

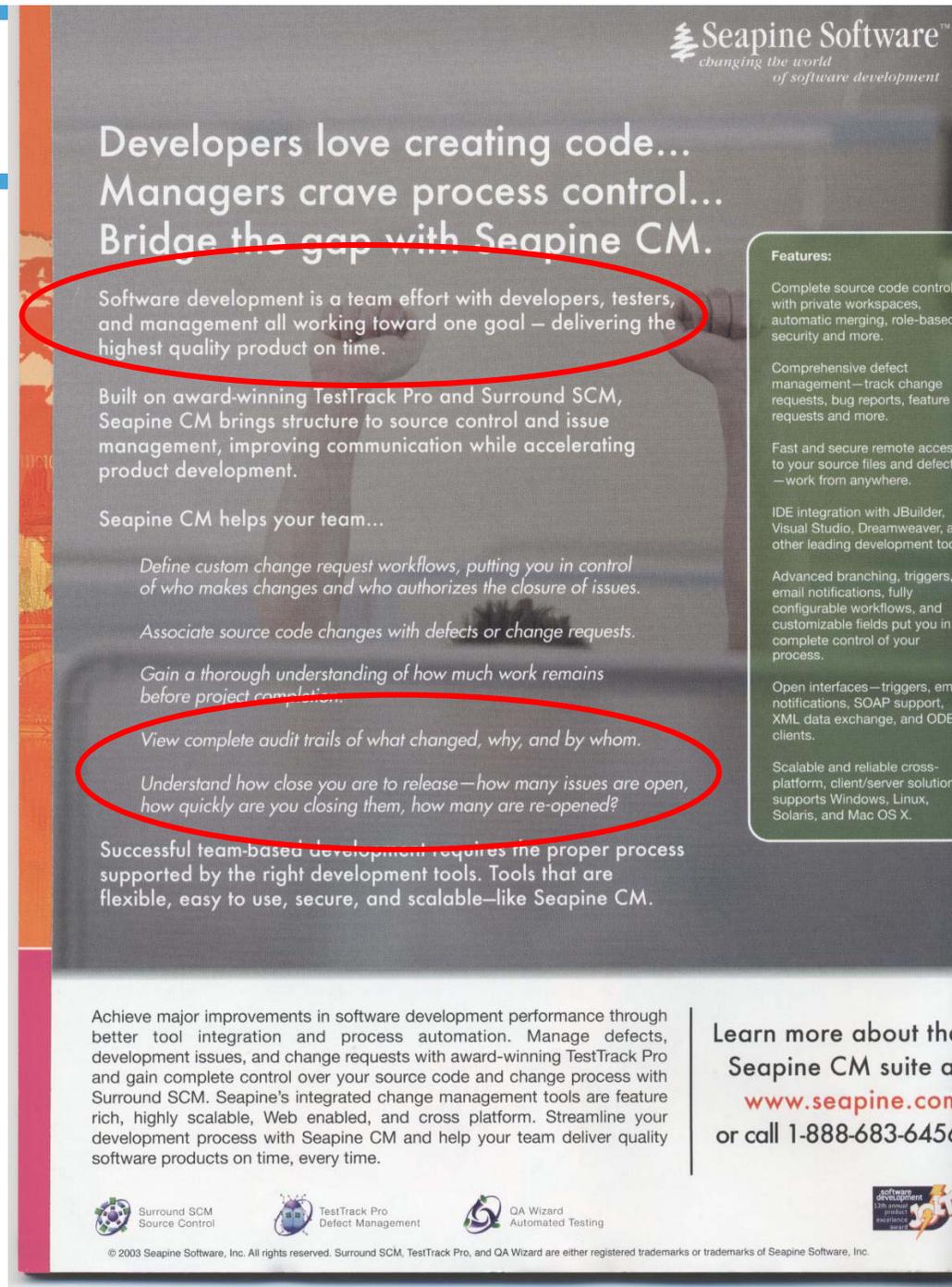
3.4 Konfigurationsmanagement (SCM)



"Das KM stellt einen Mechanismus zur Identifizierung, Lenkung und Rückverfolgung der Versionen jedes Softwareelements dar. In vielen Fällen sind auch frühere, nach wie vor in Verwendung befindliche Versionen zu warten und lenken."

ISO-9000-3

SCM ist Bestandteil zahlreicher Vorgehensmodelle und Standard-Entwicklungsprozesse (V-Modell, CMM, ...)



Seapine Software™
*changing the world
of software development*

Developers love creating code... Managers crave process control... Bridge the gap with Seapine CM.

Software development is a team effort with developers, testers, and management all working toward one goal – delivering the highest quality product on time.

