

## **Das Werkzeug Faktor-IPS**

### **Schulung: Arbeiten im Team**

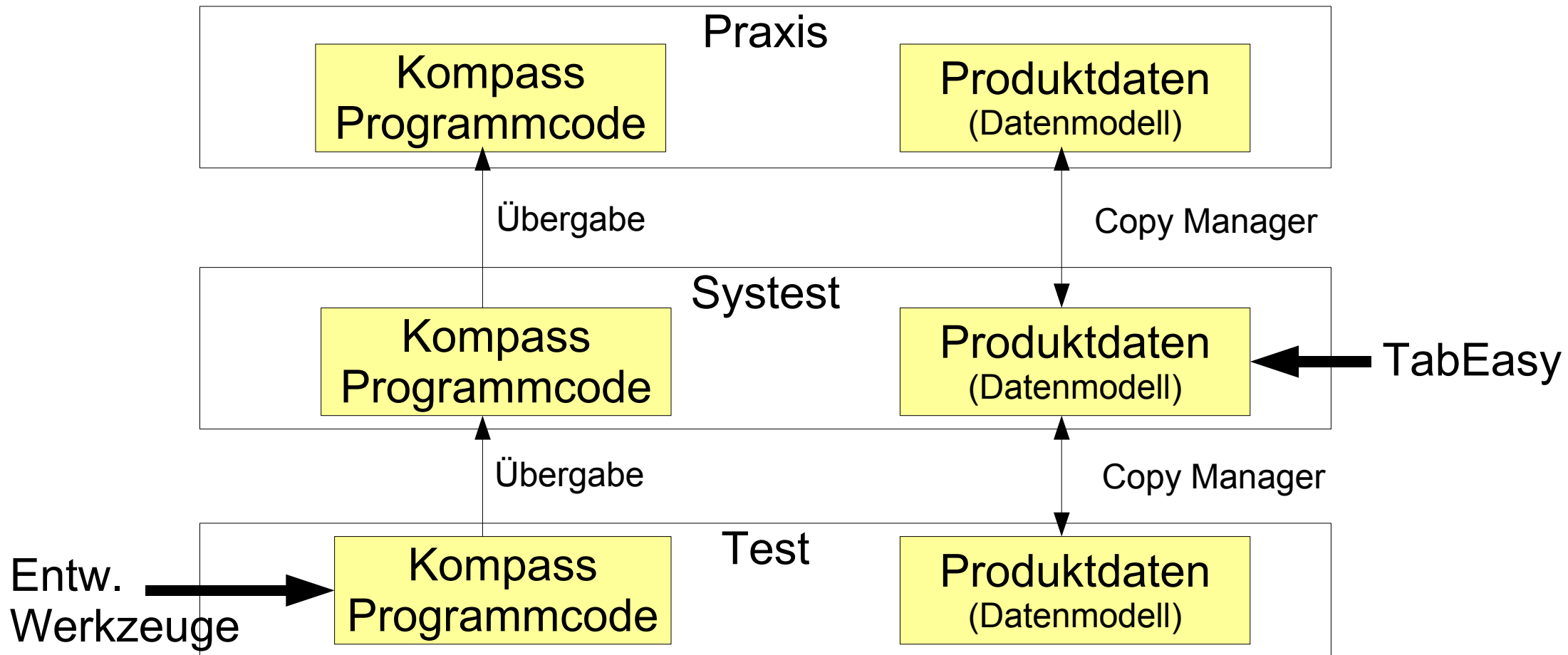
## Inhalt

- Grundlagen
- Bearbeiten von Produktdaten im Team
- Übergabeprozess

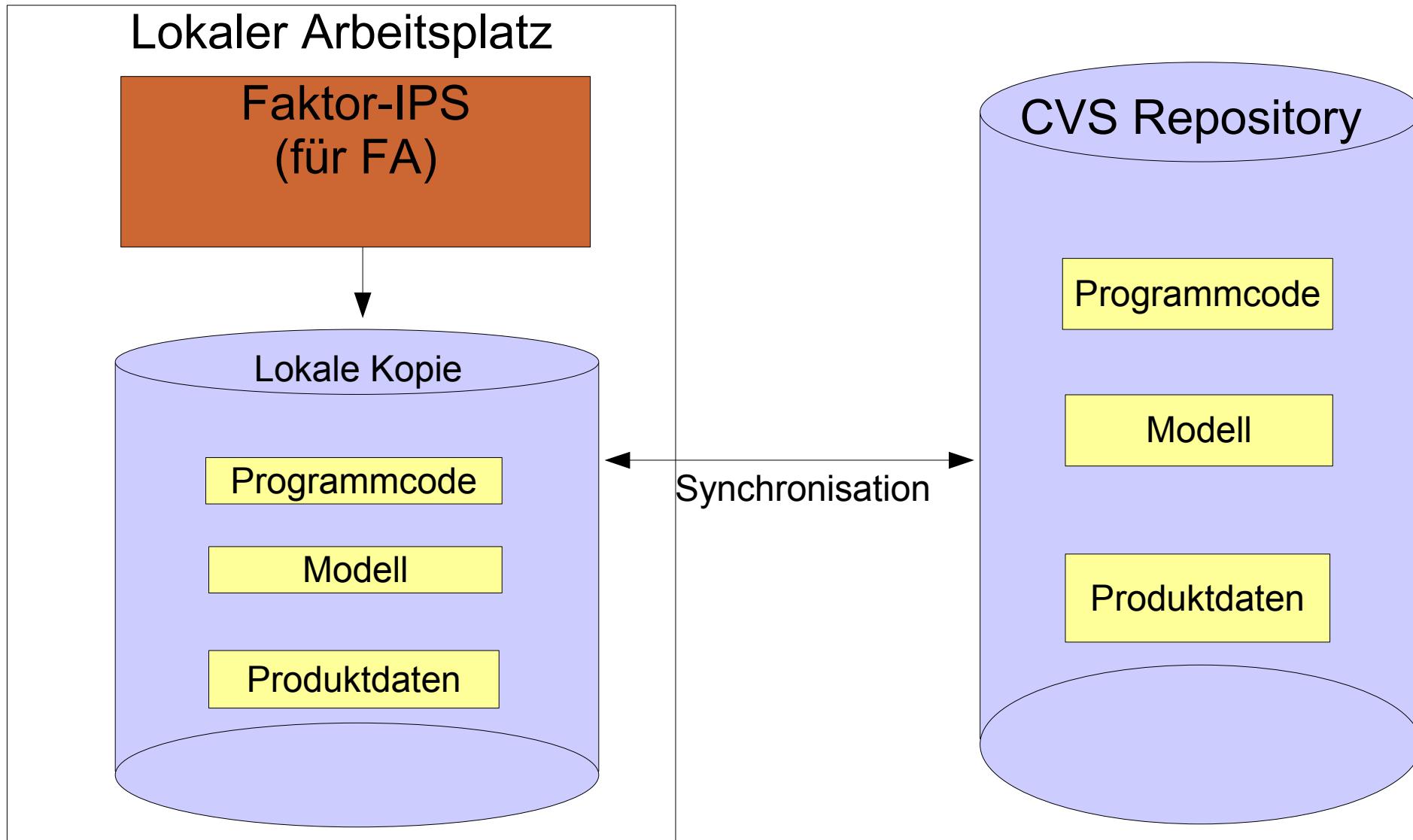
## Inhalt

- Grundlagen
