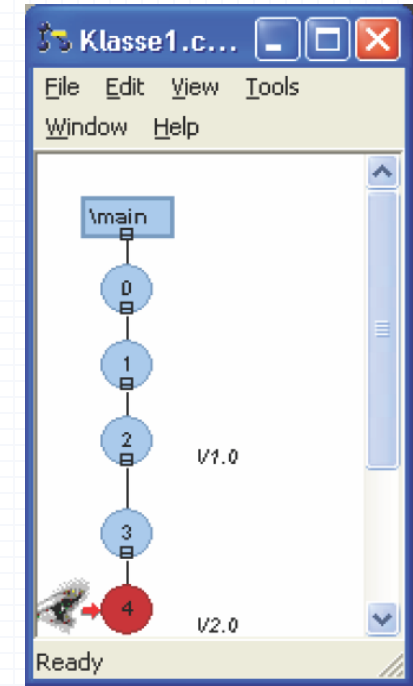
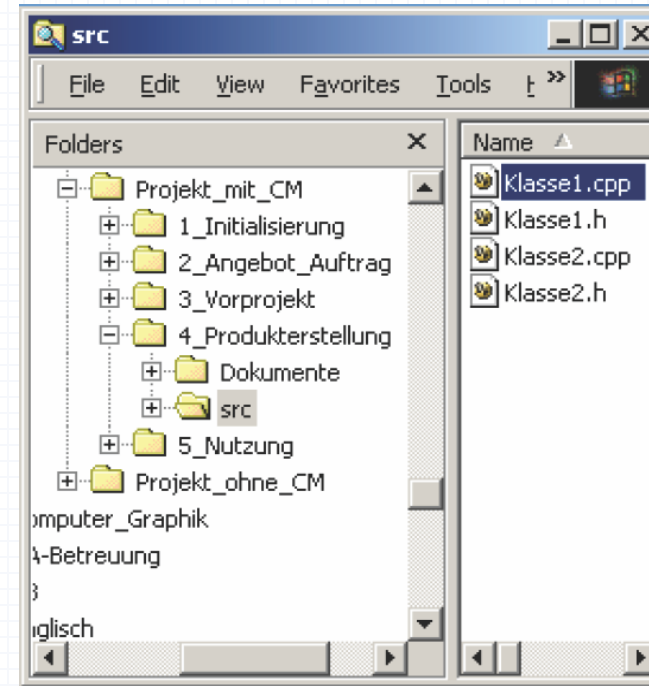
zweites Semester Versionen richtig benennen!

# Versionsverwaltung

ohne Werkzeug:



mit Werkzeug:

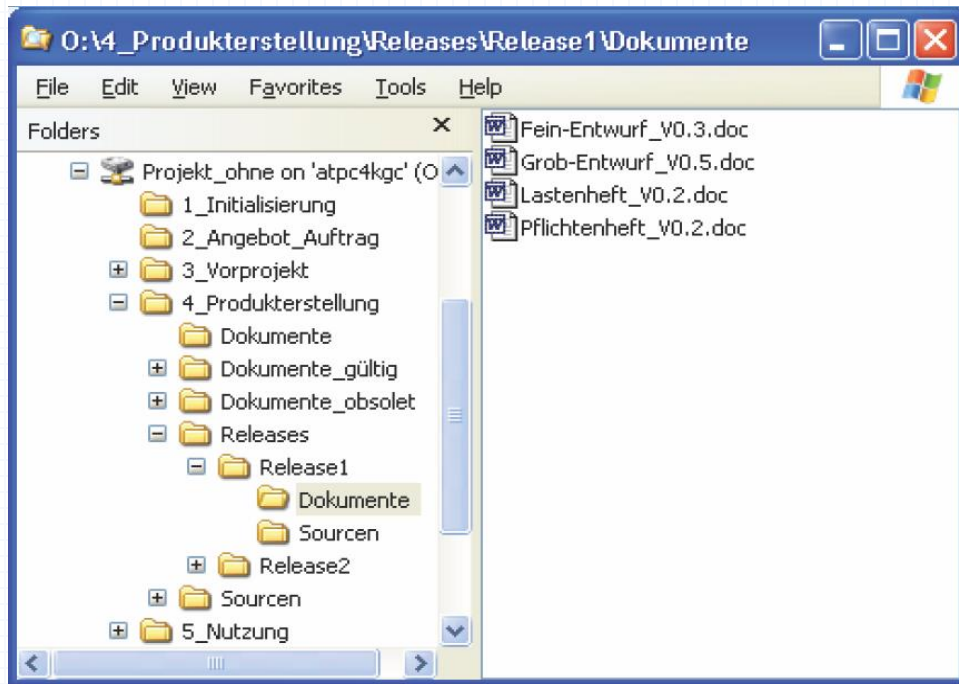


(Bilder: Rational ClearCase; Siemens PSE, Linz)

