

Inhalt der Vorlesung

- Einführung
- Kommunikation
- Software Qualität
- Vorgehensmodelle
- Requirements Engineering
- Konfiguration Management
- Software Architektur und – Design

- **G. Popp: Konfigurationsmanagement, dpunkt.verlag, 4.Auflage, 2013**
- **Maven by Example:**
<http://www.sonatype.com/resources/books/maven-by-example>
- **Maven Reference:**
<http://books.sonatype.com/mvnref-book/reference/index.html>
- **Jenkins – the definitive guide**
- **Hudson Book**

- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

Man sagt:

Wer Ordnung hält, ist nur zu faul zum Suchen!

Oder auch:

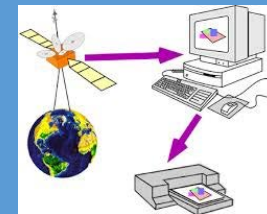
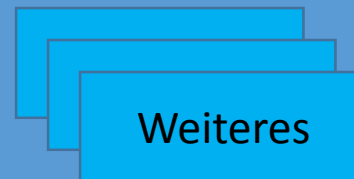
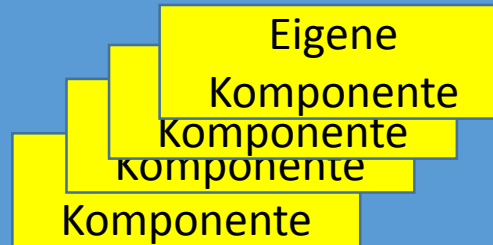
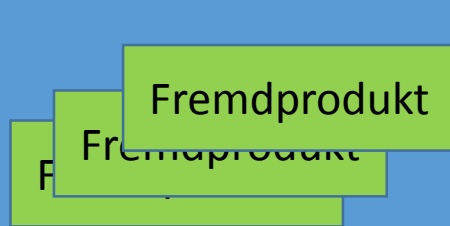
Wer Ordnung hält, ist nur zu schlau zum Suchen!

Komplexität in einem Software Projekt

- Zu einem SW-System gehören eine Vielzahl von Software-Elementen verschiedenster Typen:
 - Text-Dokumente, Spezifikationen,
 - (halb-)formale grafische Quellen
 - Code, Objekte, Module, Komponenten
 - aber auch: Werkzeuge, Basis-Software, ...
- Große SW-Systeme sind bereits komplex allein durch die Quantität ihrer Bausteine
 - Die Anzahl der zu verwaltenden Einheiten (Software-Elemente) liegt oftmals im drei- oder gar vierstelligen Bereich.

Motivation

SW Produkt Zutaten



- Verschiedenste technische Systeme müssen zusammenspielen:
 - Rechner (Host, Web-Server, PC's, Netzwerke)
 - Betriebssysteme
 - Plattformen, Werkzeuge, Entwicklungsumgebungen
- Schnittstellen verlangen z.T. enge Integration mit anderen Systemen
- Entwicklung und Wartung geschieht
 - mit vielen Menschen
 - an vielen Orten
 - manchmal für viele Kunden

Dynamik

- Bei der Entwicklung und Wartung eines SW-Systems haben wir mit vielen Objekten, mit vielen, meistens komplexen Beziehungen zwischen Objekten zu tun.
- Die Objekte und Beziehungen ändern sich mit der Zeit
 - Änderungen der angeforderten Funktionalität
 - Source Code Änderungen
 - Plattformwechsel
 - Kundenwechsel
- Diese Dynamik erschwert die Beherrschung des Systems.

Motivation

Wer auch nur ein bisschen schlau ist und das Chaos meiden möchte, betreibt bei der SW-Entwicklung Konfigurations-Management.