- Bearbeiten von Produktdaten im Team
- Übergabeprozess

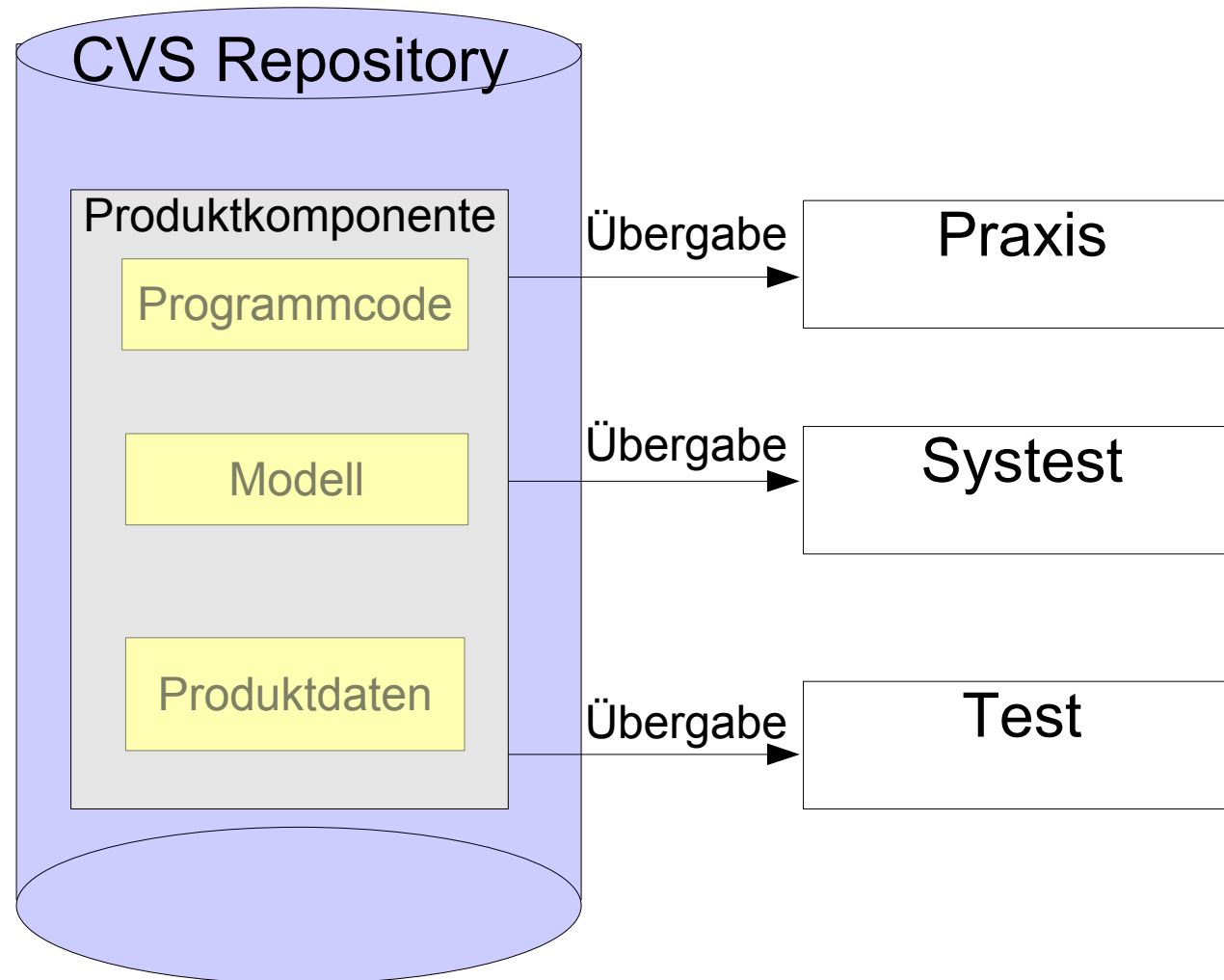
## Entwicklungsumgebung (vor Refactoring Kompass)



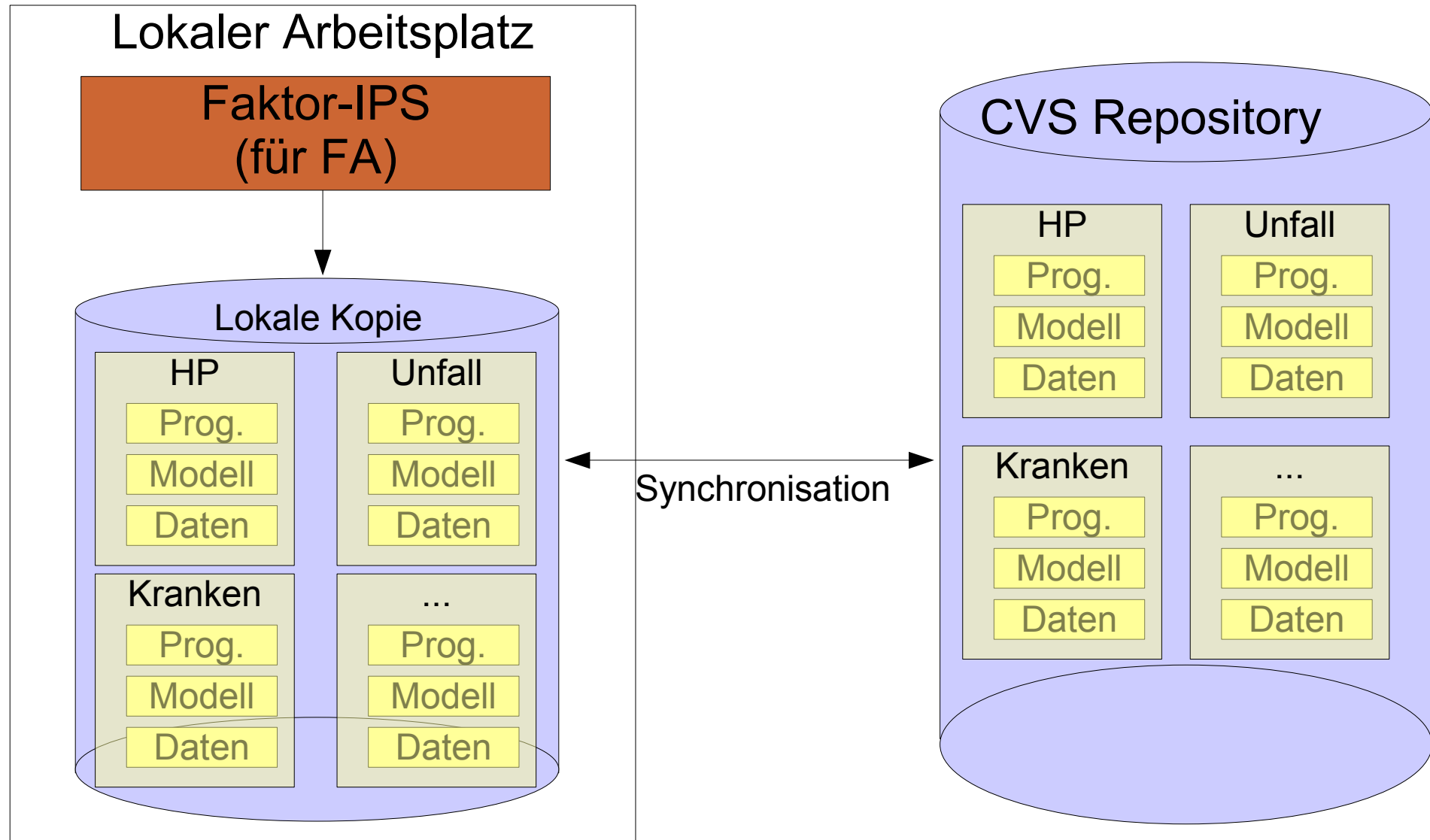
## Entwicklungsumgebung Produktkomponente