Built on award-winning TestTrack Pro and Surround SCM, Seapine CM brings structure to source control and issue management, improving communication while accelerating product development.

Seapine CM helps your team...

- Define custom change request workflows, putting you in control of who makes changes and who authorizes the closure of issues.
- Associate source code changes with defects or change requests.
- Gain a thorough understanding of how much work remains before project completion.
- View complete audit trails of what changed, why, and by whom.
- Understand how close you are to release—how many issues are open, how quickly are you closing them, how many are re-opened?

Successful team-based development requires the proper process supported by the right development tools. Tools that are flexible, easy to use, secure, and scalable—like Seapine CM.

Achieve major improvements in software development performance through better tool integration and process automation. Manage defects, development issues, and change requests with award-winning TestTrack Pro and gain complete control over your source code and change process with Surround SCM. Seapine's integrated change management tools are feature rich, highly scalable, Web enabled, and cross platform. Streamline your development process with Seapine CM and help your team deliver quality software products on time, every time.

Learn more about the
Seapine CM suite at
www.seapine.com
or call 1-888-683-6456

Surround SCM Source Control | TestTrack Pro Defect Management | QA Wizard Automated Testing

© 2003 Seapine Software, Inc. All rights reserved. Surround SCM, TestTrack Pro, and QA Wizard are either registered trademarks or trademarks of Seapine Software, Inc.



Complete source code control with private workspaces, automatic merging, role-based security and more.

Comprehensive defect management—track change requests, bug reports, feature requests and more.

Fast and secure remote access to your source files and defects—work from anywhere.

IDE integration with JBuilder, Visual Studio, Dreamweaver, and other leading development tools

Advanced branching, triggers, email notifications, fully configurable workflows, and customizable fields put you in complete control of your process.

Open interfaces—triggers, email notifications, SOAP support, XML data exchange, and ODBC clients.

Scalable and reliable cross-platform, client/server solution supports Windows, Linux,

SCM Motivation



- Software-Systeme sind umfangreich und komplex:
 - mlocs in $n \gg 100$ Dateien
 - eingebundene Fremdprodukte, generierte Komponenten (GUI, OR-Mapping)
 - verschiedenen Sprachen / Compiler / Interpreter / Plattformen
 - zahlreiche Zusatzdokumente: Anforderungen, Design, Modelle, Manuale, ...

- Parallelarbeit (ggf. geografisch verteilt)

- Herausforderungen:
 1. Aktuelle, vollständige Zusammensetzung (Konfiguration) des Systems
 2. Wiederherstellen des Entwicklungsstandes einer/s Release oder Stichtages
 3. Koordination und Integration der parallelen Arbeiten

- Lösung: **Konfigurationsmanagement (SCM)**
 - Prozess, Rollen, Werkzeugunterstützung

Ohne SCM 1 ...



- Programm auf n Rechner verstreut, uneinheitlich strukturiert
- Binärpakete (exe, dll, jar, ...) auf Zuruf erstellt
- vollständige Übersetzung aktuelle Release manuell & aufwendig
- Entwickler müssen beachten, wer/wann/wo arbeitet
- Integration paralleler Änderungen manuell & aufwendig
 - Beispiel: Bearbeitung eines MS-Word Dokumentes im Team →



III. Ausführliche Beschreibung der geplanten FuE

.s. Unterkapitel

III.1 Ausgangssituation

[Redacted] Schilderung der Ausgangssituation nach [Redacted] der Unternehmen: Wissenstransfer, Netzwerkbildung, auch: Vernetzung der bestehenden Verbände und Vereine.
1/2 Seite

<Für die Arbeitspaket-Themen im Einzelnen Ergebnisse von [Redacted] sowie weitere nötige Arbeitsschritte nennen, dann diese Arbeitsschritte unter „Vorgehensweise“ noch weiter aufschlüsseln und beschreiben.>
Community-Building/Vernetzung, Wissensmanagement, Aufbereitung Know-how/Anpassung Methoden/Ergänzung durch emp. Studien

- Software-Anteil im Automobil- und Maschinenbau sowie der Elektroindustrie steigt stetig (<Fix Me: These belegen>)
- ingenieurmäßige Softwareentwicklung ist Kernkompetenz und Erfolgsfaktor (schnelle Adaption neuer Technologien / Methoden liefert Wettbewerbsvorteile, Vorsprung vor Konkurrenz, ...)
- Bislang Defizite im Umgang mit Software-Entwicklung (Wissensdefizite, Mangel an Erfahrung, z.B. „umgeschulter Maschinenbauer“). <Fix Me: Aussagen aus der [Redacted] Studie einfügen>
- Dritte Kernkompetenz (laut [Redacted] neben Maschinenbau und E-Technik, jetzt Software-Engineering)