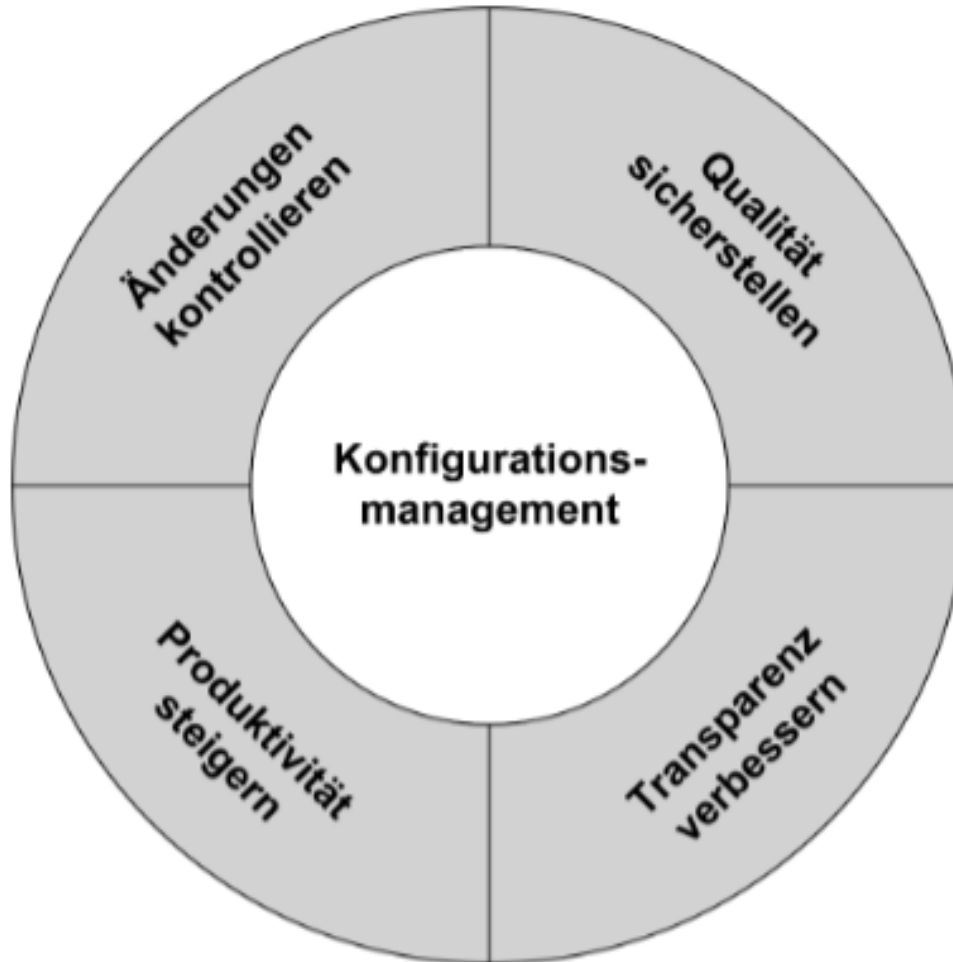
- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

Konfigurationsmanagement:
Projektergebnisse sicher verwalten und den Teammitgliedern kontrolliert Zugriff gewähren.

Ziele:

- **Änderungen kontrollieren**
- **Kommunikation vereinfachen/Transparenz verbessern**
- **Qualität sicherstellen**
- **Produktivität steigern**

Ziele des Konfigurationsmanagements



Quelle des Bilds: G. Popp: Konfigurationsmanagement, dpunkt.verlag,

Konfigurationsmanagement ist der Prozess, ein sich änderndes Softwaresystem zu verwalten.

Grundsätzliche Aktivitäten:

- 1. Versionsmanagement**

Verwaltung der verschiedenen Versionen der erstellten Artefakte.

- 2. Systemintegration**

Unterstützung bei der Festlegung, welche Version welcher Komponente verwendet wird.

- 3. Problemverfolgung**

Quelle: Ian Sommerville: Software Engineering

- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

SWE Book

(<http://www.computer.org/web/swebok/v3>):

A configuration item (CI) is an item or aggregation of hardware or software or both that is designed to be managed as a single entity.

Wir verstehen unter einem Konfigurationselement einen Typ einer Gruppe von Artefakten, die dem zu entwickelnden Produkt zugeordnet sind und somit für das Konfigurationsmanagement relevant sind.

Bestimmung der Konfigurations-Elemente

- Konfig Elemente: Alle Elemente, die zur Erstellung des SW Produkts nötig sind oder dieses Produkt beschreiben.
- Die Projektablage enthält zusätzlich alle Daten, die zur Steuerung und Durchführung eines Projekts nötig sind.

Erster Schritt in der Einführung eines Konfig Management Prozesses: Auswahl der Elemente

Eindeutige Konfigurationselemente:

- Quelltext
- Anforderungsdokumente (z.B. Usecases)
- Architektur und Design Dokumente
- KM Handbuch
- Schnittstellenverträge
- Testspezifikationen und Testdaten
- Build Skripte
- Meta- und Konfigurationsdateien
- Benutzer Dokumentation
- Installationsanleitung, Release Notes etc.

*Quelle: G. Popp:
Konfigurationsmanagement*

Mögliche Konfig Elemente

- Werkzeuge (Entwicklungsumgebung, Compiler, Build Tools)
- Bibliotheken und Frameworks
- Generierte Artefakte

Keine Konfig Elemente

- Meetingprotokolle
- Binäre Auslieferungsdateien
- Generierte Dateien, z.B. kompilierte Quelltexte
- Projektpläne
- LOPs etc.

- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

Ein Beispiel für ein
Konfigurationsmanagement Handbuch finden
Sie unter

<http://www.km-buch.de/downloads.html>

und auf der elearning Plattform.

- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

Konfig Elemente sind identifiziert und werden im KM Handbuch beschrieben.

