

Konfigurationsmanagement

Seminar Simulation und Bildanalyse mit Java SS04

Andreas Stieger

Motivation

Konfigurationsmanagement

- Zusammenarbeit mehrerer Entwickler
- komplexe Softwareprojekte
- Sourcen, Dokumente, Daten, Testfälle
- Versionierung, Archivierung, Abhängigkeiten
- Parallele Versionen
- Softwarepflege

4 Säulen des Konfigurationsmanagement

- Version-Management
- Build-Management
- Release-Management
- Change-Management

Fragen Konfigurationsmanagement

- Wer hat meine Datei verändert?
- Können wir das rückgängig machen?
- Warum sind meine Änderungen verschwunden?
- Durch welche Änderung lässt sich das nicht mehr compilieren?
- Wie bekommen wir das jetzt zu einem Paket zusammen?
- Welche Version haben wir verschickt?
- Wie und wo beheben wir den Fehler?

Versionsmanagement

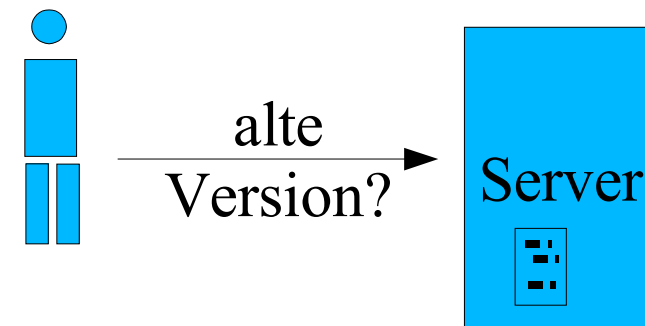
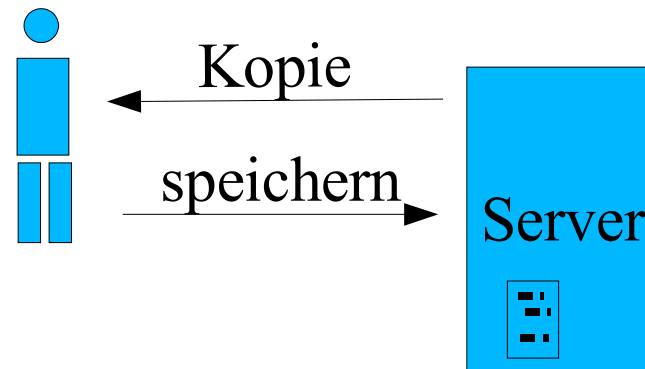
- verwaltet und lenkt Versionen
- von Programmen und Dokumenten
- Abbildung des Lebenszyklus durch Zustände
- Arbeitsbereiche für Entwickler
- lenkt gleichzeitige Bearbeitung durch mehrere Entwickler

Problemstellung

- Anton und Bernd arbeiten an einem Projekt
- Quellen auf Netzwerkshare
- Entwicklung mit lokaler Kopie
- nach Arbeitsende Kopie zum Projektverzeichnis

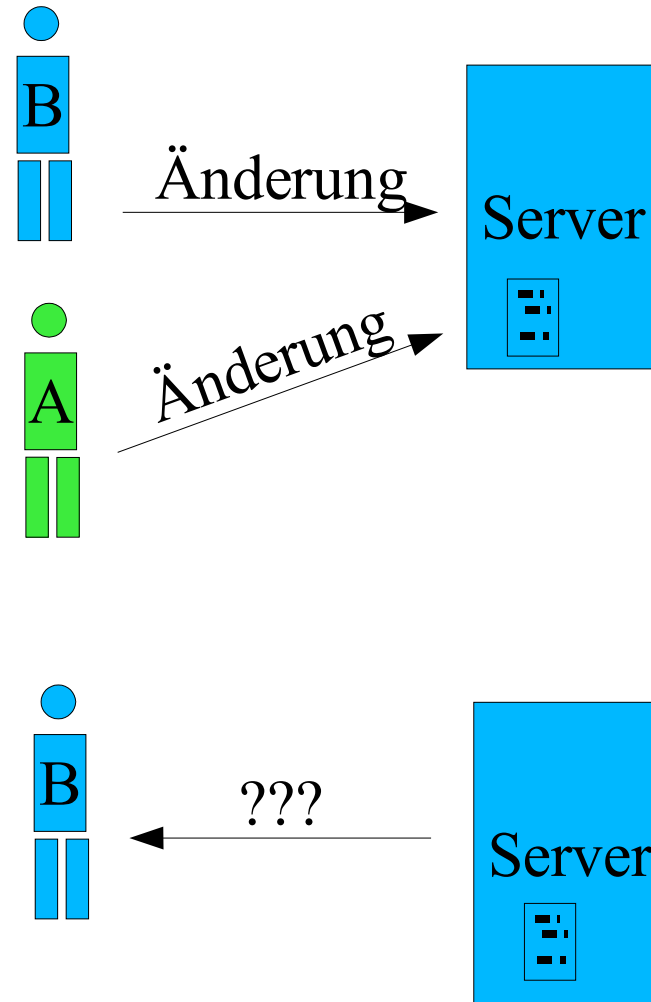
Problemstellung (2)