- Ausgangssituation (Problem, Motivation)
- Darstellung des Ziels, geplante Vorgehensweise
- Beschreibung der einzelnen Arbeitspakete (grob)
- Dokumentation effektiven Projektmanagements
- Arbeitspakete sollten Evaluierungspaket enthalten!
- Nutzung vorliegender Ergebnisse (z.B. Standards)
- Arbeitspakete sollten Dissemination vorsehen
- Achten, dass Beschreibung auf Themengebiet nach

Kommentar [RK4]: Abgrenzung zu Abschnitt II ist mir noch nicht ganz klar – sollten hier die Ergebnisse von [Redacted] noch einmal ausführlicher dargestellt werden, evtl. mit mehr quantitativen Daten.

III.2 Ziele und Vorgehensweise

[Redacted] Beschreibung des Projektziels und der verfolgten Lösungsstrategie (Vorgehensweise im Projekt), idealerweise mit Diagrammen illustriert.
1/2-1 Seite

Die systematische Aktualisierung, Erweiterung und Pflege des Software-Engineering-Know-Hows in kleinen und mittleren Unternehmen ist das Ziel von [Redacted]. Das heißt:

1. Systematischer Kompetenzaufbau und Wissensbeschaffung
2. Aktives Wissensmanagement und Wissenstransfer in KMUs (Dissemination)
3. Verbesserte Verbreitung über Betrieb durch Interessenverband aus der Industrie, verstärktes Marketing

Abbildung x visualisiert das als [Redacted]-Haus. Das Software-Engineering Kompetenzzentrum [Redacted] steht auf dem Fundament von bewährtem, qualitätsgesichertem Software-Engineering Wissen.

Über die fünf Säulen: [Redacted], Ansprechpartner, Weiterbildung, Erfahrungsaustausch und Beratung werden kleinere und mittlere Unternehmen in Kompetenzaufbau und -pflege aktiv unterstützt.

Der Betrieb durch einen Interessenverband aus der Industrie ([Redacted]) und ein verstärktes Marketing sorgen für Verbreitung und Akzeptanz der Angebote des Software-Engineering Kompetenzzentrums.

Kommentar [RK5]: Das ist kein „Forschungsziel“ für das [Redacted]. Sollen wir nicht eher auf die Aufbereitung und Rückführung von Industriefahrung in die Forschung abzielen, so dass bedarfsgerecht geforscht wird?

Gelöscht: KMUs

Gelöscht: Das bedeutet

Gelöscht: - Betrieb von [Redacted] durch Interessenverband aus der Industrie zur Verbesserung der Verbreitung und der Akzeptanz[.]

Gelöscht: -

Gelöscht: im Bereich des Software-Engineerings für KMUs und

Gelöscht: - Proaktiver Umgang mit neuen Methoden und Technologien.[.]

... ohne SCM 2



- alte Release nicht wieder herstellbar
 - Fehlermeldung weder nachvollziehbar noch behebbbar
 - durchgeführte Änderung nicht ohne weiteres reversibel
- individuelle Änderungen für eine Release unmöglich
- Dokumentation & Programm, i.d.R. nicht konsistent
- Änderungshistorie unbekannt
- keine statistischen Daten bzw. Trendanalyse über Evolution