➔ mindestens zwei Angaben:

- Kurze inhaltliche Beschreibung und
- Namenstemplate

Namenstemplate muss drei Anforderungen erfüllen

- Jede Instanz des Konfig Elements muss anhand des Namens eindeutig identifizierbar sein.
- Der Name der Datei sollte auf das übergeordnete Element hinweisen.
- Beziehungen zwischen Konfig Elementen sollten aus den Dateinamen ersichtlich sein.

Bsp für Namenstemplates

Konfigurationselement : Use Case
Beschreibung : Ein Anwendungsfall des Systems
Namenstemplate : UC_<UseCaseName>.doc



UC_Katalog_öffnen.doc



UC_Preisliste_erstellen.doc



UC_Produkt_ändern.doc

Konfigurationselement : Entwurfsdokument
Beschreibung : Detailliertes Design pro Use Case
Namenstemplate : DES_<UseCaseName>.doc



DES_Katalog_öffnen.doc



DES_Preisliste_erstellen.doc

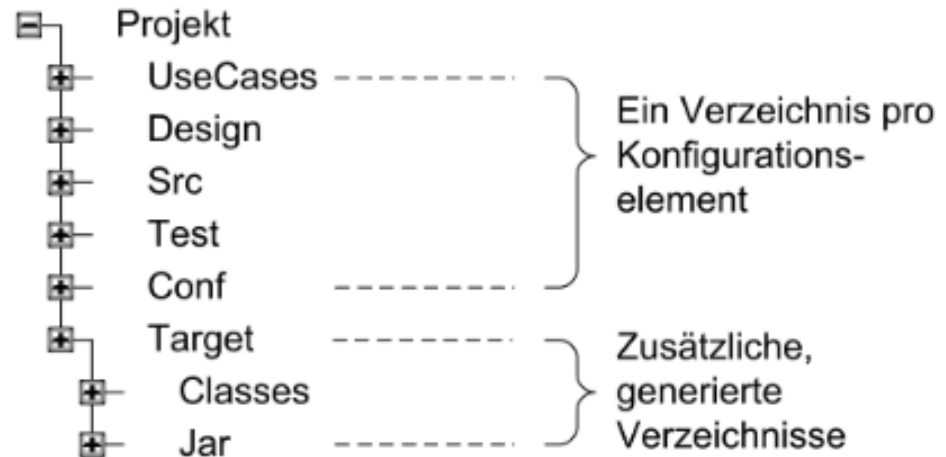


DES_Produkt_ändern.doc

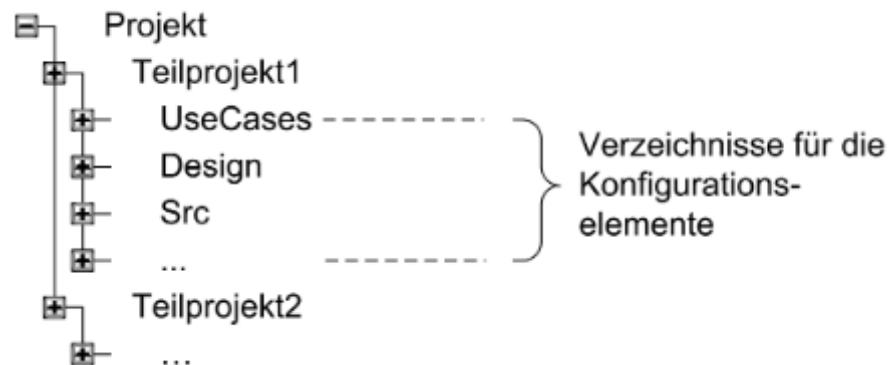
- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