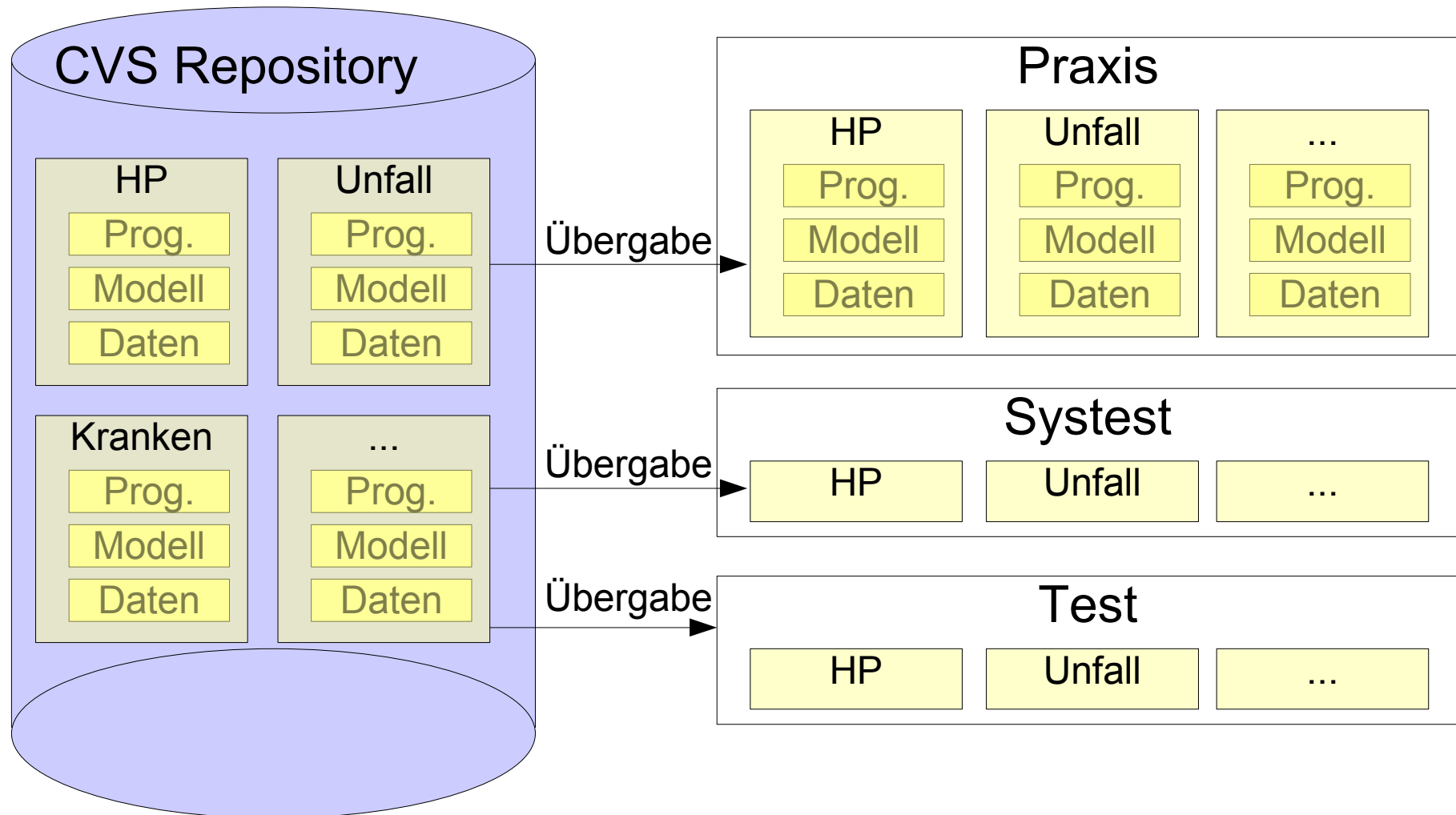
# Übergabe der Produktkomponente



## Trennung der Produktkomponenten nach Sparte



# Übergabe erfolgt getrennt pro Sparte





## Konsequenz aus der Trennung der Produktdaten nach Sparten

- Die Produktdaten der Sparten können völlig unabhängig voneinander bearbeitet werden.
- Jede Sparte kann völlig unabhängig von den anderen Sparten übergeben (deployed) werden.
- Parallele Änderungen an den Produktdaten durch mehrere Mitarbeiter müssen lediglich besonders abgestimmt werden, wenn sie innerhalb einer Sparte erfolgen.

## Inhalt

- Grundlagen
- Bearbeiten von Produktdaten im Team
  - Einrichten des Zugriffs auf das Repository
  - Grundfunktionen zum Arbeiten im Team
  - Lösung von „Konflikten“
- Übergabeprozess

## Einrichten des Zugriffs auf das Repository (1)

- Voraussetzungen:  
User und Passwort sind vorhanden.
- Wechseln in die Perspektive „CVS Repository durchsuchen“  
„Fenster->CVS Repository durchsuchen“
- Über die Toolbar ein neues CVS Repository hinzufügen
- In dem folgenden Dialog die Felder wie folgt befüllen (s. n. Folie)
  - Host: cvs.qv.de
  - Repository-Pfad: /var/cvs/produktdata
  - Benutzer: der eigene Benutzername
  - Passwort: das eigene Passwort
  - Verbindungstyp: pserver (ist der Default)
  - Standortport verwenden: Ja

## Einrichten des Zugriffs auf das Repository (2)



**CVS-Repository hinzufügen**

Neues CVS-Repository hinzufügen  
Neues CVS-Repository zur Sicht 'CVS-Repositories' hinzufügen

**Position**

Host: cvs.qv.de

Repository-Pfad: /var/cvs/produktdata

**Authentifizierung**

Benutzer:

Kennwort:

**Verbindung**

Verbindungstyp: pserver

☒ Standardport verwenden

☐ Folgenden Port verwenden:

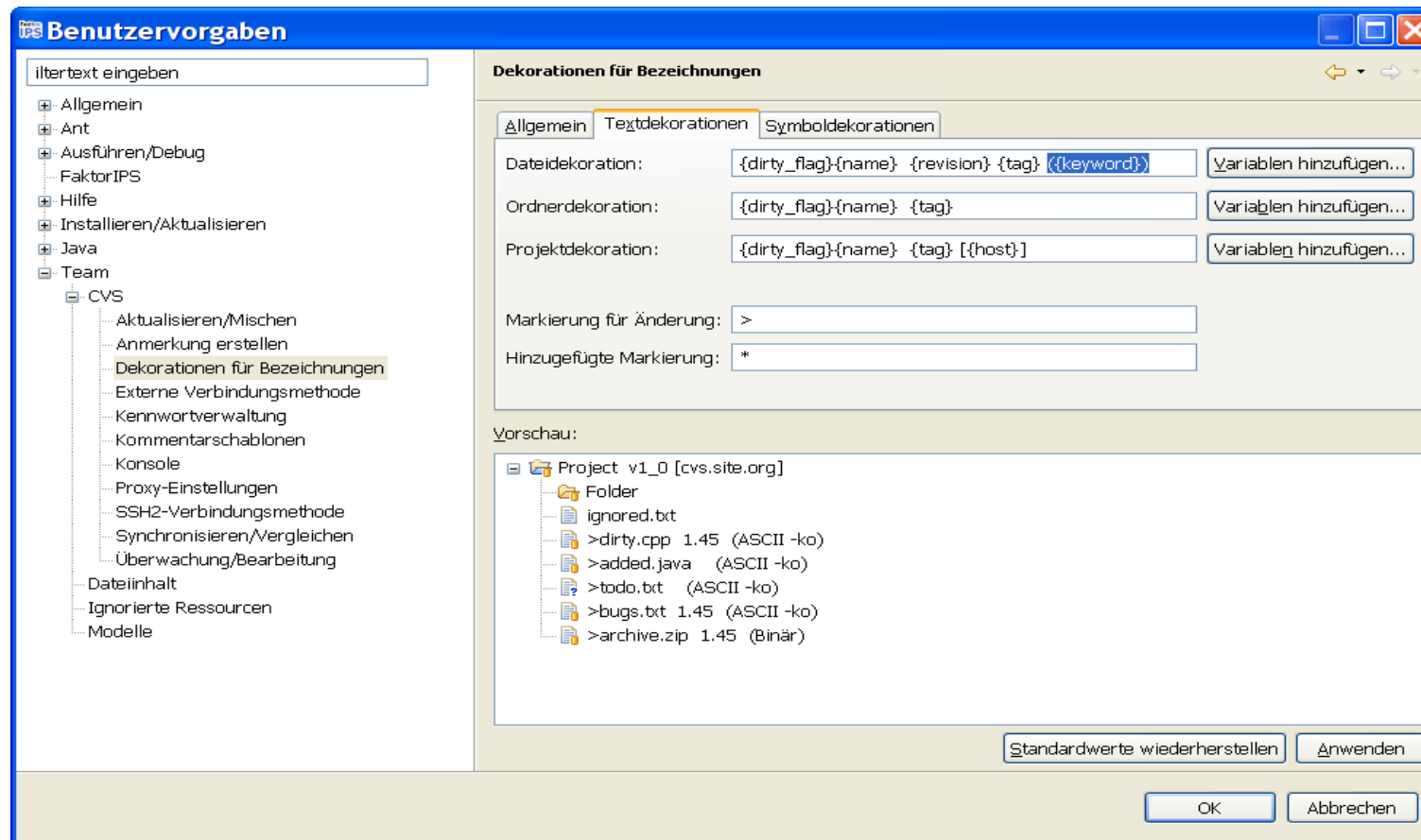
☒ Verbindung bei Fertigstellung prüfen

☐ Kennwort speichern

⚠ Gespeicherte Kennwörter werden auf dem Computer in einer Datei gespeichert. Das Lesen dieser Datei ist für einen Eindringling zwar schwierig, aber nicht unmöglich.

Fertig stellen Abbrechen

# Einstellungen für Anzeige des Status im Explorer

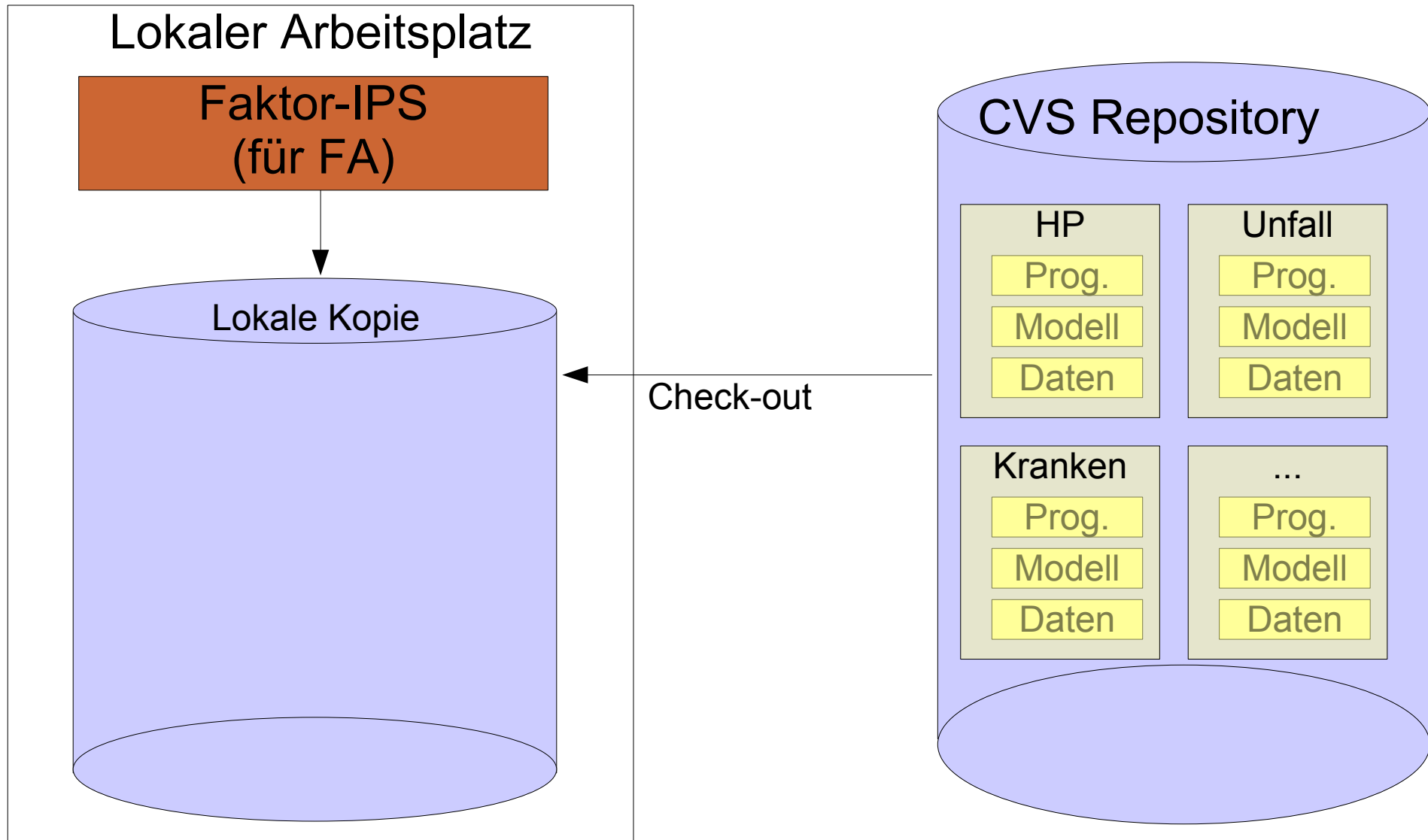


Aufruf der Benutzervorgaben über das Menü „Fenster->Benutzervorgaben“.  
Wählen des Abschnitts „Team->CVS->Dekorationen für Bezeichnungen“  
Entfernen des blau markierten Bereichs ({keyword}) deaktiviert die Ausgabe von  
ASCII / Binary am Ende der Dateibezeichnung