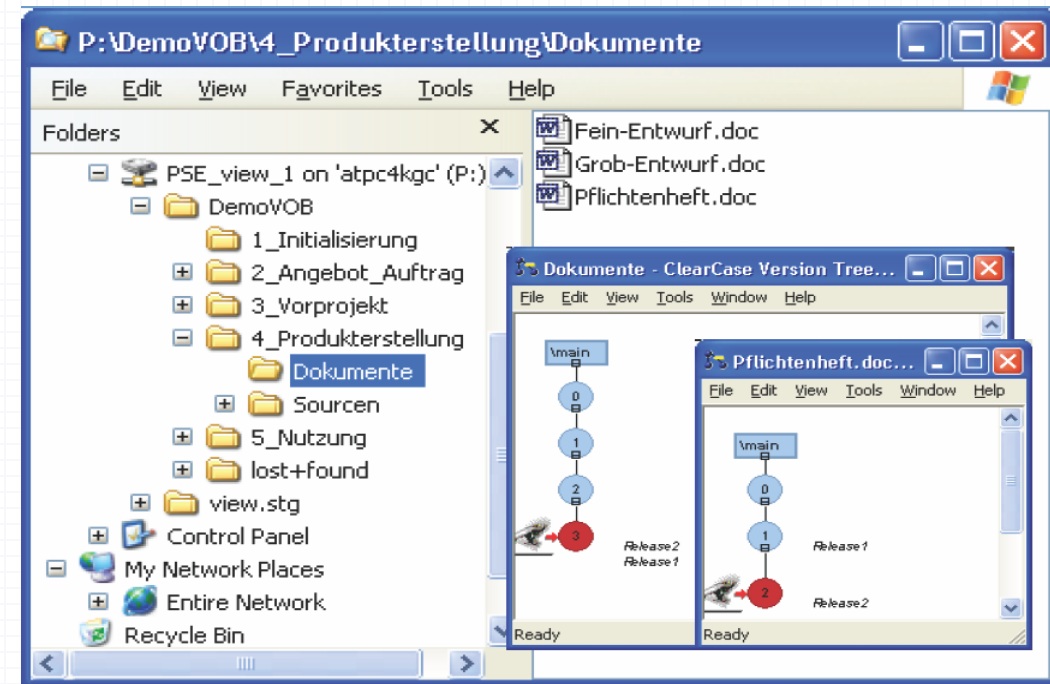


# Konfigurationsverwaltung

ohne Werkzeug:



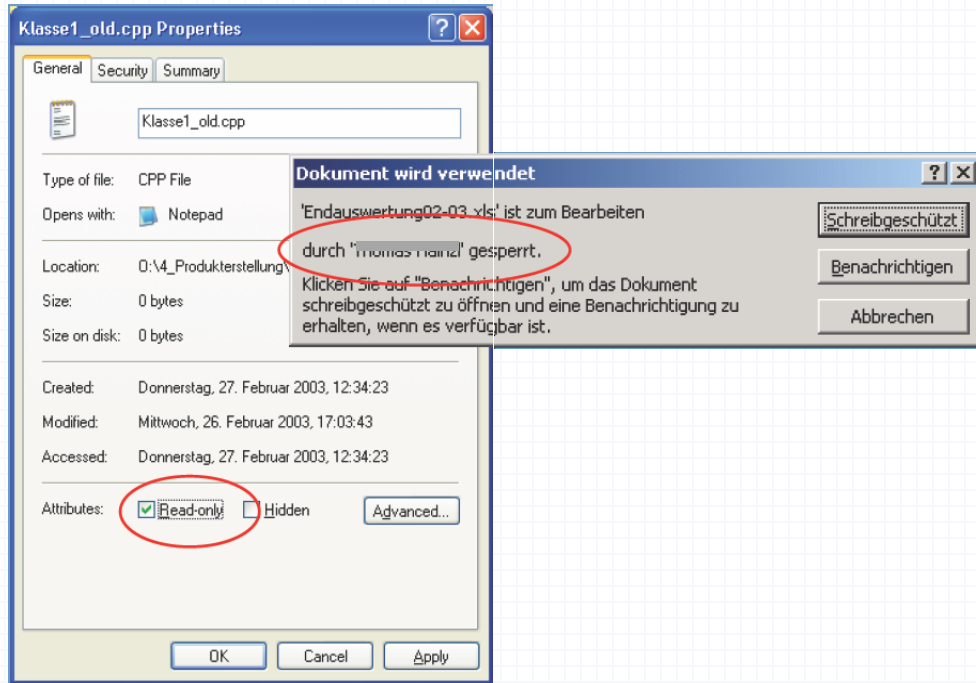
mit Werkzeug:



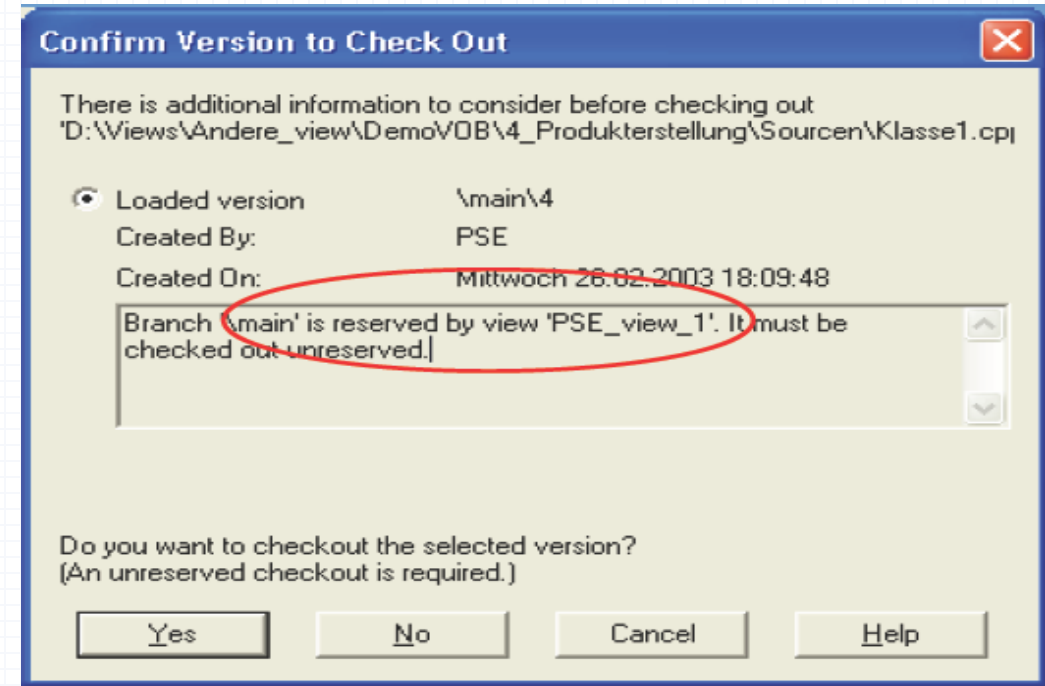
(Bilder: Rational ClearCase; Siemens PSE, Linz)

# Zugriffsregelung

ohne Werkzeug:



mit Werkzeug:



(Bilder: Rational ClearCase; Siemens PSE, Linz)

# Auswertungen

ohne Werkzeug:

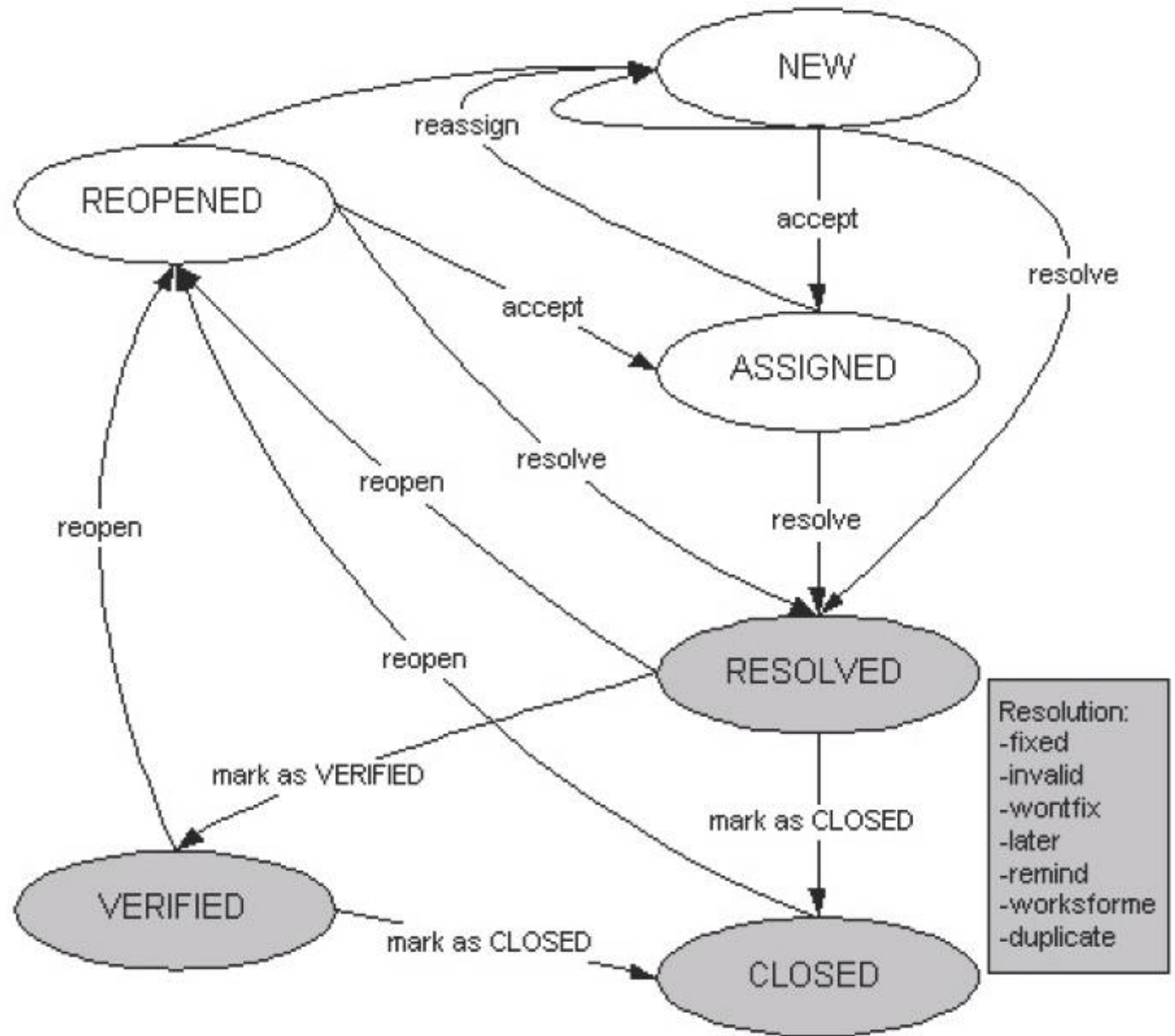
- **händisch:** Durchsuchen der Dokumente, Führen von Listen (Konsistenz, Aktualität?)
- **unterstützt durch Makros oder Scripts:** Suche in bestimmten Dateien (Führen von Tags notwendig)
- **Verwenden von Desktop-Suchen**

mit Werkzeug:

- **Attributieren der Elemente:** jeder Eintrag kann mit Attributen versehen werden; Definition von Suchabfragen möglich
- **Setzen des Status für jedes Dokument:** vorgefertigte Begriffe (requested, planned, in-work, in-review, released etc.); Definition von Abhängigkeiten möglich (Workflow!)
- **automatisches Erstellen von Listen:** periodisch und bei Bedarf