Festlegung der Projektstruktur

Nach Konfig Elementen

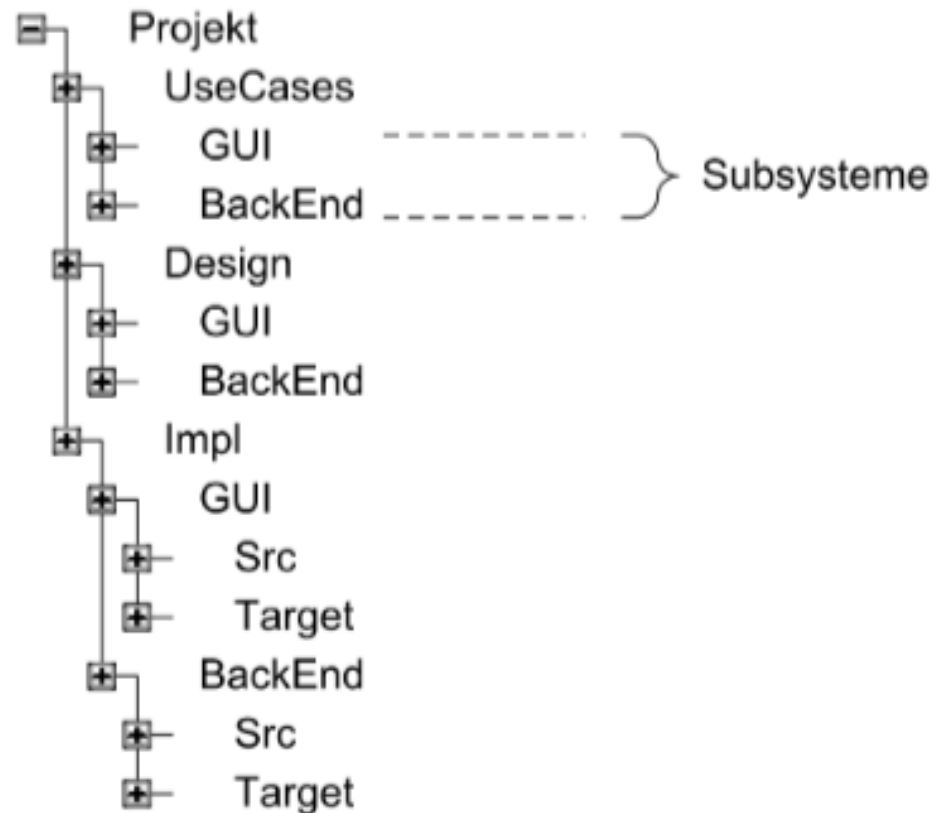


Nach Projekt Struktur



Festlegung der Projektstruktur

Nach Software Architektur



Weitere Einflussfaktoren auf die Projektstruktur

- Technische Einflussfaktoren wie z.B. das Tagging und Branching Konzept von Subversion müssen ebenfalls in der Struktur berücksichtigt werden.
- ➔ siehe Kapitel über Source Verwaltung

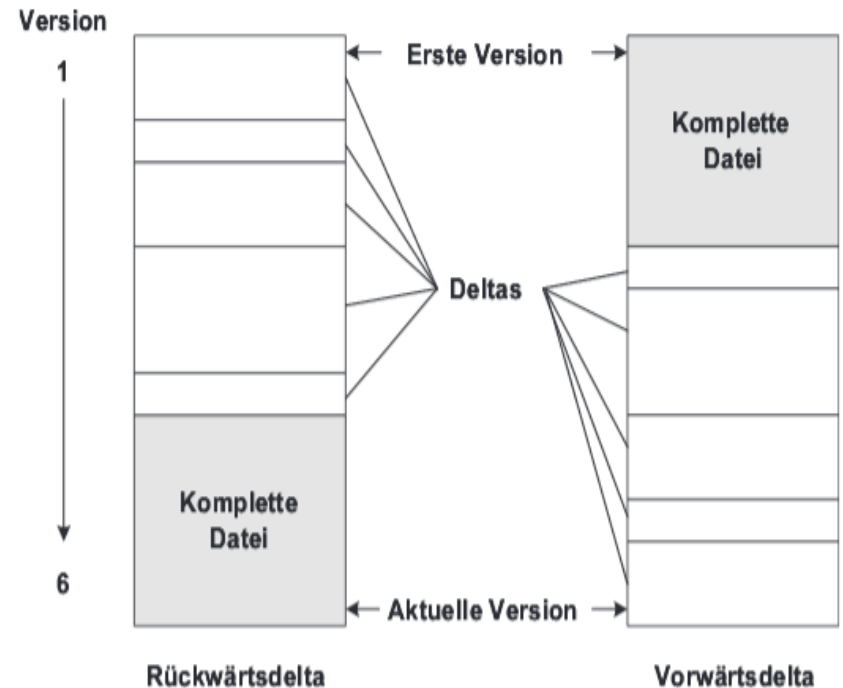
- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

- Sicherstellung der Verfügbarkeit der Dateien.
- Gewährleistung der Integrität, insbsd. bei gleichzeitigen Änderungen durch verschiedene Nutzer.
- Verhinderung unberechtigter Zugriffe.
- Nachvollziehbarkeit der Änderungen.