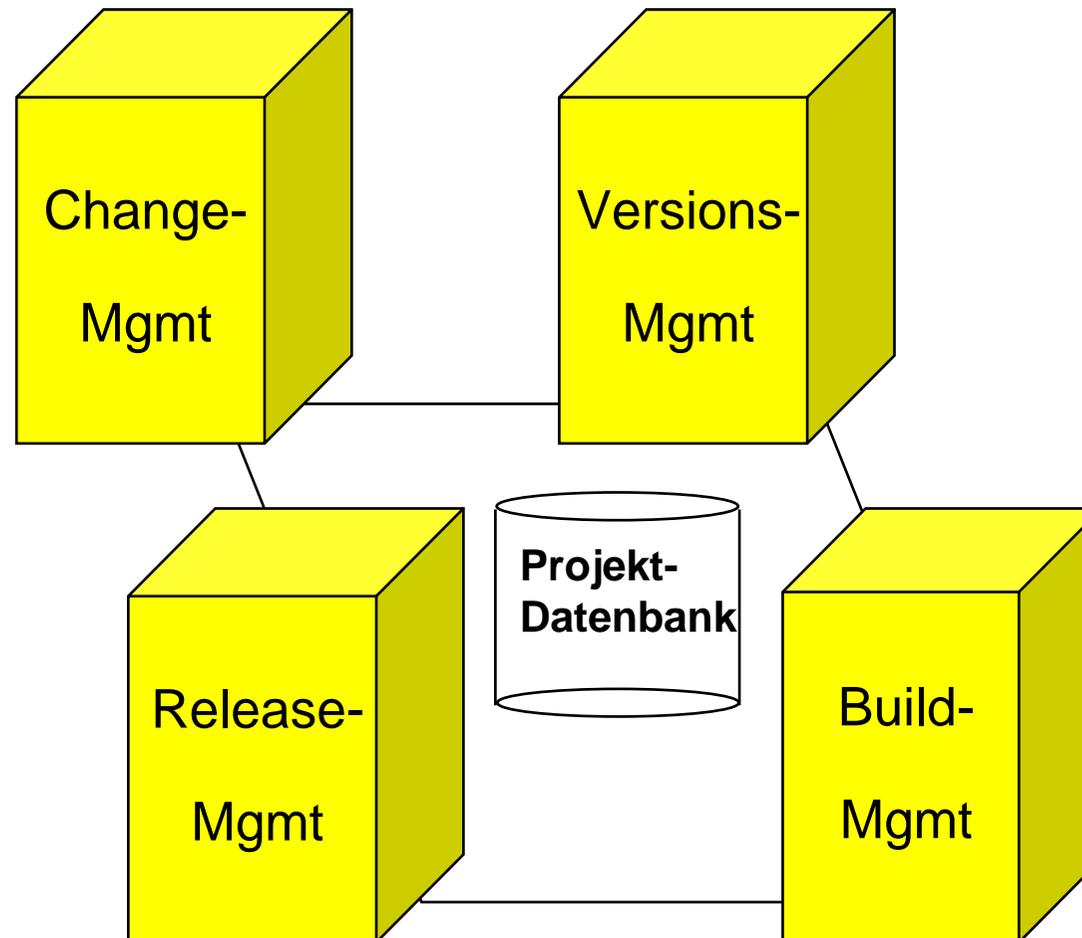
Primitiv SCM



- Benennung von Dateien/Verzeichnissen mit Datum
- z.B. help-040609.c
- Defizite
 - manueller Aufwand, Fehlerpotential
 - nach wie vor unklar,
wer warum und was genau in help.c am 040609 geändert hat
 - Parallelarbeit an help.c?
 - Manipulation des Dateinamens vor der Übersetzung des Projekts?



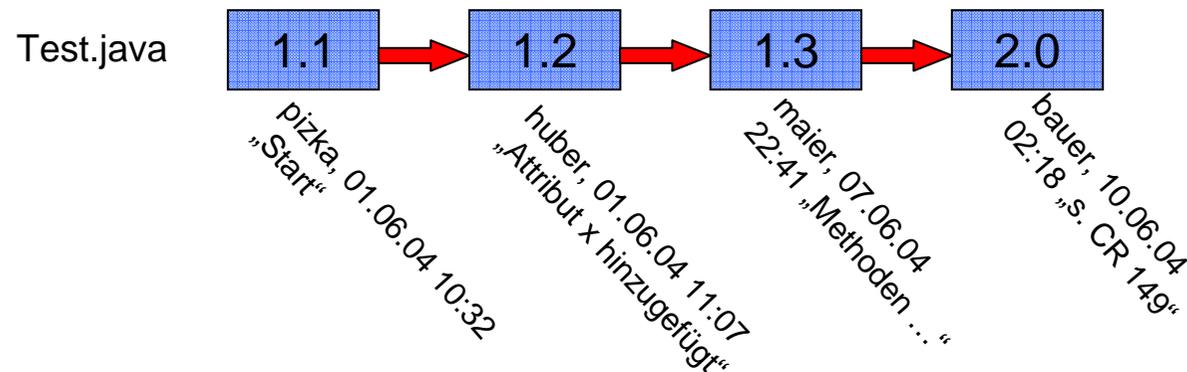
Säulen des SCM



Versionsmanagement (VM)



- Software = Menge D von Dokumenten (inkl. Programme, Libs, ...)
- Jedes $d \in D$ **individuell versioniert**
 - Erweiterung der Bezeichnung von d mit einer **Versionskennung** (-nummer)
 - Festlegung einer neuen Version von d nach Änderung explizit ~ **commit**
i.d.R. mit **Erläuterung**, was + warum
 - **Archivierung** aller Versionen von d inkl. Erläuterungen