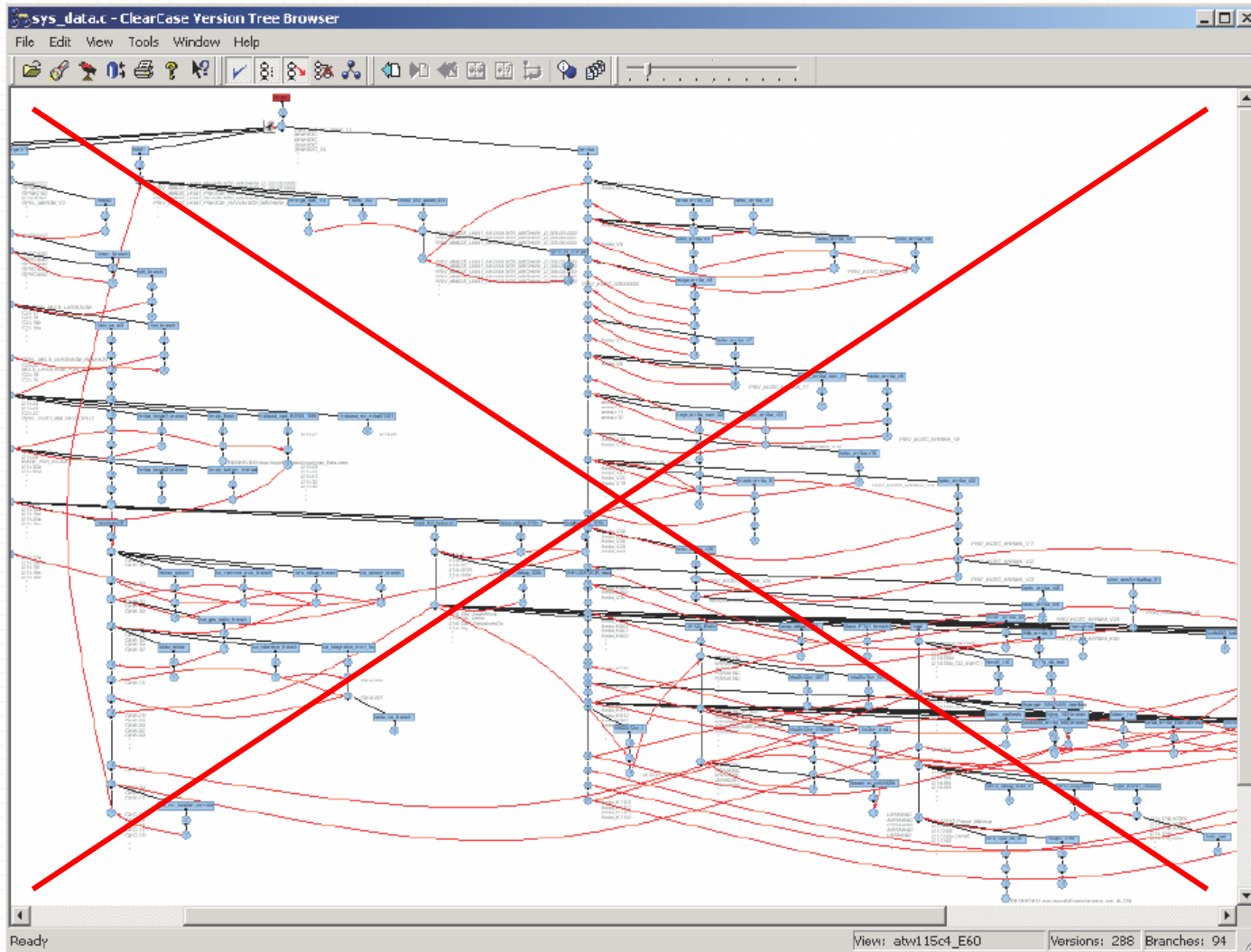
# Beispiels-Workflow

## Änderungsmanagement



(Bild: Rational ClearCase; IBM)