➔ Verwendung eines Repositories

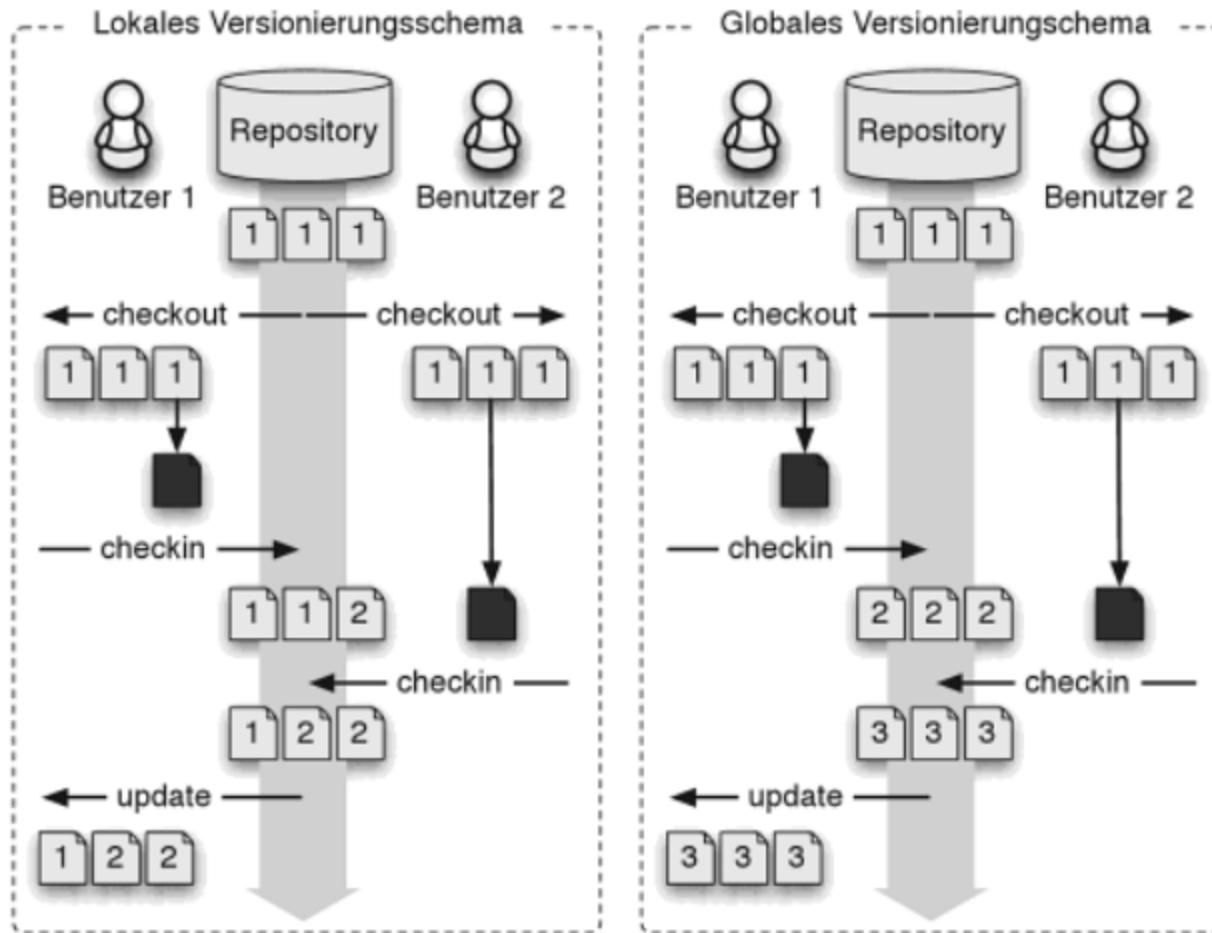
Versionierung durch das Repository erlaubt:

- Wiederherstellen eines alten Standes
- Wiederherstellen irrtümlich gelöschter Dateien
- Vergleich verschiedener Versionen einer Datei
- Parallele Arbeit an zwei unterschiedlichen Versionen einer Datei