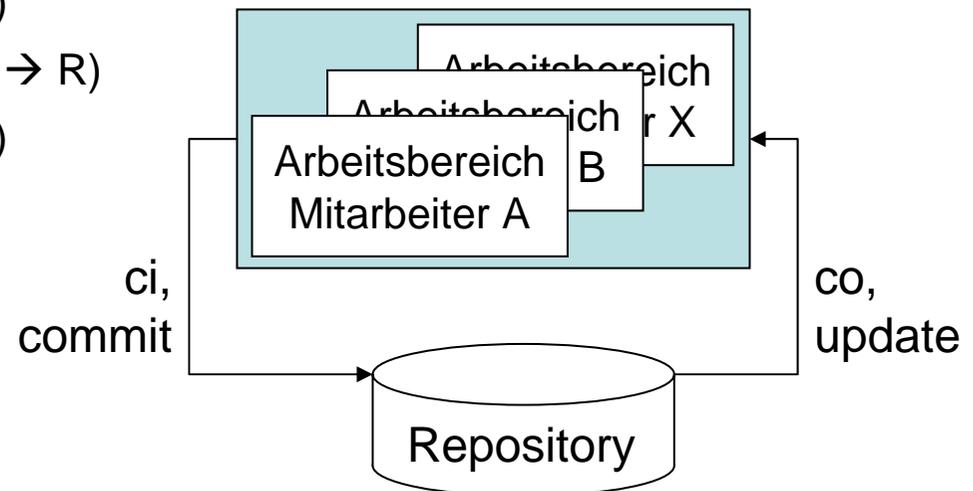


- Voraussetzung: Werkzeug-Unterstützung (z.B. RCS)

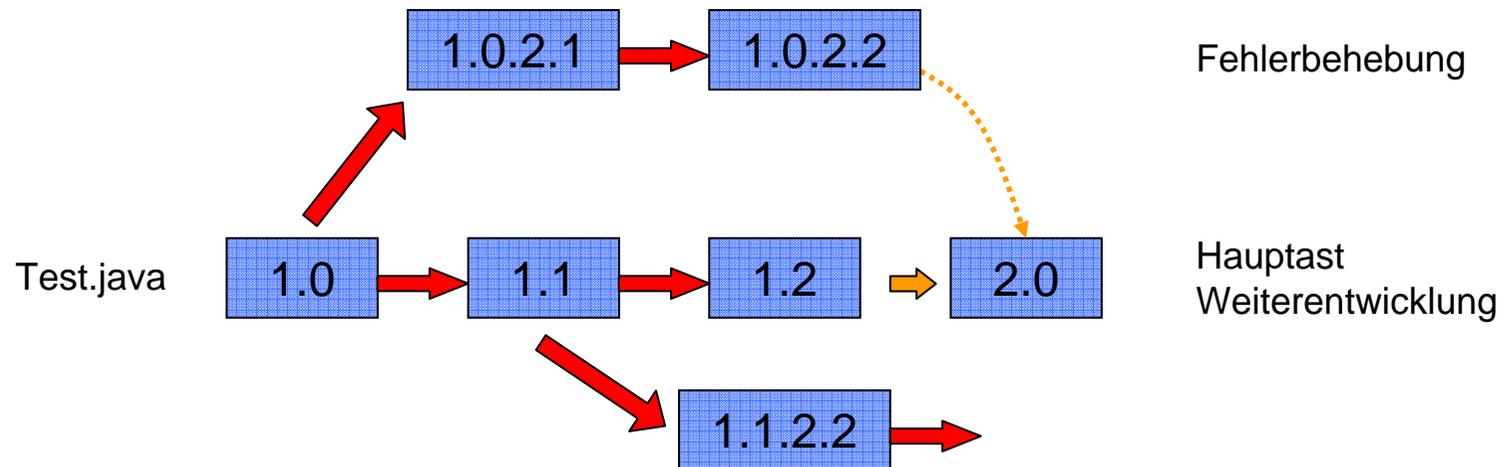
Archiv und Arbeitsbereich



- Ablage aller Versionen aller $d \in D$ in **gemeinsamen Archiv** (Repository)
- Entwickler arbeiten auf Kopien in **privaten Arbeitsbereichen** (Workspace)
- Transfer von $d \in D$ zwischen Archiv und Arbeitsbereich:
 - **Check-Out** (neues Dok $R \rightarrow W$)
 - **Update** (W aktualisieren)
 - **Check-In** (neues Dok $W \rightarrow R$)
 - **Commit** (R aktualisieren)



Versionsbäume



- Szenario: Fehlerbehebung in Version 1.0, wobei 1.2 aktuell
- **Verzweigung** von Version 1.0
- individuelle Fehlerbehebung **parallel** zur Weiterentwicklung
- ggf. **Vereinigung** mit Hauptast (u.U. aufwendig)