- Bernd ändert eine Quelle am Freitag Abend
- will sie Montag früh wieder zurücknehmen
- hat aber die ursprüngliche Quelle nicht mehr



Problemstellung (3)

- Bernd ändert 2. Quelle
- am Montag ist Änderung verschwunden
- Anton hat sie mit gleichzeitigen Änderungen überschrieben



Elemente im Versionsmanagement

- Element = identifizierbare Einheit, die im Projektverlauf entsteht
- Format:
 - Programmquellen: Text
 - Dokumente: Text oder strukturiert
- Programmquellen
- Dokumente
- Spezifikationen
- Konstruktion
- Testfallbeschreibungen
- Projektplanungen

Elemente im Versionmanagement

- Daten
- Testtreiber, -scripte, -daten, -ergebnisse
- Verzeichnisse
- Bilder und Multimediadateien
- generierte Elemente:
 - ausführbare Programme
 - Zwischenprodukte

Versionierung, Labels

- Überarbeitungsstand
 - identifizieren
 - wiederherstellen
- neue Version wenn Überarbeitungsstand festgehalten werden soll

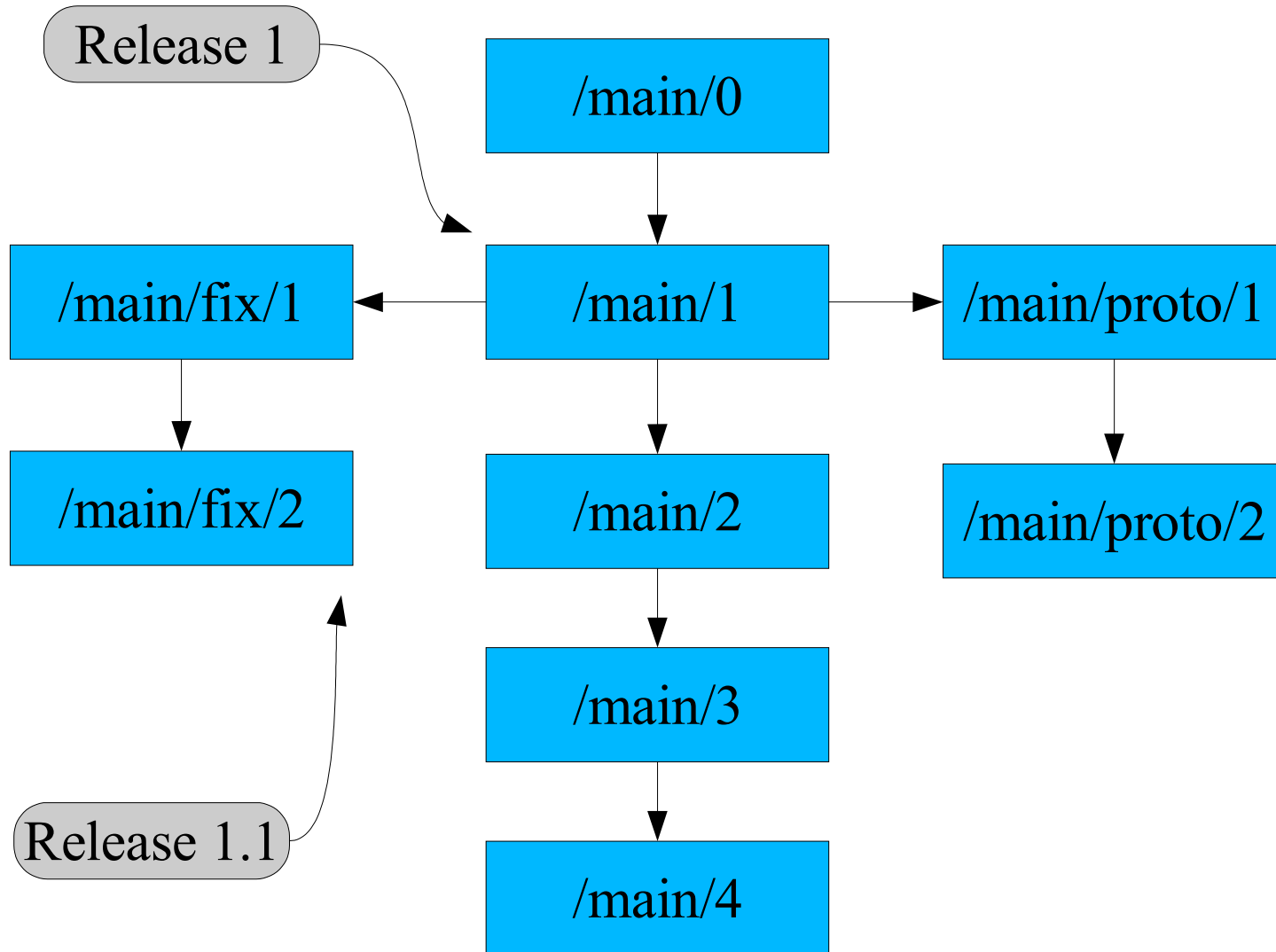
Basisoperationen

- check in
 - von lokaler, geänderter Kopie
 - keine nachträgliche Änderung (nur durch neue Version)
 - ggf. auch während Weiterbearbeitung
- check out
 - von Versionsverwaltung zu lokaler Kopie
 - ggf. mit Sperren verbunden

Versionsbenennung

- eindeutiger Name jeder Version („Revision“)
- trivial: Zeitstempel
 - nicht ausreichend für mehrere parallele Versionen
- Major- und Minor-Nummern:
 - 1.0, 1.1, 1.0.1.1 (rcs Konvention)
- Kombination fortlaufender Nummern und Namen
 - main: Hauptzweig, fix: Wartungszweig

Lebenszyklus Elemente



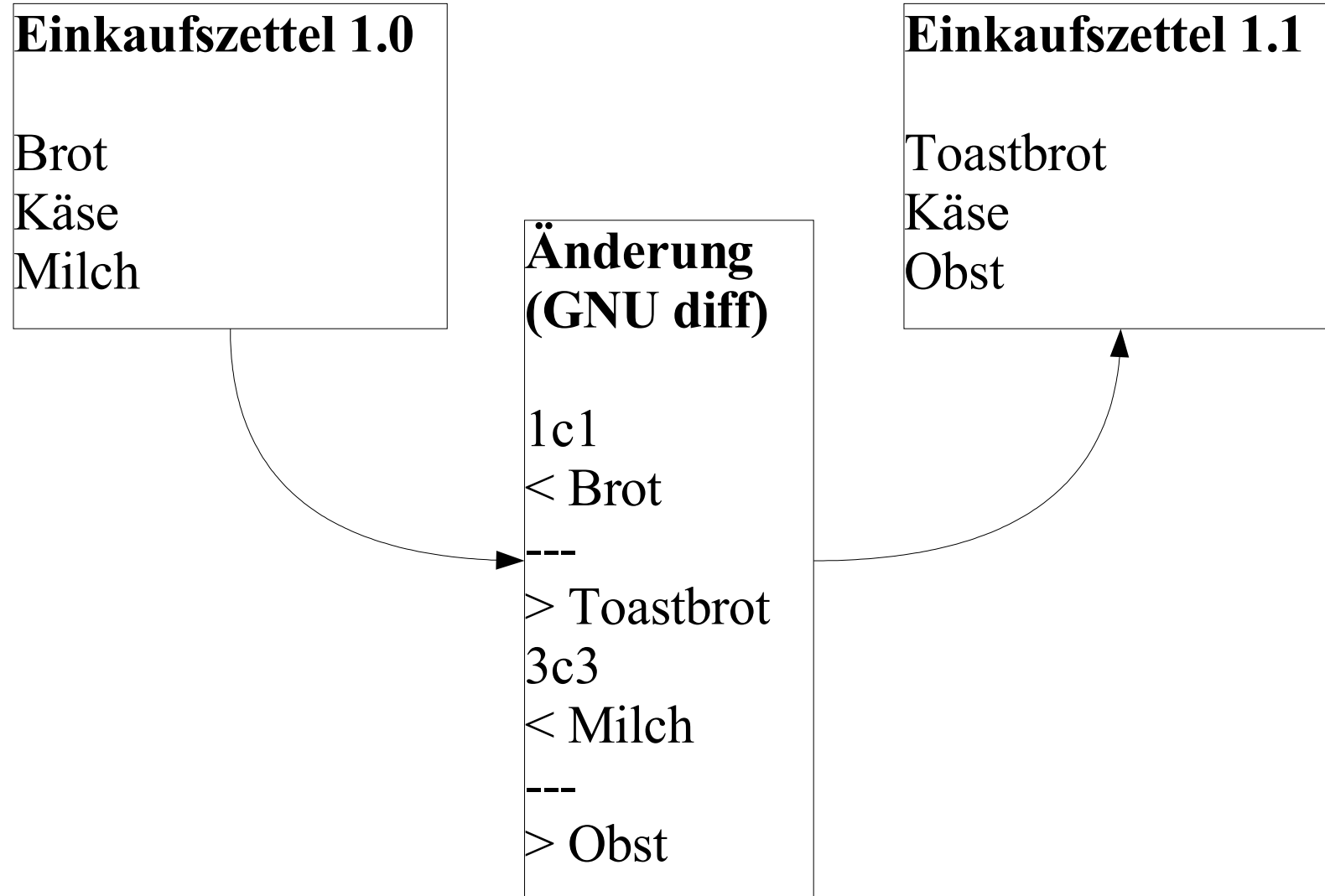
Eigenschaften von Elementen und Versionen