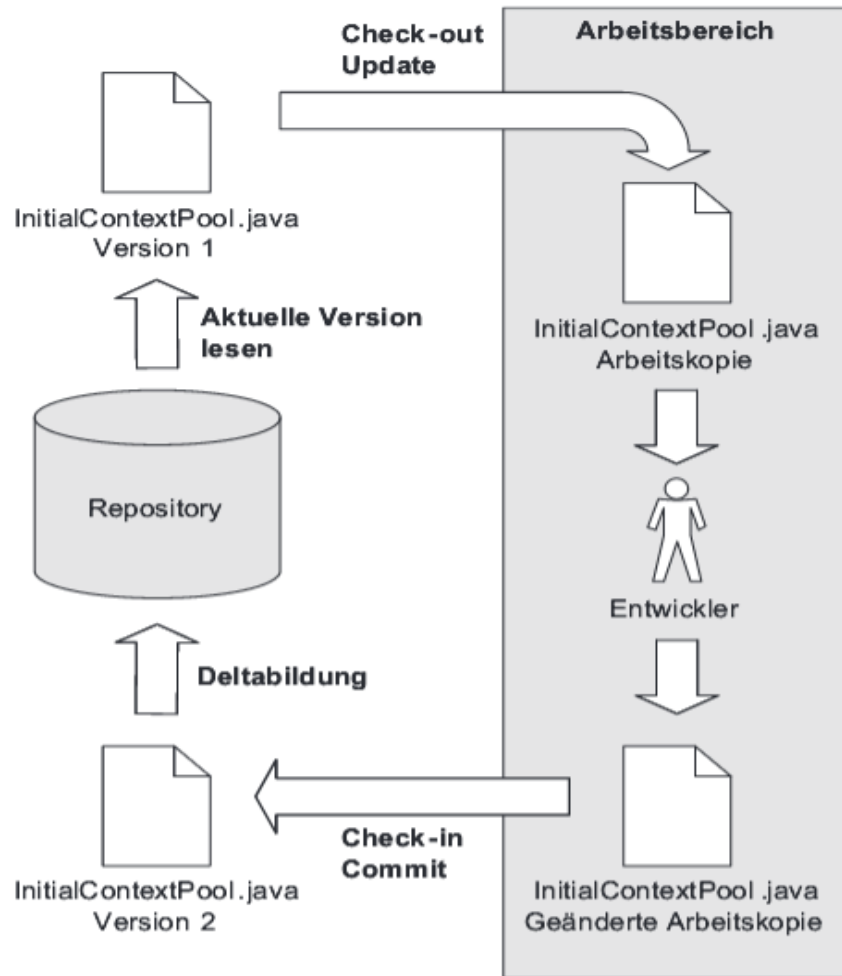
Quelle des Bilds: G. Popp: Konfigurationsmanagement, dpunkt.verlag,

Versionierung – lokal vs global



Quelle des Bilds: G. Popp: Konfigurationsmanagement, dpunkt.verlag,

Check-out und Check-in



*Quelle des Bilds:
G. Popp:
Konfigurationsmanagement,
dpunkt.verlag,*

Was tun bei parallelen Änderungen?

Zwei Ansätze:

- Lock – Modify –Unlock (Reserved Checkout)
- Copy – Modify – Merge (Unreserved Checkout)

Vor- und Nachteile

Vorteil Copy-Modify-Merge:
niemand wird blockiert

Vorteil Lock – Modify –Unlock:
Vermeidung von Konflikten
auch für binäre Dateien gangbar

Bei subversion pro Konfig Element konfigurierbar,
welcher Ansatz gewählt wird.