## Inhalt

- Grundlagen
- Bearbeiten von Produktdaten im Team
  - Einrichten des Zugriffs auf das Repository
  - Grundfunktionen zum Arbeiten im Team
  - Lösung von „Konflikten“
- Übergabeprozess

## Check-out von Projekten (1)

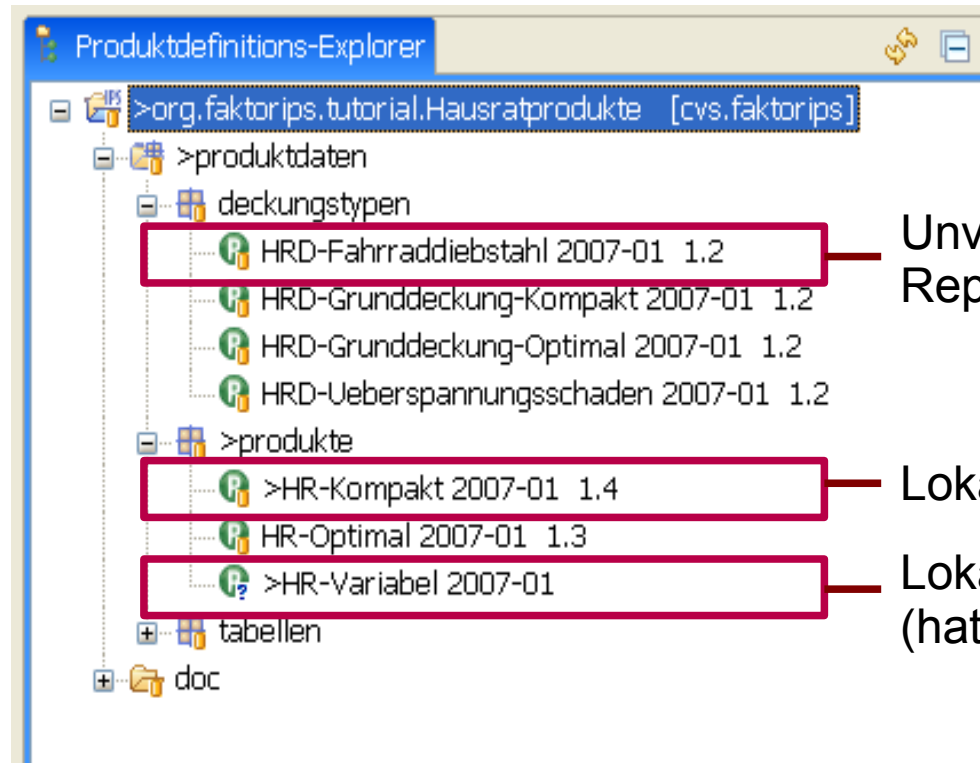


## Check-out von Projekten (2)

- Mit Check-out wird ein Projekt initial aus dem Repository in den lokalen Arbeitsbereich kopiert.
- Check-out wird in den folgenden Situationen benötigt:
  - Erstes Arbeiten mit Faktor-IPS bzw. den Produktdaten einer Sparte
  - Nach Löschen eines Projektes aus dem lokalen Arbeitsbereich
- Erfolgt in der Perspektive „CVS Repository durchsuchen“
  - Wechseln in diese Perspektive im Menü über „Fenster->CVS Repository durchsuchen“



## Darstellung von Änderungen im Explorer

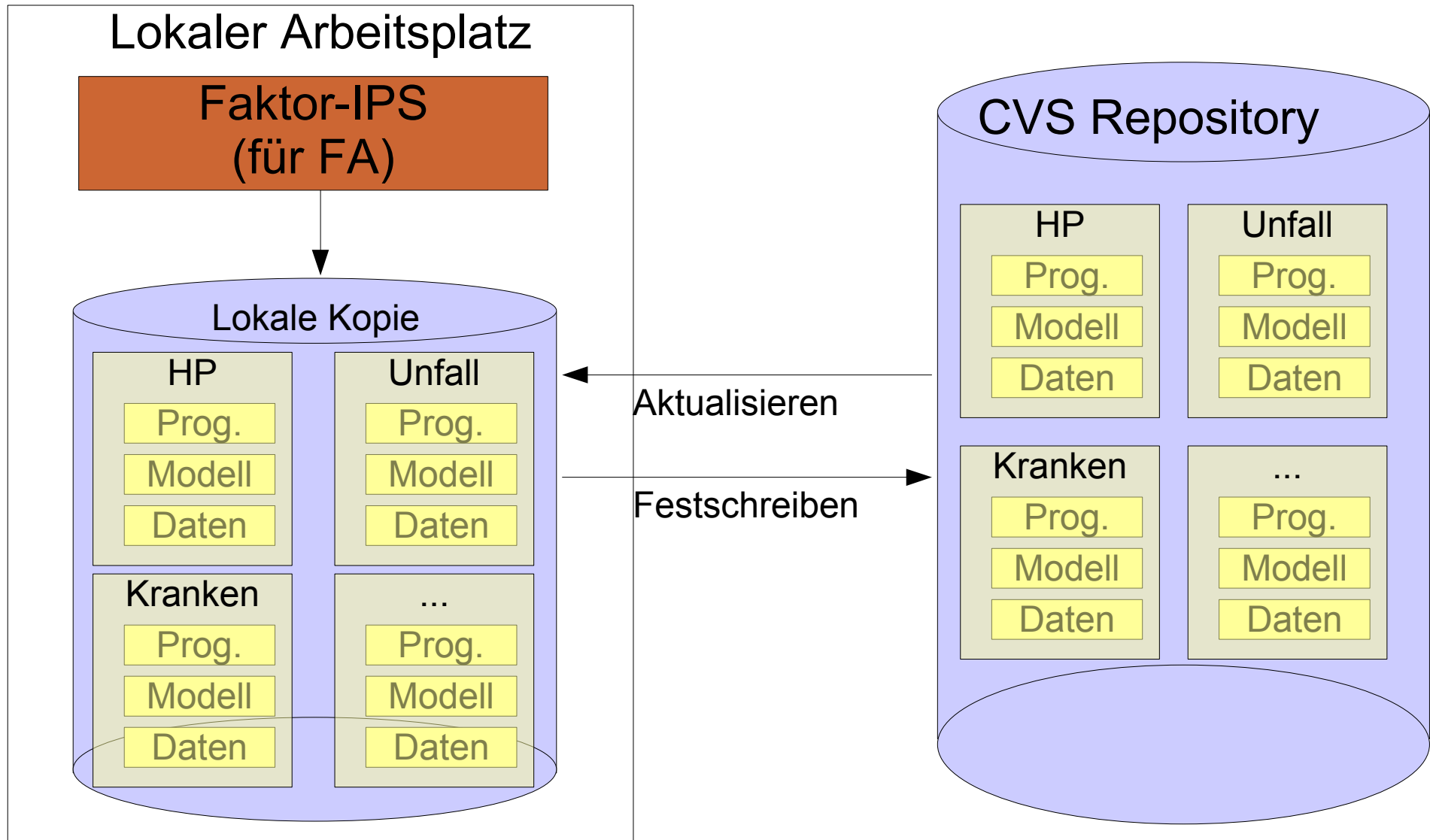


Unveränderte Dateirevision 1.2 aus dem Repository

Lokale geänderte Dateirevision 1.4

Lokal neu erzeugte Datei  
(hat noch keine Revisionsnummer)

## Der Aktualisieren-Festschreiben-Zyklus (1)



## Der Aktualisieren-Festschreiben-Zyklus (2)

- Aktualisieren (Update)
  - Mit Aktualisieren werden die aktuellen Daten aus dem Repository auf den lokalen Arbeitsplatz geholt.
  - Bereits gemachte Änderungen auf dem lokalen Arbeitsplatz bleiben erhalten.
- Festschreiben (Commit)
  - Die lokalen Änderungen werden in das Repository geschrieben.
  - Für geänderte Dateien wird im Repository eine neue Überarbeitung (Revision ) angelegt (mit einer höheren Revisionsnummer).
- Beide Befehle beziehen sich immer auf ein gesamtes Projekt.