- Elementtyp
- Nutzerkennung des Entwicklers
- zugehörige Release, BugID oder Changerequest
- Datum, Grösse, letzte Änderung (ggf. schon im Dateisystem)
- Labels

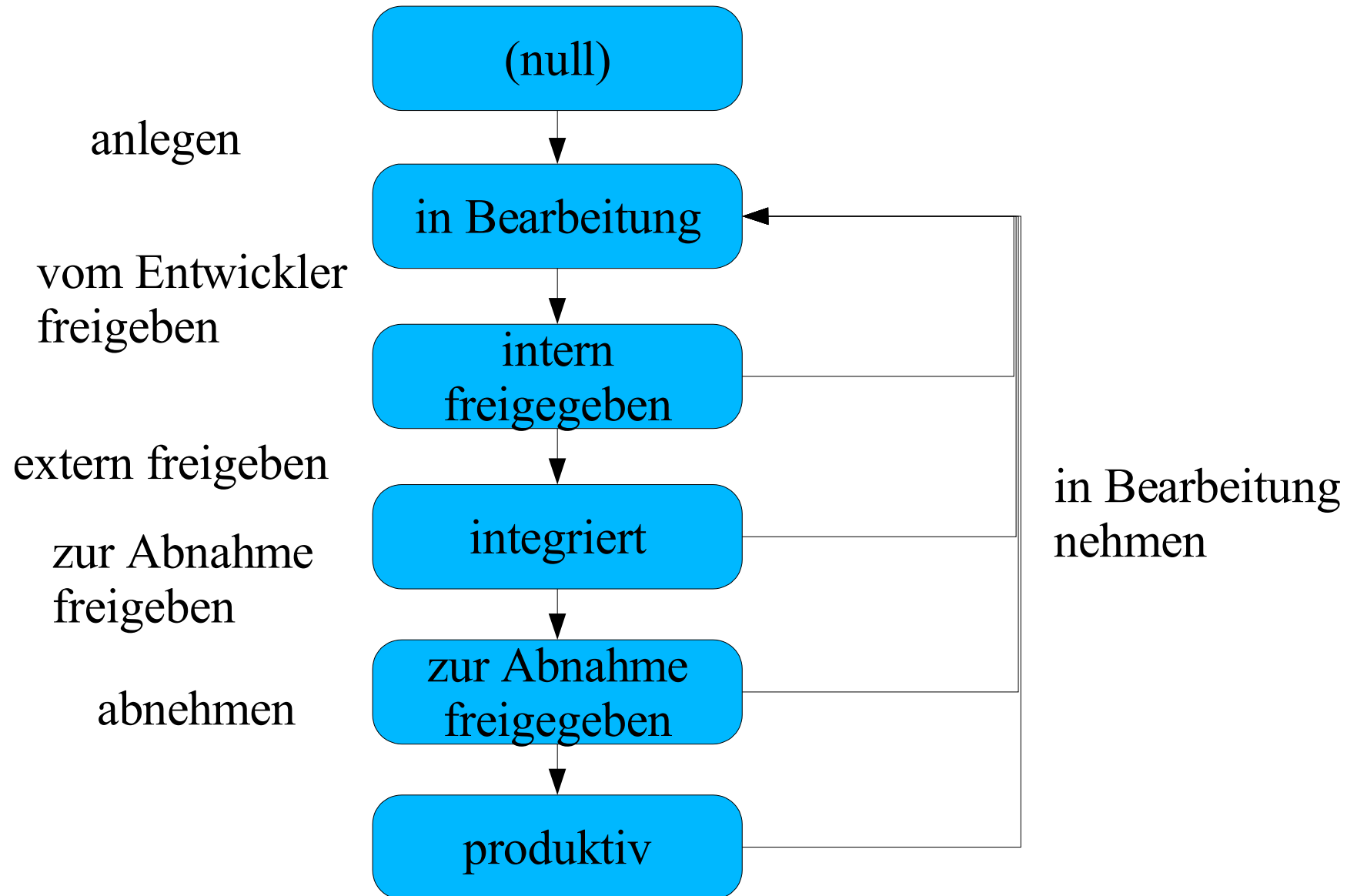
Speicherung von Dateien

- Text
 - volle Datei (schneller)
 - Differenzen (weniger Speicher)
- Binär
 - nur volle Datei

Beispiel Differenz



Zustandsgraph Lebenszyklus



Entwicklungsraum

- für jeden Entwickler
- keine Änderungen und Einsicht durch andere
- Modelling, Editierung (IDE)
- Übersetzung, Bindung
- Tests
- Freigabe
- wird anfangs durch einen checkout gefüllt