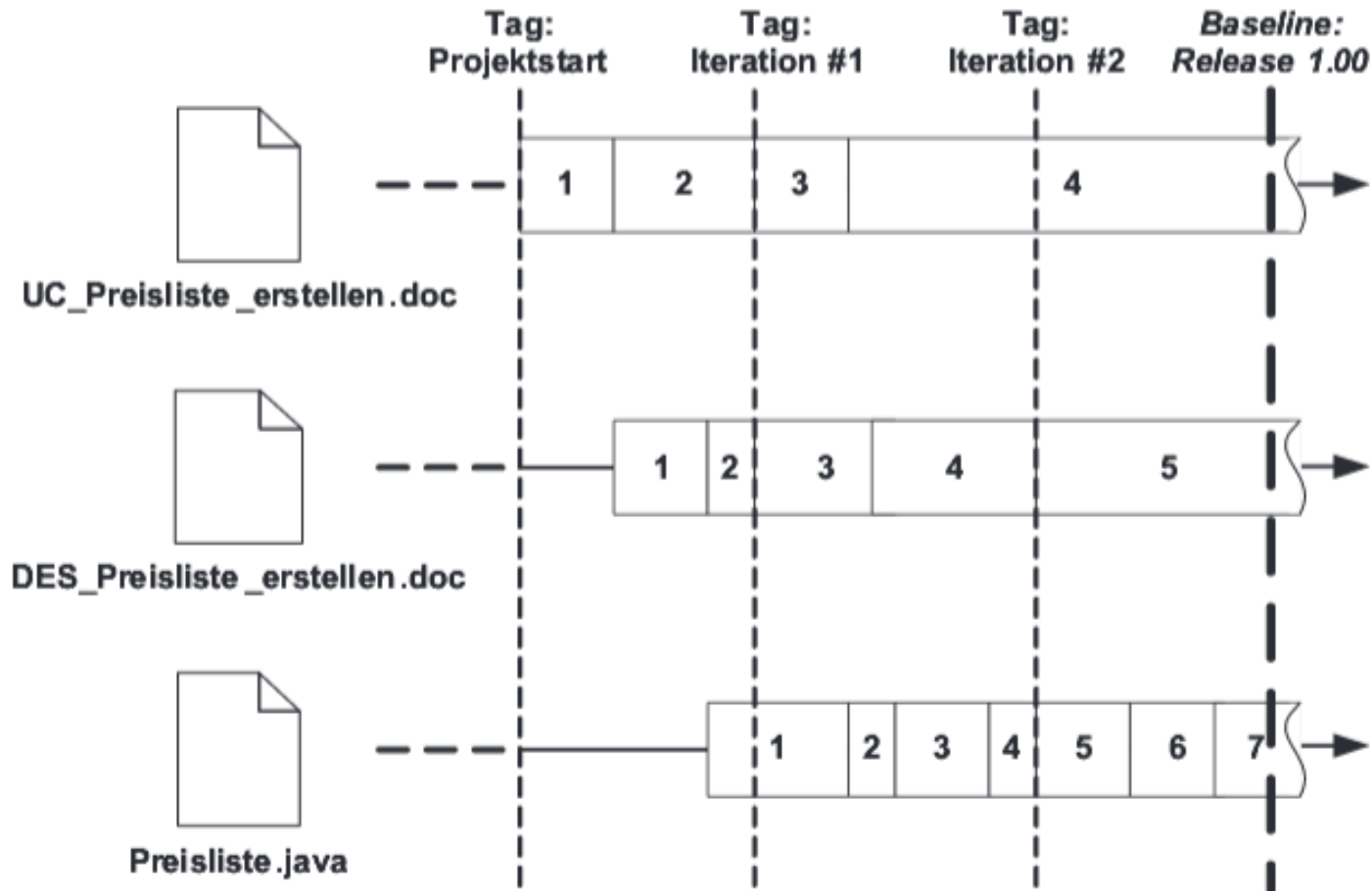
Tags und Baselines

Versionsnummern sind schwer zu merken. Daher bieten die Repositories die Möglichkeit, **Tags** zu erstellen – Bezeichner für eine zu einem Zeitpunkt gültige Version für alle Dateien des Repositories.

Ein bedeutsames Ereignis im Repository wird durch eine **baseline** repräsentiert – technisch kein Unterschied zum Tag.

Auslieferungen der Software werden **Releases** genannt. Für jedes Release wird eine baseline erstellt, aber nicht umgekehrt.