# Wenn ein Werkzeug ohne Experten ...



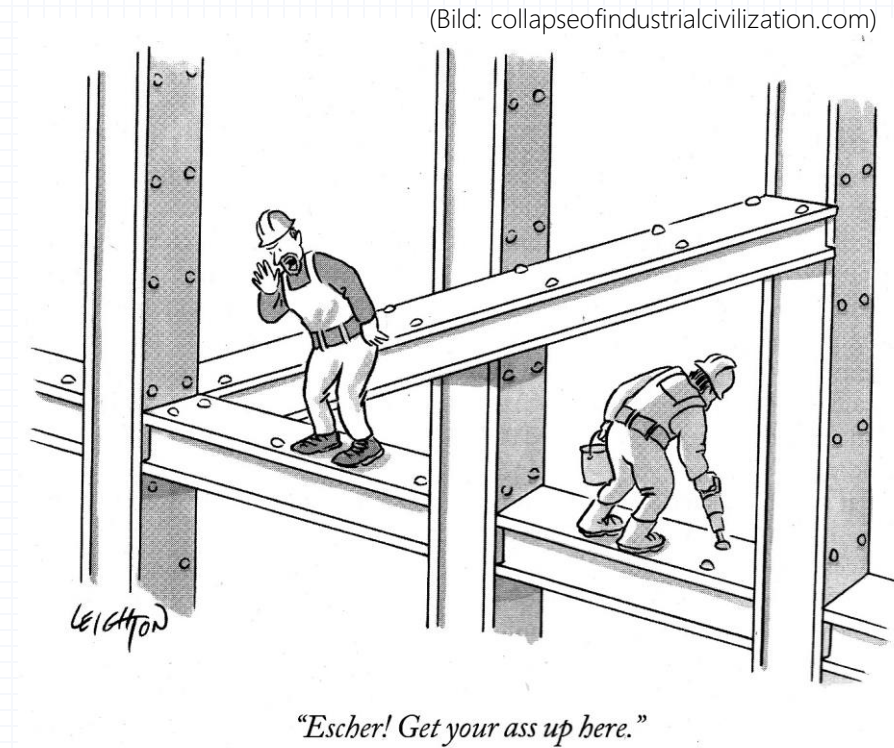
Verantwortlicher für  
Konfigurationsmanagement  
ist nötig!

(Bild: Rational ClearCase; Siemens PSE, Linz)

# Konfigurationsmanagement-Verantwortlicher

## Aufgaben

- Einrichten/Konfigurieren des KM-Werkzeugs
- Erstellen eines „Projekts“, „Workspaces“ etc.
- Erstellen und laufendes Aktualisieren des KM-Plans
- Schulen der Mitarbeiter
- Überführen bestehender Daten
- Überwachen der KM-Disziplin
- Anlaufstelle für Probleme mit dem KM-Werkzeug



-> Je besser ein Konfigurationsmanagement-Werkzeug funktioniert, desto weniger merkt man davon!