Sperren

- (Erinnerung) Wer hat meine Datei überschrieben?
- check out
 - Sperrung für andere Entwickler
 - nur lesen
- check in
 - Auflösung der Sperre, Änderungen sichtbar

Alternative zu Sperren

- kontrolliere parallele Bearbeitung aller Entwickler
- Zusammenführung durch Merge
- keine Verzweigung (Branch), da alle Versionen auf gleichem Entwicklungszweig
- Voraussetzung: Werkzeug erkennt Änderungen (wie z.B. diff)
- umgesetzt in cvs

CVS

- (anstelle der Demonstration beim Vortrag hier ein kleiner Abriss der grundlegenden Operationen)
- Repository-Verzeichnis erstellen:
 - ```
export CVSROOT=/var/log/CVS
mkdir -p /var/log/CVS/CVSROOT
mkdir -p /var/log/CVS/meinprojekt
cd /var/log/CVS
```
- Lese- Schreibrechte setzen, am besten eine Usergruppe „cvs“ anlegen und Rechte 775 setzen.
  - ```
chown cvs:cvs meinprojekt  
chmod 775 meinprojekt
```
- als User der Gruppe cvs kann man jetzt Sourcen importieren:
 - ```
cd ~/meinprojekt
cvs import -m „initial import“ meinprojekt Original alpha
```
- jeder User der Gruppe kann dann das Projekt auschecken
  - ```
cd ~  
cvs checkout meinprojekt
```

CVS

- Wir nehmen an, dass CVS eingerichtet ist und das Projekt ausgecheckt ist. Vor dem Beginn der Arbeit sollte man die lokalen Quellen aktualisieren, sich ggf. die Änderungen ansehen und testen, ob Funktionalität im eigenen Zuständigkeitsbereich betroffen ist.
 - `cd ~/meinprojekt`
`cvs update`
- Ein „?“ bedeutet hier, dass die Datei lokal vorhanden, aber nicht im Repository ist. Diese kann mit `cvs add` hinzugefügt werden. Ein „U“ bedeutet die Änderung durch einen anderen Entwickler.
 - `? file1.c`
`U file2.c`
- Nach den Änderungen sollte man erneut ein `cvs update` ausführen, und wieder testen. Beim Einchecken mit `cvs commit` wird man nach einen Kommentar fuer die Änderungen gefragt. Dieser sollte die Änderung vollstaendig, wenn auch nicht im Detail, beschreiben, z.B.: „fixed exception in calculate() for x=0.“
 - `cvs update`
`cvs commit`
- Weitere wichtige Befehle:
 - `cvs add|delete` // hinzufügen, entfernen von Dateien
`cvs log|diff datei.c` // Anzeigen von Logs oder Differenzen



General

Version Control

Repository
First Project
sPac
TCSstresser
Team Coherence
GP-Version 7
Addins
Builder
BuildObjects
Common
UI
Components
Forms
Images
Project
Units
SampleDlgAddin
SDF
Tracker
Build
Client
Common
Comms DLL
Help
Images
Indy
Interfaces
Server
UI
Components
Forms
Project
Test Harness
Units
Web Site
Test ProjectGroup
Test SCC
Locked Files
Recycle Bin

Name	Size	Rev	Type	Modified	Status	Lock
BaseDlg.pas	3 KB	1	Delphi Unit...	07/01/2001 17:43:40		Ewan
frmFileMasks.pas	11 KB	4	Delphi Unit...	23/10/2002 11:37:06		Ewan
frmGPVAbout.pas	5 KB	5	Delphi Unit...	21/04/2002 23:45:04		Ewan
frmGPVAssignVersion.pas	8 KB	3	Delphi Unit...	21/06/2002 16:49:28		Ewan
frmGPVBrowseFolder.pas	25 KB	2	Delphi Unit...	24/01/2002 15:49:24		Ewan
frmGPVChangePassword...	8 KB	2	Delphi Unit...	23/01/2002 23:01:36		Ewan
frmGPVCheckIn.pas	43 KB	13	Delphi Unit...	18/10/2002 00:37:44		Ewan
frmGPVCheckOut.pas	34 KB	7	Delphi Unit...	20/10/2002 22:06:40		Ewan
frmGPVComments.pas	10 KB	4	Delphi Unit...	01/10/2002 00:03:30		Ewan
frmGPVDiffs.pas	5 KB	1	Delphi Unit...	23/10/2002 13:06:32	Modified	Ewan
frmGPVFileGroupProps.pas	8 KB	3	Delphi Unit...	21/06/2002 16:45:26		Ewan
frmGPVFileGroups.pas	13 KB	4	Delphi Unit...	21/06/2002 16:45:32		Ewan
frmGPVFileProps.pas	60 KB	9	Delphi Unit...	18/10/2002 00:37:32		Ewan
frmGPVFindFiles.pas	59 KB	9	Delphi Unit	23/10/2002 12:16:02	Modified	Ewan
frmGPVFolderProps.pas	38 KB	8	Delphi Unit...	23/10/2002 00:35:22		Ewan
frmGPVGroupProps.pas	11 KB	3	Delphi Unit...	21/06/2002 16:46:08		Ewan
frmGPVHeader.pas	4 KB	3	Delphi Unit...	21/06/2002 16:49:26		Ewan
frmGPVHistory.pas	18 KB	3	Delphi Unit...	24/07/2002 12:17:18		Ewan
frmGPVLicenceGen.pas	22 KB	7	Delphi Unit...	16/04/2002 22:18:58		Ewan
frmGPVLogin.pas	8 KB	6	Delphi Unit...	28/02/2002 13:34:08		Ewan
frmGPVMain.pas	232 KB	33	Delphi Unit	22/10/2002 22:31:18	Modified	Ewan
frmGPVMixedProps.pas	9 KB	3	Delphi Unit...	21/06/2002 16:49:26		Ewan
frmGPVModPromotion.pas	13 KB	4	Delphi Unit...	21/06/2002 16:46:38		Ewan
frmGPVModVersion.pas	10 KB	4	Delphi Unit...	21/06/2002 16:46:44		Ewan
frmGPVModView.pas	18 KB	7	Delphi Unit...	24/06/2002 15:01:38		Ewan
frmGPVNewFolder.pas	5 KB	2	Delphi Unit...	18/10/2002 00:45:52	Modified	Ewan
frmGPVProgress.pas	7 KB	7	Delphi Unit...	17/07/2002 15:57:30		Ewan