Fortgeschrittenes VM



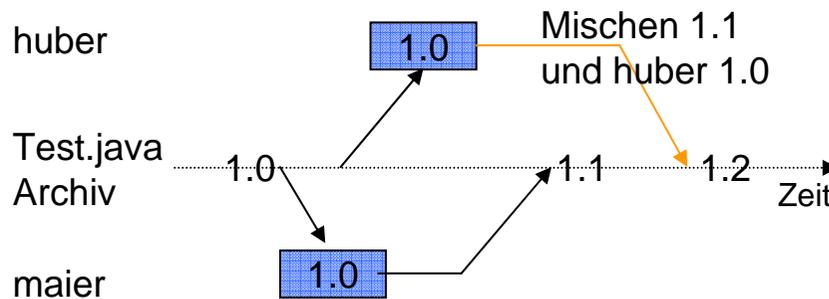
- Visualisierung von Unterschieden und Erstellung von „Patches“
- verteilte Archive
- automatische Benachrichtigung
- automatische Prüfung bei commit (z.B. Programmierrichtlinien)
- Sperren (Voreinstellung i.d.R. optimistisches Locking)
- differenzierte Rechte
- statistische Auswertungen
- Versionierung der System-Umgebung inkl. Hardware :-)

VM – Technische Aspekte



- Client/Server Architektur → verteilter Zugriff
- Speicherung von Differenzen → reduzierter Speicherbedarf
- Differenzbildung auf Zeilenebene
- bei *update*
 - Analyse Differenz Archiv/Arbeitsbereich
 - autom. (zeilenweises) Mischen

paralleles Ändern an genau 1 Datei



Einschränkungen?

```

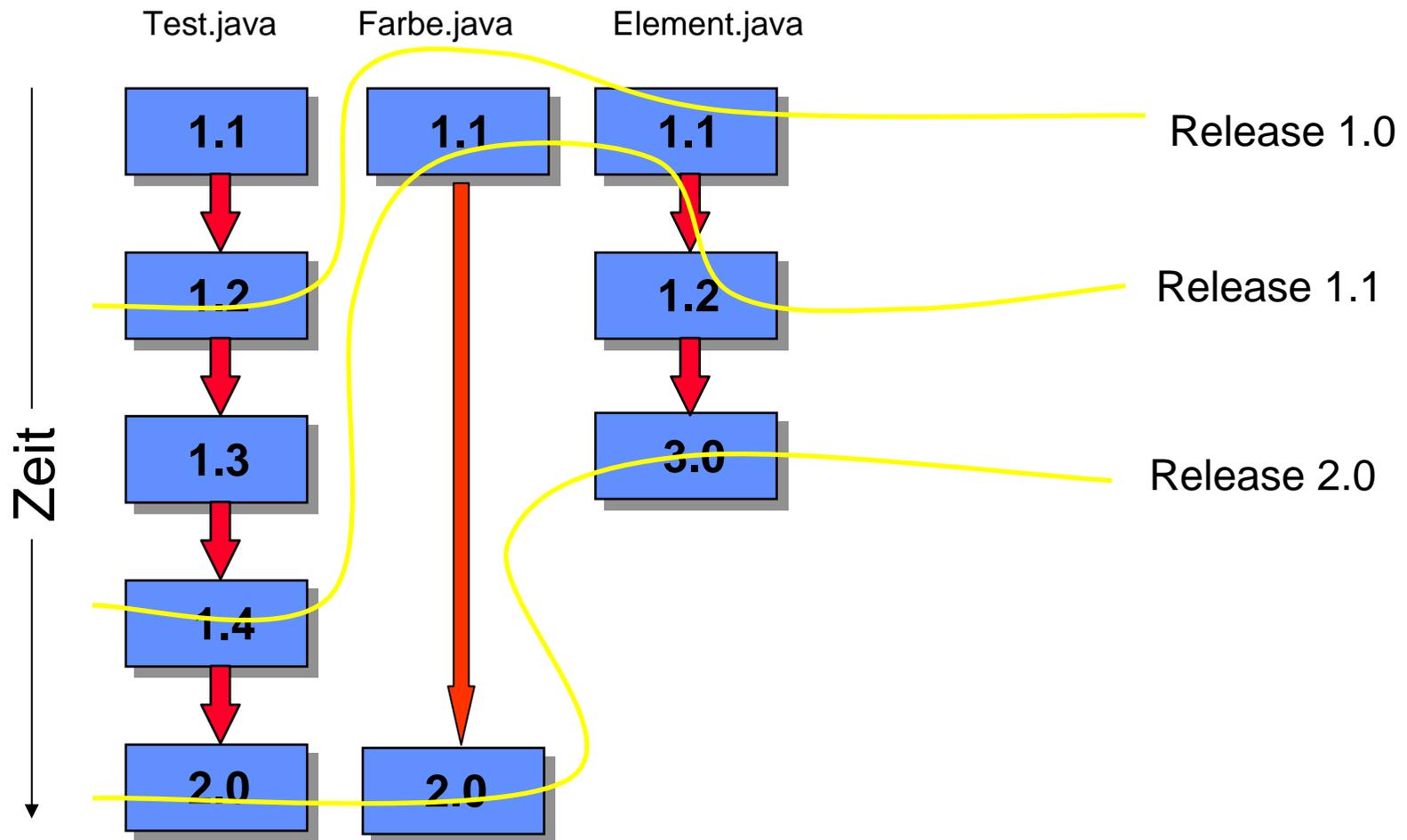
head      1.1;
access;
symbols;
locks; strict;
comment   @% @;
1.1
date      98.04.27.09.30.56; author pizka;
state Exp;
branches;
next;

1.1
log
@checking of final version of ICSE-01 paper
@
text
@%
% $Author: pizka $
% $Date: 1995/10/09 15:20:59 $
% $Revision: 0 $

\documentstyle[times,art10,twocolumn,latex8,e
psf]{article}

\def\fig#1{/usr/wiss/pizka/lib/bilder/figs/#1}
\def\bib#1{/usr/literatur/#1}
  
