CVS-Kurzübersicht

Arbeiten mit CVS

Erzeugen eines Repository	**Ermitteln von Änderungen**
`mkdir` *verzeichnis* `export CVSROOT=`*verzeichnis* `cvs init`	`cvs update` *datei* **Übertragen von Änderungen** `cvs commit` *datei*
Festlegen des Repository	**Erzeugen eines Releases**
export CVSROOT=*verzeichnis*	`cvs tag` *name* `[`*datei(en)*`]`
Importieren eines bestehenden Projektes in das Repository	**Erzeugen eines Zweiges**
`cvs import` *modulname hersteller* `start`	`cvs tag -b` *zweig*
Einspielen eines Projektes in die lokale Umgebung	**Wechseln in den Zweig**
`cvs checkout` *modulname*	`cvs update -r` *zweig*
Anmelden an das Repsoitory	**Zusammenführen eines Zweiges**
`cvs login`	`cvs update -j` *zweig*
Abmelden von dem Repository	**Löschen eines Projektes aus der lokalen Umgebung**
`cvs logout`	`cvs release` *verzeichnis*
Hinzufügen einer neuen Datei	**Vergleichen mit vorheriger Revision**
`cvs add` *datei*	`cvs diff [-j` *revision*`]` *datei*
Entfernen einer Datei	**Abfragen des Status**
`cvs remove` *datei*	`cvs status` *datei*

Überall dort, wo von Datei gesprochen wird, kann selbstverständlich auch ein Verzeichnis sein.

Administrative Befehle

Befehle für Netzwerkzugriff

Optionen

Vorgabe von Kommentaren	**Bestimmung des Repository**
-m „*Kommentar*“	-d *verzeichnis*

Zweige in der PG 369

Zyklus 1			**Zyklus 2**		
Design	Implementierung	Test	Re-Design	Implementierung	Test

Die Phasen Design (Zyklus 1), Test (Zyklus 1), Re-Design (Zyklus 2) und Test (Zyklus 2) verlaufen in der Großgruppe zusammen. Implementierung (Zyklus 1) und Implementierung (Zyklus 2) stellen für die einzelnen Komponenten Zweige dar.
Die Zweige und deren Bezeichnung werden zentral vorgegeben. Ebenso, wie zu gegebenen Zeitpunkten in die Zweige bzw. aus den Zweigen heraus zurück in den Hauptzweig gewechselt wird.

Sonstiges

Compilierfähigkeit

Nachdem ein `commit` ausgeführt wurde, ist das Repository auf Compilierfähigkeit zu prüfen.

Reihenfolge von `update` und `commit`

Zuerst immer `update`, dann nötigenfalls Konflikte bereinigen und dann erst `commit` ausführen.

Konflikte

Konflikte erscheinen im Quelltext als:

```
<<<<<<< datei
        Quelltext aus dem Repositiory
=======
        lokaler Quelltext
>>>>>>> revision
```

Dabei können Dateien mit bestehenden Konflikten nicht eingespielt werden.