Name	Comments	Timestamp	Author	LockedBy
1.8	Beta release 3	02/10/2002 0...	Ewan	Ewan
1.7	Initial release of the 7.1 Beta	25/07/2002 0...	Ewan	
1.6	New Release. Checkin before holiday	03/05/2002 1...	Ewan	
1.5	New release. Checking in prior to mov...	18/04/2002 0...	Ewan	
1.4	Release 7.0.2.0	11/02/2002 0...	Ewan	
1.3	Just in case...	16/12/2001 2...	Ewan	
1.2	Just in case - 19-Sept-2001	19/09/2001 2...	Ewan	
1.1	Just in Case	23/08/2001 0...	Ewan	
1.0	Initial Revision	17/07/2001 0...	Supervisor	

Revisions Notes Tracker

Build Management

- Quelle --> lauffähige Software
- trivial: Übersetzung aller Quellen (Zeit)
- besser: nur tatsächlich geänderte Teile übersetzen

- notwendig: Beschreibung der Abhängigkeiten
- Beispiele:
 - make (Unices, Windows)
 - ant (Java)

Abhängigkeiten

- Wen benutze ich?
 - Aufrufe
 - includes
 - Vererbung
 - ...
- Wer benutzt mich?
 - entscheidend für Buildprozess
 - schwieriger zu finden