## Wiederherstellen aus dem Repository

- Mit Wiederherstellen können lokal gelöschte Dateien (Produktbausteine, Tabellen, Testfälle) wieder aus dem Repository auf den lokalen Arbeitsplatz kopiert werden.
- Der Befehl kann auf das gesamte Projekt oder einzelne Pakete ausgeführt werden.

## Änderungsprotokoll anzeigen

- Mit „Protokoll/Historie anzeigen“ kann die Änderungshistorie einer Datei angezeigt werden.
- Zwei Arten von Historien (Überarbeitungen)
  - Im Repository gespeicherte Historien
  - Im lokalen Arbeitsplatz gespeicherte Historien
- Mögliche Aktionen auf einer Historie
  - Öffnen
    - *Die Historie zur Ansicht öffnen*
  - Inhalt abrufen
    - *Den aktuellen Stand durch die Historie ersetzen*
  - Aktuelles Element mit lokalem Vergleichen
    - *Den historischen Stand mit dem aktuellen Stand vergleichen*
  - Sticky Überarbeitung abrufen
    - *Für Fortgeschritte. Nicht ohne genaues Verständnis benutzen.*

## Synchronisieren

- Anzeigen der Differenzen zwischen dem Repository und dem lokalen Arbeitsbereich

## Vergleichen mit ...

- letzte aus HEAD
  - Vergleicht den lokalen Stand mit dem aktuellen Stand im Repository
- andere Verzweigung oder Version
  - für Fortgeschrittene
- Einander
  - verfügbar, wenn zwei Bausteine markiert sind.
- Protokoll
  - Vergleicht den lokalen Stand mit einem Stand aus der Änderungshistorie
- Lokales Protokoll
  - nicht verwenden: statt dessen Protokoll verwenden

## Ersetzen durch ...

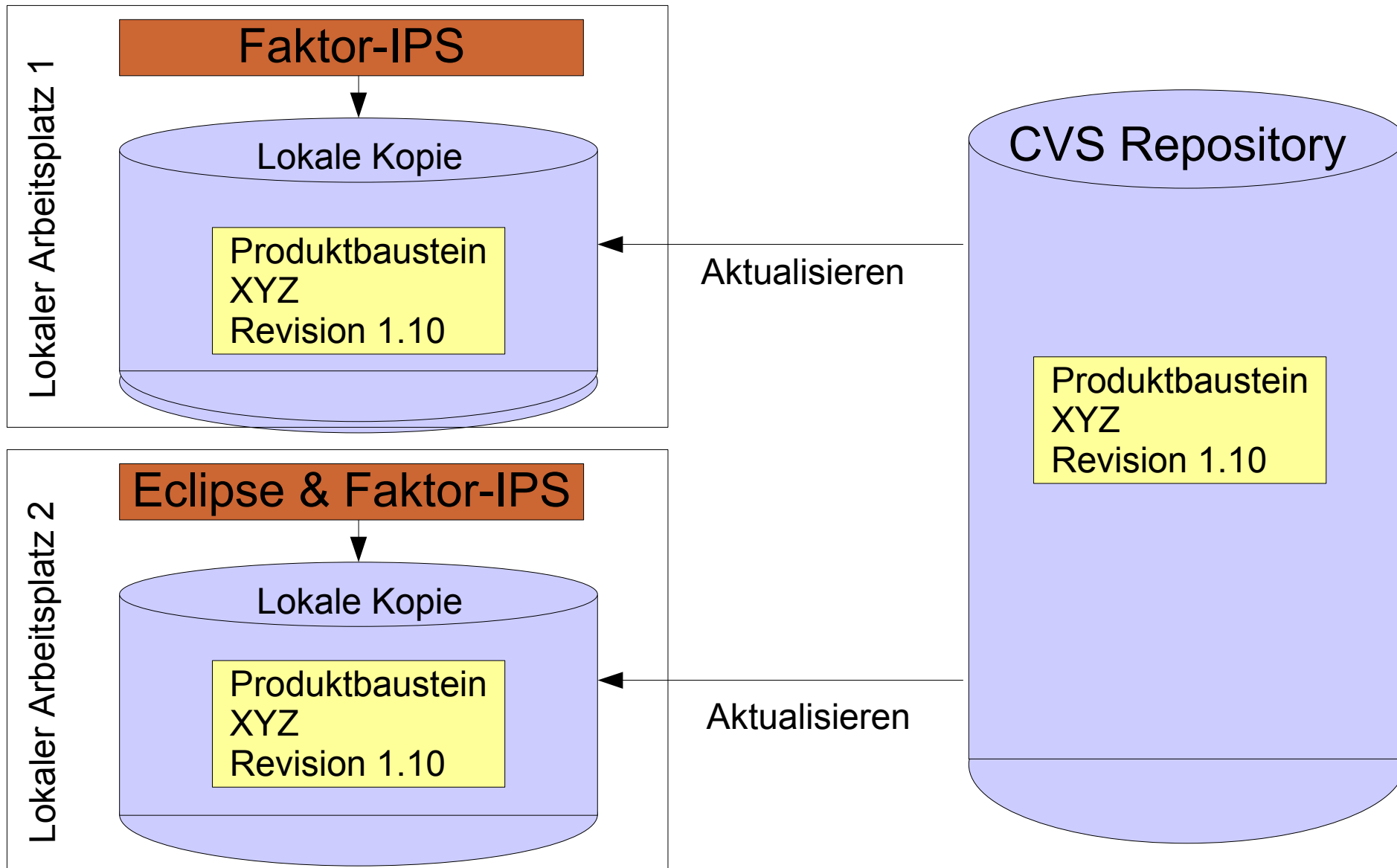
- letzte aus Repository (HEAD)
  - Ersetzt den lokalen Stand mit dem aktuellen aus dem Repository
- andere Verzweigung oder Version
  - für Fortgeschrittene
- Protokoll
  - Ersetzt den lokalen Stand mit einem Stand aus der Änderungshistorie
- Vorheriges Element aus lokalem Protokoll
  - Ersetzt den lokalen Stand mit dem vorherigem lokalem Stand (vor dem letzten Speichern)
- Lokales Protokoll
  - nicht verwenden: statt dessen Protokoll verwenden