```

Versionen und Releases



Releasemanagement



- Arten von Releases:
 - haupt (Major): neue Produktversion für Kunde
 - neben (Minor): kompatibel, Korrektur/Anpassung/geringfügige Erweiterung
 - intern: zur „in-house“ Verwendung durch das Entwicklerteam

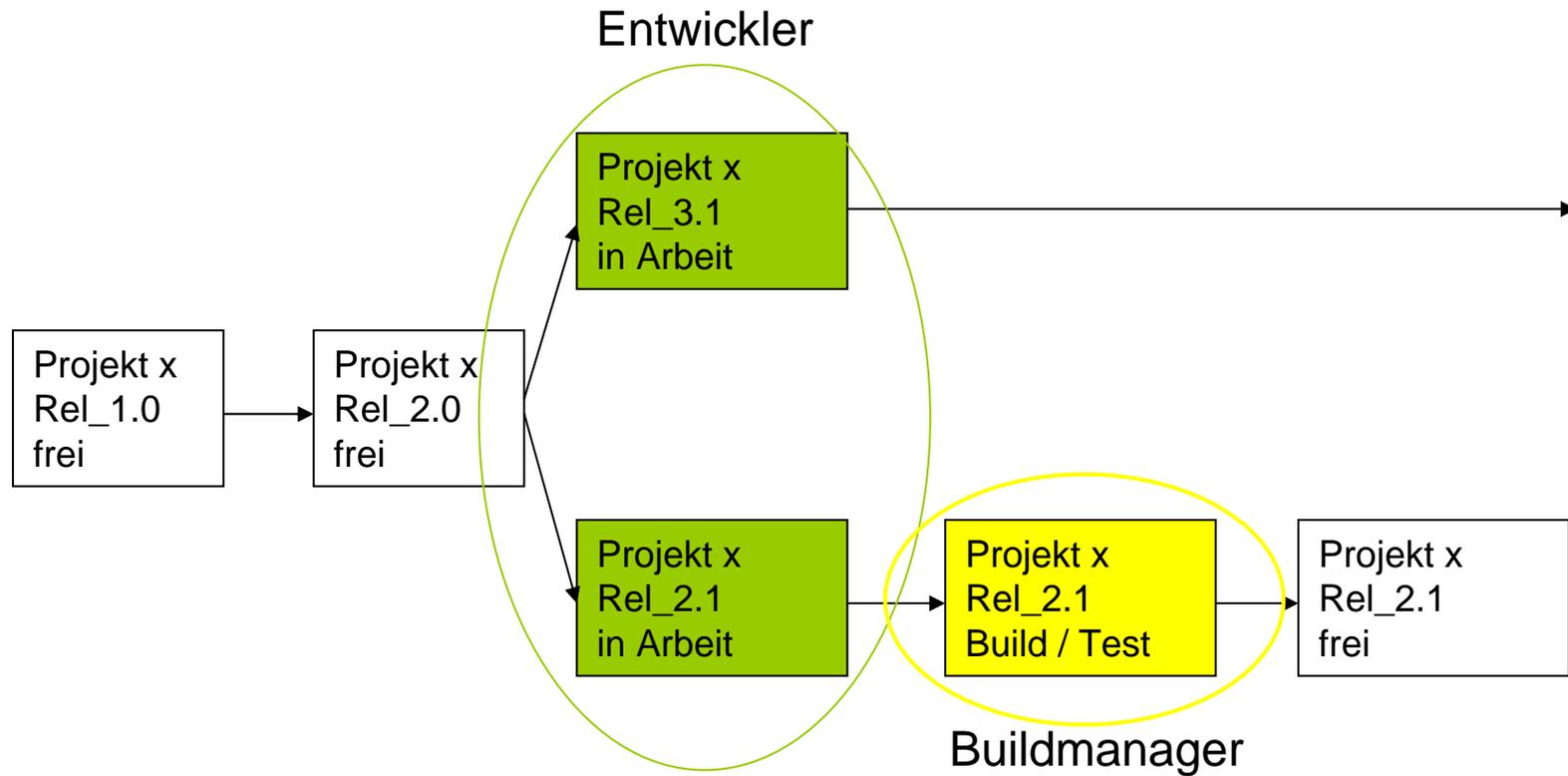
- häufig parallele Entwicklung an verschiedenen Releases in separaten Zweigen der Hauptentwicklung