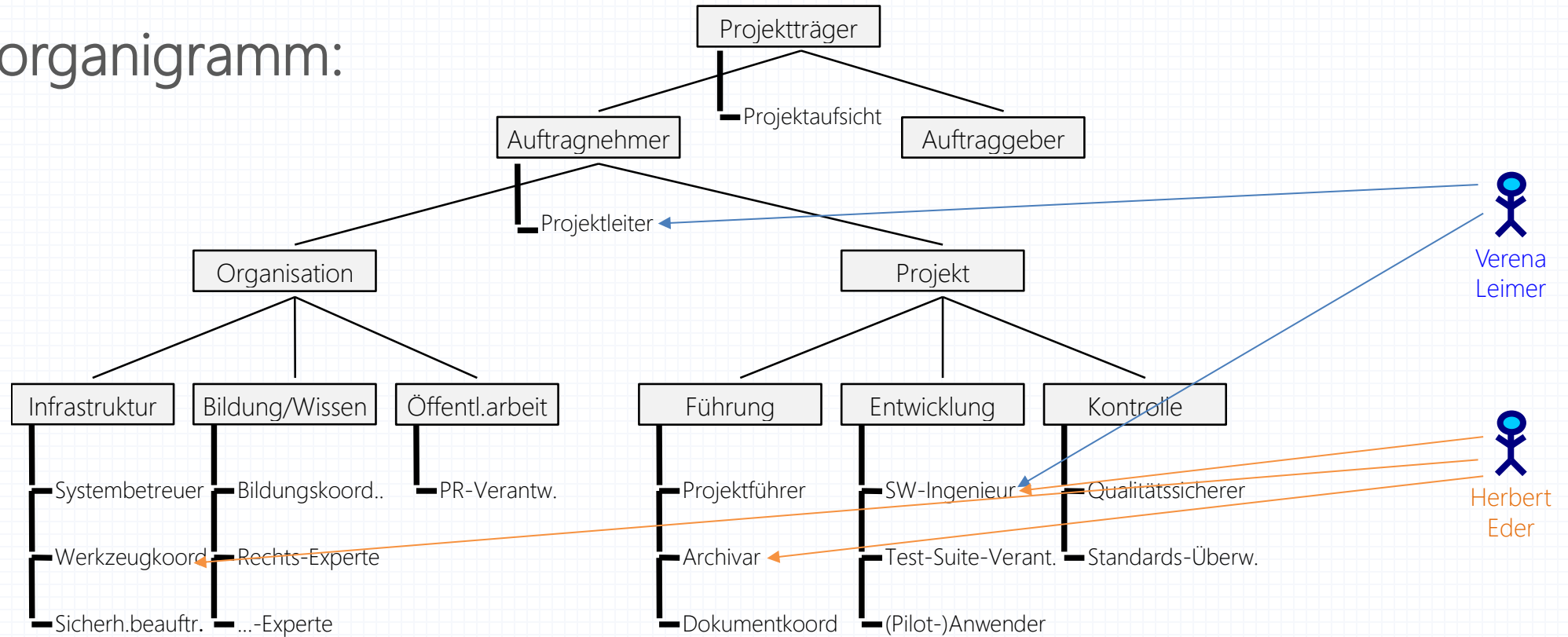


# Dokumente der Aufbauorganisation

# Organigramm

Darstellung von Gesamtstruktur und Stellen (Funktionen) in der Aufbauorg.

Beispielsorganigramm:



Stellen werden durch Mitarbeiter in bestimmten Rollen besetzt (n:m-Bezug).

# Stellenbeschreibung

Beschreibung von Aufgaben, Kompetenzen und Verantwortlichkeiten zur Schaffung klarer Zuständigkeiten

Eine Stellenbeschreibung enthält:

- Stellenbezeichnung
- Aufgaben der Stelle
- notwendige Qualifikationen
- Verweis auf Vorgesetzte(n)
- Regelung der Stellvertretung

Eine Stellenbeschreibung wird **je Stelle erstellt** (NICHT je Mitarbeiter/Rolle)!

The image shows a tilted template for a job description form. The title 'Stellenbeschreibung' is at the top right. Below it are several labeled fields: 'Stellenbezeichnung:', 'Bereich:', 'Stelleninhaber:', 'Stellenbezeichnung des unmittelbaren Vorgesetzten:', 'Stellenbezeichnung direkt unterstellter Mitarbeiter:', 'Der Stelleninhaber vertritt (Führungsverantwortung):', 'Hd.N.:', 'Hauptaufgaben der Stelle', and 'Zeitanteil in %'. The 'Hauptaufgaben der Stelle' section is a large grid with many rows and columns for detailed task description.

(Bild: blog.nyenhuis.org)

# Beispiel: Erstellen Sie für Ihr LEGO-Projekt Organigramm und Stellenbeschreibungen

# Dokumente der Ablauforganisation

## 7.4.1 Grundlegende Dokumente

**Ziel:** einheitliche Gliederung der Dokumente

Beispiele	Brief	Fax	Aktennotiz	Telefon- Gesprächsnotiz	Agenda	Protokoll	Bericht
Erstellungs-/Updatedatum	✓	✓	✓	✓	✓	✓	✓
Revisionsnummer					✓	✓	(✓)
Autor/Protokollführer	✓	✓	✓	✓	✓	✓	✓
Kontaktadresse/Absender/Institution	✓	✓	✓	✓	✓	✓	✓
Telefonnummer Absender	✓	✓			✓	✓	(✓)
Faxnummer Absender	(✓)	✓					(✓)
E-Mail-Adresse Absender	(✓)	(✓)					(✓)
Projekt/Betreff	✓	✓	✓	✓	✓	✓	✓
Anlass (inkl. Beginn, Ende)	✓	✓	✓	✓	✓	✓	✓
Datum des Anlasses	(✓)	(✓)	(✓)		✓	✓	
Teilnehmer					✓	✓	
Tagesordnung					✓	✓	
Verteiler/Gespr.partner/Adressat	✓	✓	(✓)	(✓)	✓	✓	(✓)
Ablage (Hardcopy)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
Ablage (elektronisch)	✓	(✓)	✓	✓	✓	✓	✓