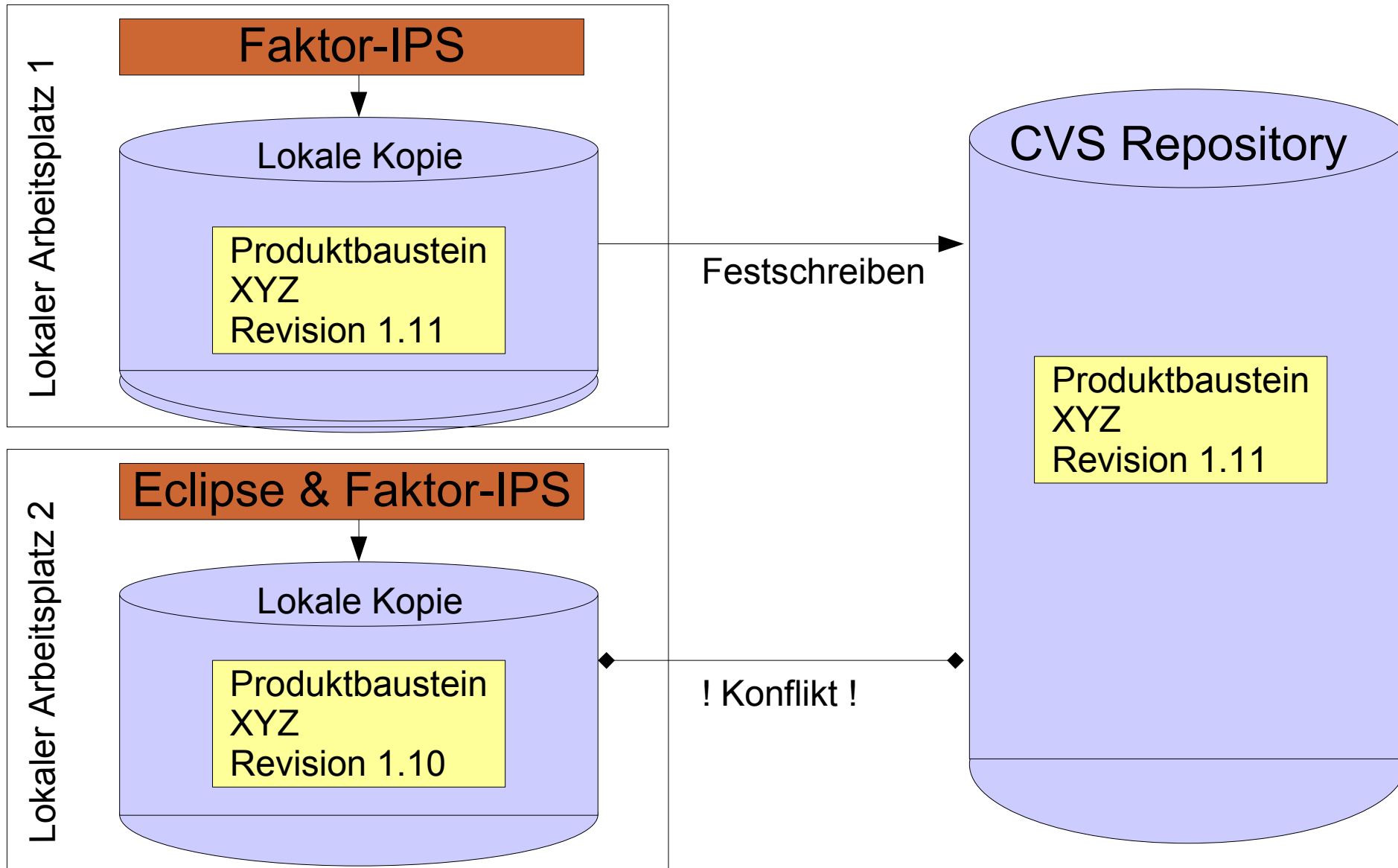
## Inhalt

- Grundlagen
- Bearbeiten von Produktdaten im Team
  - Einrichten des Zugriffs auf das Repository
  - Grundfunktionen zum Arbeiten im Team
  - Lösung von „Konflikten“
- Übergabeprozess

## Entstehung von Konflikten (1)



## Entstehung von Konflikten (2)



## Konfliktproblem

- Zwei Anwender haben die gleiche Revision eines Produktbausteins aus dem Repository geholt.
- Anwender 1 hat den Baustein geändert und wieder ins Repository festgeschrieben.
- Anwender 2 hat den Baustein ebenfalls geändert.
- Beim Festschreiben ins Repository bekommt er einen Konflikt gemeldet, da die Änderungen von Anwender 1 nicht einfach überschrieben werden dürfen.
- Anwender 2 muss den „Konflikt“ lösen

## Vorgehen bei einem Konflikt

- Klären: Welche Daten soll der Produktbaustein enthalten?
- 3 Fälle
  - Der Zustand im Repository ist bereits korrekt. Die lokalen Änderungen können verworfen werden.
  - Der Zustand im lokalen Arbeitsbereich ist korrekt, der Stand im Repository kann überschrieben werden.
  - Der Sollzustand ergibt sich aus den eigenen lokalen Änderungen und den Zustand im Repository.

## Fall 1: Der Zustand im Repository ist korrekt.

- Die lokalen Änderungen werden mit der Aktion „Ersetzen durch“ verworfen.
- Achtung: Funktion „Aktualisieren“ funktioniert nicht, da hiermit nur konfliktfreie Daten aus dem Repository geholt werden!

## Fall 2: Der Zustand im lokalen Arbeitsbereich ist korrekt

- Das gesamte Projekt mit dem Repository synchronisieren
  - Die Datei bei der ein Konflikt besteht als gemischt (zusammengeführt) markieren.
  - Danach kann die Datei ganz normal festgeschrieben werden.