Buildmanagement (1)



- V/R-Management gewährleistet keine **syntaktische / semantische Konsistenz**
- notwendig: **Produktion** (Build) des Zielsystems aus Archivinhalt
 - Generierung / Beschaffung zusätzlicher Dokumente
 - Übersetzung, Binden
 - Test
- Zeitbedarf oft erheblich! Entwicklung soll nicht blockiert werden!
- Lösung:
 1. Build-Spezifikationen und Werkzeuge (make, ANT)
 2. Release-Zustände
 3. Berechtigungskonzept

Buildmanagement (2)



Variantenmanagement



- Zusätzliche Dimension des SCM zu Versionen und Releases
- Unterschiedliche Varianten desselben Produkts
 - verschiedene Branchen
 - verschiedene Länder/Regionen (vgl. Microsoft Produkte)
 - verschiedene Leistungsstufen
- Prinzip:
 - Archiv i.d.R. allgemeinste Version
 - Berücksichtigung der Spezifika entweder
 - durch Konfiguration bei der Installation
 - Selektion von Dokumenten durch SCM-Tool (selten)
 - erweiterter Build-Prozess zur Produktion der Produkt-Varianten

KM-Werkzeuge – WinCVS (OS)



The screenshot displays the WinCVS application window. The title bar reads "[Untitled]". The menu bar includes File, Edit, View, Cvs Admin, Cvs Files, Cvs Folders, Selections, Window, and Help. The toolbar contains various icons for file operations and CVS-specific actions. The left pane shows a tree view of the local CVS repository structure, including folders like 'admin', 'cvstest', and 'km'. The main pane displays a table of files with columns for Name, Rev., Option, Status, Tag, Date, and Conflict. The file 'pvcs.m4' is selected. A context menu is open over the selected file, showing options like 'Update selection...', 'Commit selection...', 'Add selection', etc. The bottom pane shows the output of a 'cvs update' command, indicating that the local files are being updated from the server.

Name	Rev.	Option	Status	Tag	Date	Conflict
clearcase.m4	1.8		File		Fri Mar 19 14:42:51 1999	
continuous.m4	1.6		File		Mon Mar 08 15:25:00 1999	
cvs.m4	1.5		File		Mon Mar 08 15:25:01 1999	
index.m4	1.3		File		Mon Mar 01 13:45:27 1999	
pvcs.m4	1.1.1.1		File			
qvcs.m4	1.1.1.1		File			
razor.m4	1.1.1.1		File			

```

cvs update -P -d (in directory D:\cvslocal\km)
cvs server: Updating .
cvs server: Updating Bilder
cvs server: Updating Kontakte
cvs server: Updating Produkte
P Produkte/clearcase.m4
cvs server: Updating Projekte
cvs server: Updating Quellen
cvs server: Updating Themen
cvs server: Updating Wissensbroker

*****CVS exited normally with code 0*****
  
```

KM-Werkzeuge – TortoiseCVS (OS)



The screenshot shows a Windows Explorer window titled 'Dokumente' with the address bar set to 'D:\pizka\proj\vissek\WISEK_I_2001_2003\Dokumente'. The left pane shows a tree view with folders like 'PQL-04', 'vissek', and 'WISEK_I_2001_2003'. The right pane shows a list of files including 'vissek_literatur.bib', '030624_Rumpe_SERP03_UML...', and '030618_WISEK_Review_Repor...'. A context menu is open over the file '030618_WISEK_Review_Repor...', with 'CVS' selected. The 'CVS' submenu is open, showing options like 'Webansicht...', 'Historie...', and 'Revisionsgraph...'. The 'Revisionsgraph' option is highlighted, and a separate window titled 'TortoiseCVS - Revisionsgraph' is overlaid on the right, showing a vertical revision history for the file '030801_pspm_evolution.tex' with versions 1.1 through 1.11.

Kommerzielle Werkzeuge



- Telelogic Synergy / CM
- Rational (IBM) ClearCase
- PVCS
- MKS Integrity
- ...