# Projekthandbuch

verbindliche Festlegung der Projektorganisation:

Das Projekthandbuch enthält:

- Organigramm, Stellenbeschreibungen
- Stellenzuordnungen
- Projektleitung, Führungsgrundsätze
- Sitzungskonzept
- Dokumentationsvorgaben, -standards
- Vorgehensmethode(n), Werkzeuge
- Vorgaben für Projektplanung (Arten, Umfang, Häufigkeiten)
- Festlegungen zur Qualitätssicherung



(Bild: [www.dom-publishers.com](http://www.dom-publishers.com))

# Projekttagebuch

entwicklerinterne, chronologische Aufzeichnung aller Projektstätigkeiten

Ein Eintrag enthält:

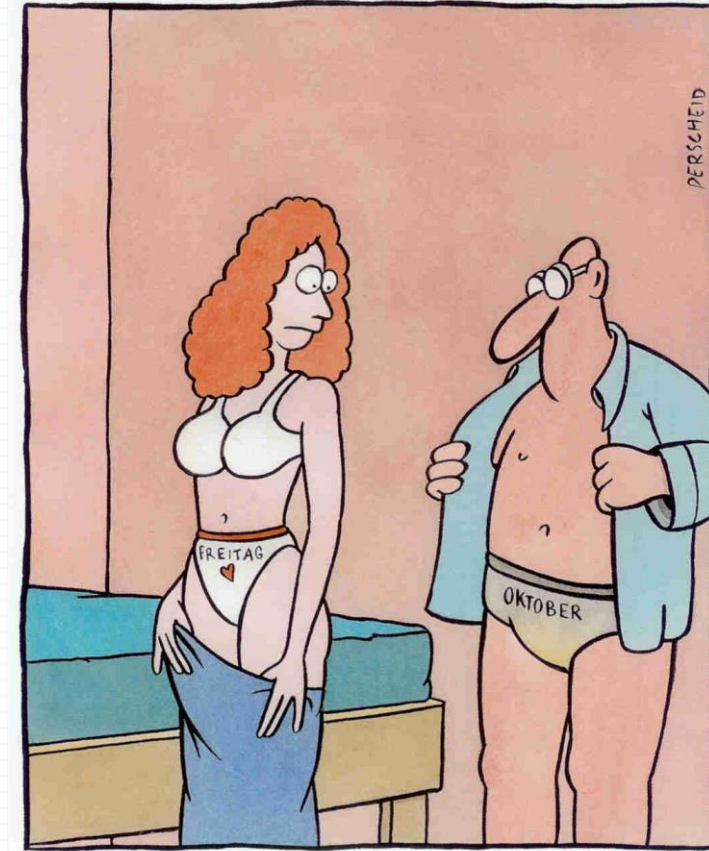
- Datum, (Uhrzeit)
- Beteiligte, (Ressourcen)
- Aktivität
- Aufwand

Projekttagebuch Projekt "_____"				
Datum	von-bis	Team (Kürzel)	Tätigkeit	Bemerkung

Das Projekttagebuch wird von allen Projektmitarbeitern gemeinsam geführt und ist laufend (täglich) zu aktualisieren.

# Projektfortschrittsbericht

- Terminbericht zum Zweck des Soll-Ist-Vergleichs
- ergeht von Projektleiter AN an Projektleiter AG oder Projektträger
- ist regelmäßig zu erstellen (z.B. alle 4 Wochen) sowie im Anlassfall (z.B. Revision)



(Bild: [www.Martin-Perscheid.de](http://www.Martin-Perscheid.de))

# Arbeitsbericht (Tätigkeitsbericht)

- Dokumentation einer abgeschlossenen Aufgabe
- von den involvierten Mitarbeitern gemeinsam zu erstellen

**Beispiel:** Abschlussbericht eines (Teil-)Projekts



(Bild: [www.brainzooming.com](http://www.brainzooming.com))

# Projektbibliothek

- Verzeichnis aller zum Projekt gehörigen Dokumente
- meist elektronisch geführt
- Dokumentstatus ist laufend zu aktualisieren



(Bild: bmvit.gv.at)

# PROJEKT ENGINEERING

## Informationsaustausch

Herwig Mayr

Fakultät für Informatik, Kommunikation und Medien  
Fachhochschule OÖ, Hagenberg