## Fall 3: Sollzustand ergibt sich aus den eigenen lokalen Änderungen und den Zustand im Repository.

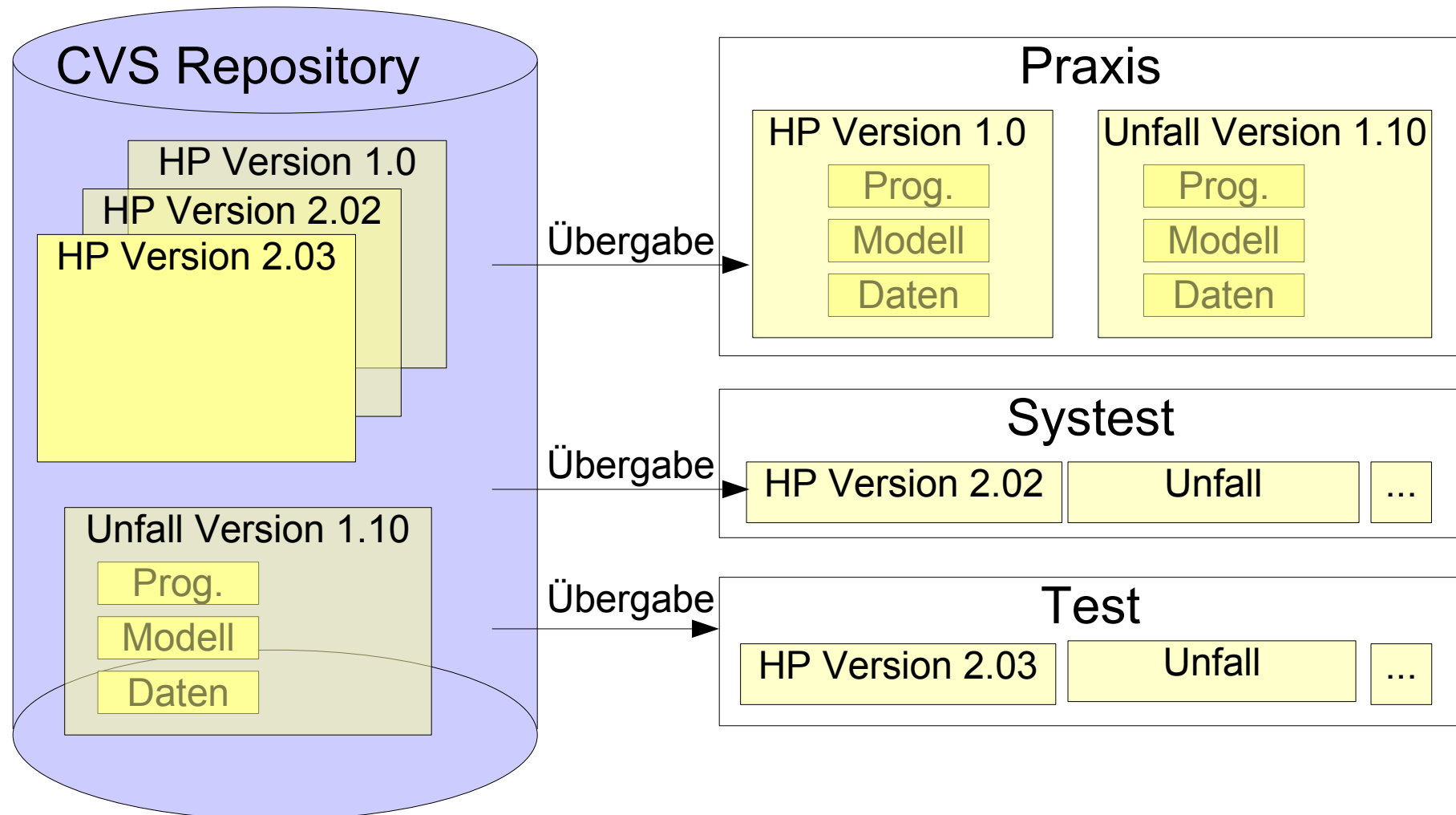
- Am einfachsten:
  - Lokale Änderungen verwerfen, indem man sich mit der Aktion „Ersetzen durch“ den Stand aus dem Repository holt.
  - Änderungen erneut durchführen
  - Änderungen festschreiben
- Alternativ
  - Änderungen des anderen Benutzers in der eigenen lokalen Kopie nachziehen.
  - Weiter wie in Fall 2, also
    - *Projekt synchronisieren*
    - *Datei als gemischt / zusammengeführt markieren*
    - *Änderungen festschreiben.*



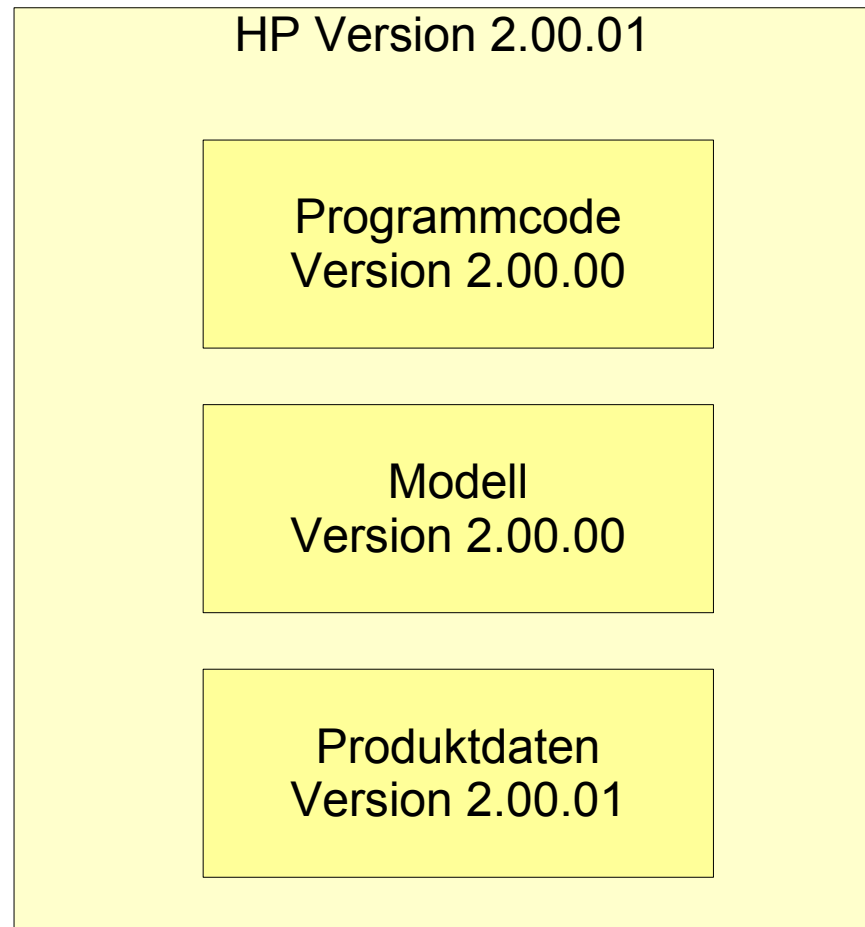
## Inhalt

- Grundlagen
- Bearbeiten von Produktdaten im Team
  - Grundfunktionen zum Arbeiten im Team
  - Lösung von „Konflikten“
- Übergabeprozess

# Übergeben wird immer eine Version einer Sparte



## Beispiel: Version von Haftpflicht



## Beispiel: Version der Produktdaten von Haftpflicht

HP Produktdaten 2.00.01	
KqvHpPolart	2003-11 1.14
KqvHpPolart	2002-09 1.10
Tariftabelle	2003-11 1.20
Tariftabelle	2002-09 1.20
...	

## Zusammenhang Revision / Version

HP Produktdaten 2.00.01	HP Produktdaten 2.00.02
KqvHpPolart 2003-11 1.14	KqvHpPolart 2003-11 1.14
KqvHpPolart 2002-09 1.10	KqvHpPolart 2002-09 1.10
<b>Tariftabelle 2003-11 1.20</b>	<b>Tariftabelle 2003-11 1.23</b>
Tariftabelle 2002-09 1.20	Tariftabelle 2002-09 1.20
...	...

Für die Tarifgeneration 2003-11 wurde im Zuge einer Beitragsanpassung die Tariftabelle geändert.

=> Die neue Version der Produktdaten enthält diese neue Revision der Tariftabelle und alle unveränderten Daten.

=> Eine Revision kann zu mehreren Versionen gehören.

## Übergabeprozess

- ITK erstellt eine neue Version der Produktdaten.
- ITK benachrichtigt den für die Sparte verantwortlichen PK-Verantwortlichen in der AE.
- Der PK Verantwortliche erstellt eine neue Version der Produktkomponente (für die Sparte).
- Der PK Verantwortliche deployed die PK in die Zielumgebung(en)
  - nach Test kann der PK Verantwortliche direkt deployen
  - für eine Übergabe nach Systest und Praxis stellt er einen Gabun-Auftrag.

## Zielumgebungen

### Praxis

Host

AppServer  
Intranet

AppServer  
Internet

Quisy  
Installationen

### Systest

Host

AppServer  
Intranet

AppServer  
Internet

Quisy  
Installationen

### Test

Host

AppServer  
Intranet

AppServer  
Internet

Quisy  
Installationen