Integrationsraum

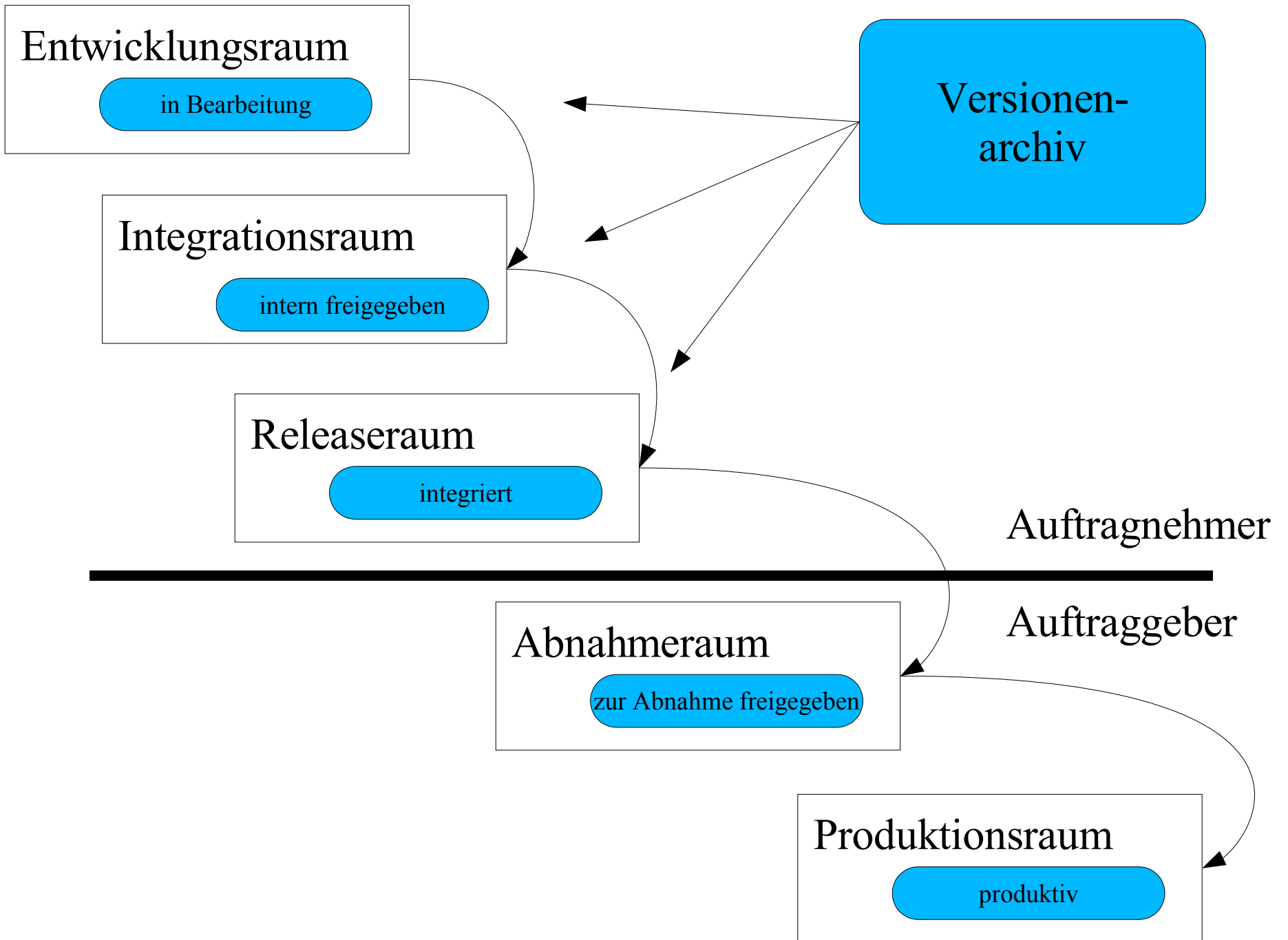
- Übersetzung des modulgetesteten Systems
 - (smoke test)
- beinhaltet alle intern freigegebenen Quellen
- Build erzeugt lauffähige Programme

Release Management

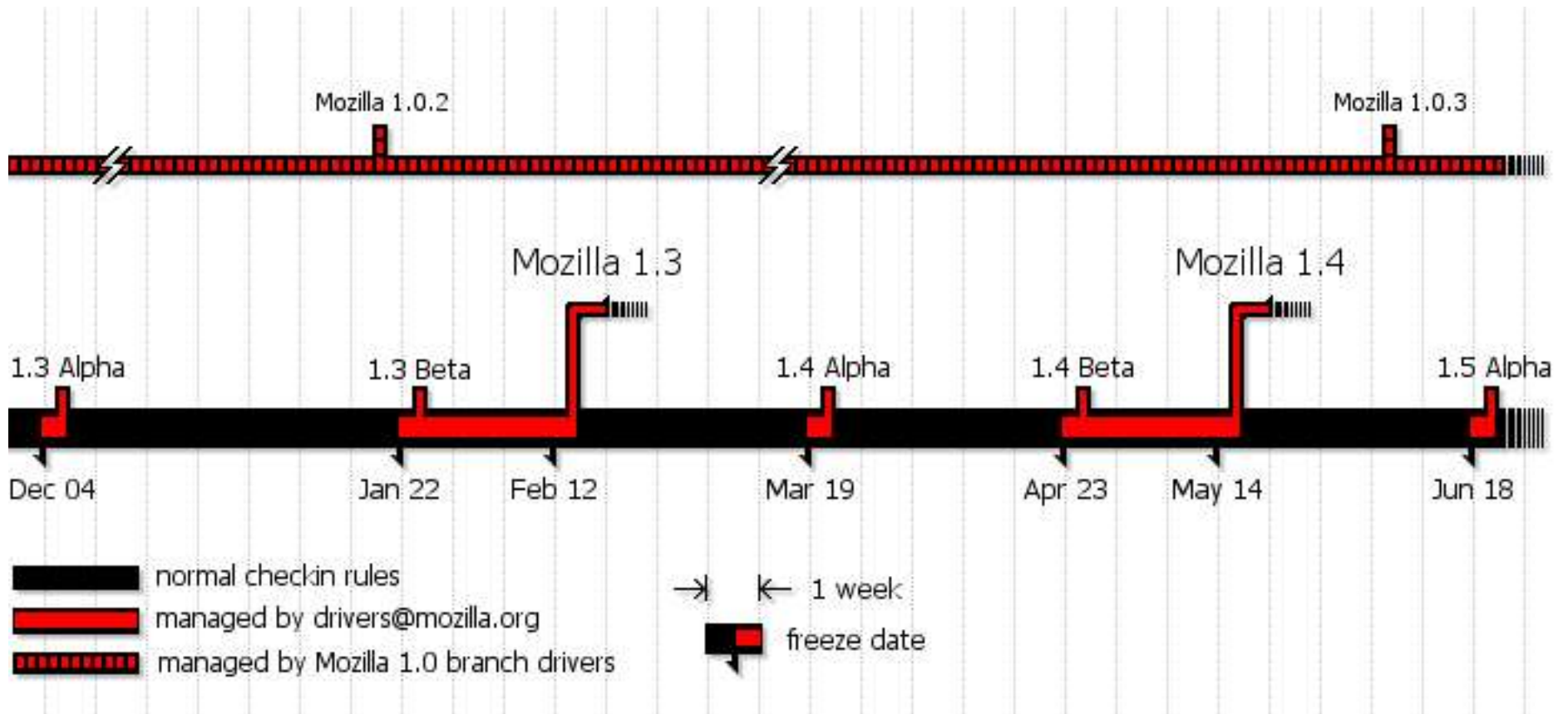
- "Release ist eine Beta, die aus dem Labor entkommen ist."
- oder: Jedes Softwarepaket, das an den Auftraggeber geliefert wird, ist ein Release
- Benennung und Archivierung notwendig für
Wartung

Raumkonzept

- Abnahmeraum
 - Abnahmetest durch Auftraggeber
- Produktionsraum
 - Software produktiv, bzw. Projekt „live“
 - beinhaltet nur noch lauffähige Programme



Branching: Mozilla Projekt



The Linux Kernel Archives - Mozilla

File Edit View Go Bookmarks Tools Window Help

http://www.kernel.org/ Search

Home Bookmarks SUSE mozilla.org Latest Builds Forum

The Linux Kernel Archives

Welcome to the Linux Kernel Archives. This is the primary site for the Linux kernel source, but it has much more than just kernels.

Protocol	Location
HTTP	http://www.kernel.org/pub/
FTP	ftp://ftp.kernel.org/pub/
RSYNC	rsync://rsync.kernel.org/pub/

The latest stable version of the Linux kernel is: [2.6.5](#) 2004-04-04 04:17 UTC [F](#) [V](#) [VI](#) [C](#) [Changelog](#)

The latest [prepatch](#) for the stable Linux kernel tree is: [2.6.6-rc3](#) 2004-04-28 02:22 UTC [V](#) [VI](#) [C](#) [Changelog](#)

The latest [snapshot](#) for the stable Linux kernel tree is: [2.6.6-rc3-bk8](#) 2004-05-06 11:50 UTC [V](#)

The latest 2.4 version of the Linux kernel is: [2.4.26](#) 2004-04-14 13:14 UTC [F](#) [V](#) [VI](#) [C](#) [Changelog](#)

The latest [prepatch](#) for the 2.4 Linux kernel tree is: [2.4.27-pre2](#) 2004-05-03 23:08 UTC [V](#) [VI](#) [C](#) [Changelog](#)

The latest [snapshot](#) for the 2.4 Linux kernel tree is: [2.4.26-bk1](#) 2004-04-16 16:41 UTC [V](#) [Changelog](#)

The latest 2.2 version of the Linux kernel is: [2.2.26](#) 2004-02-25 00:28 UTC [F](#) [V](#) [Changelog](#)

The latest [prepatch](#) for the 2.2 Linux kernel tree is: [2.2.27-pre2](#) 2004-04-20 19:26 UTC [V](#) [VI](#) [Changelog](#)

The latest 2.0 version of the Linux kernel is: [2.0.40](#) 2004-02-08 07:13 UTC [F](#) [V](#) [VI](#) [Changelog](#)

The latest [-mm patch](#) to the stable Linux kernels is: [2.6.6-rc3-mm2](#) 2004-05-05 07:34 UTC [V](#) [Changelog](#)

F = full source, V = view patch, VI = view incremental, C = current [changesets](#)
 Changelogs are provided by the kernel authors directly. Please don't write the webmaster about them.

Done Adblock

Change-Management

- vertraglich zugesicherte Leitungen und Abnahmebedingungen (Fehlerfreiheit)
- ebenso: vereinbarter Wartungszeitraum (Fehlerbehebung)
- gewünschte Änderungen (Change-Requests)
- Änderungen betreffen Builds bzw. Releases
- Koordination mit Versionkontrolle notwendig

Fazit

- Versionsmanagement
 - schon bei kleinen Projekten
 - zwingend bei mehreren Entwicklern
 - kann gute Kommunikation nicht ersetzen
- Build- und Releasemanagement
 - ab mittlerer Projektgrösse
 - bei Vertragsprojekten / Produkten

Literatur, Verweise

- F. Schweiggert: Software Engineering Praxis, Skript, Universität Ulm, 2003
- J. Siedersleben: Softwaretechnik: Praxiswissen für Softwareingenieure, 2. Aufl., Carl Hanser Verlag, 2003
- cvs: <http://www.cvshome.org/>
- QSC Team Coherence
<http://www.qsc.co.uk/teamcoherence.htm>