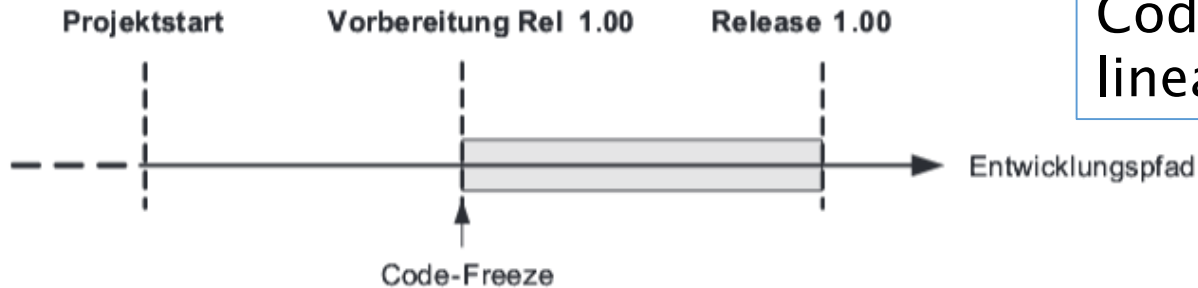
Tags und Baselines



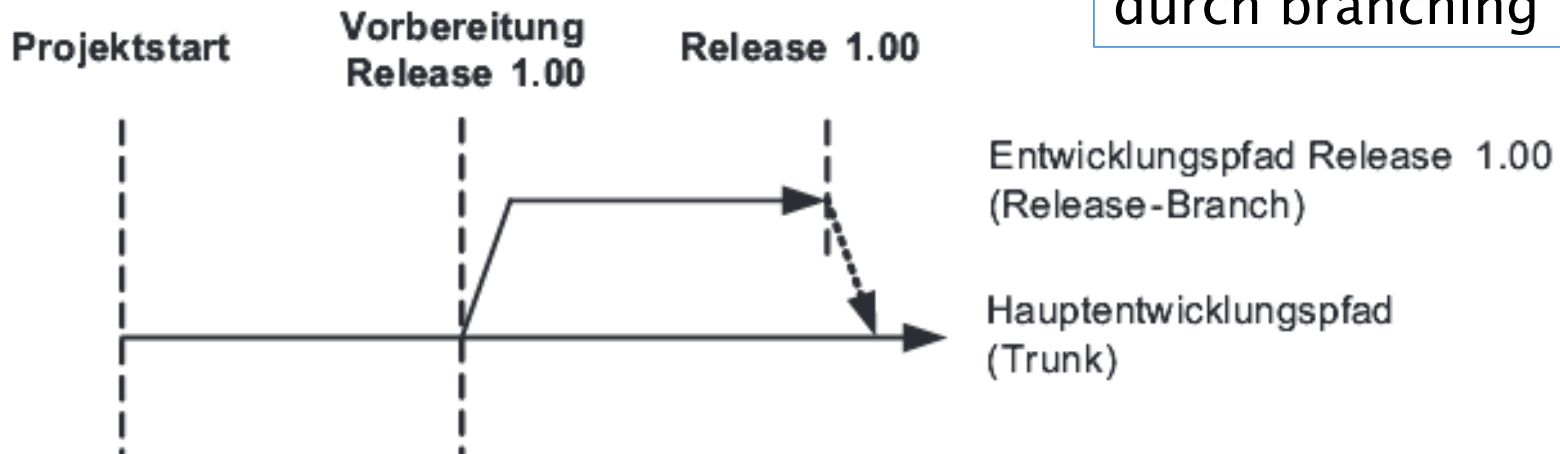
Branching und merging

- Was tun bei verschiedenen Produktlinien?
 - Was tun bei Feature Freeze für ein Release?
- ➔ Branching and merging

Branching und Merging



Code freeze im
linearen Entwicklungspfad



Weiterentwicklung
durch branching

Releases

- Konfigurationen, die veröffentlicht werden, d.h. an den Kunden, Abnehmer, Anwender zum Einsatz übergeben werden, nennt man *Releases*.
- Physisch ist ein Release meist eine Zusammenstellung der ausführbaren Elemente mit zugehöriger Dokumentation und der Definition des Installations-Vorganges (auch dies möglichst automatisiert).

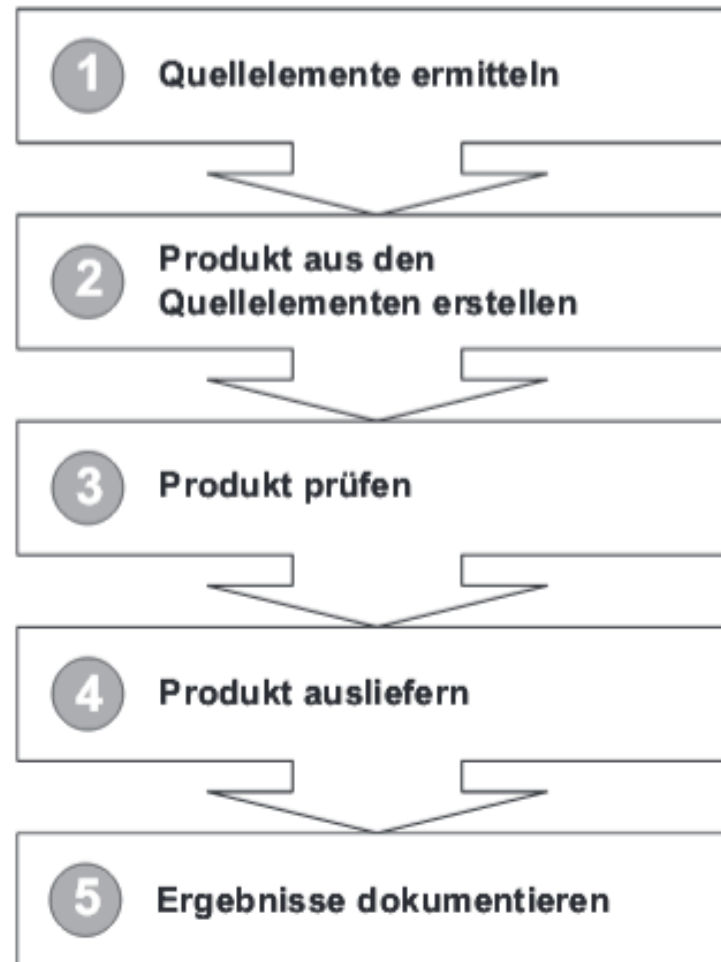
- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

Wer schlau ist, ist faul und macht's ...

... automatisch:

- Der Produktionsprozess
 - findet bei effizienter Entwicklung großer Systeme oft täglich statt,
 - dauert manuell bis zu vielen Stunden, manchmal Tage,
 - muss die Abhängigkeiten zwischen den Elementen beachten,
 - und muss deshalb automatisiert werden!
- Nur durch Automatisierung des Builds (Produktionsprozesses) lassen sich Fehler und kleine Nachlässigkeiten reproduzierbar erkennen und beseitigen.

Projektautomatisierung



▪ **Entwicklerbuild**

- Auf lokaler Umgebung
- Wichtig: schnell

▪ **Integrationsbuild**

- Regelmäßiger build auf einer Integrationsumgebung
- Incl. Der Qualitätssicherungsschritte

▪ **Releasebuild**

- Ähnlich Integrationsbuild, incl Setzen von Release Tag

Umsetzen des Build Prozesses

- Shell Skripte
- Ant
- Maven
- ...

- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

Außer bei der ersten Entwicklung eines SW-Systems werden die Änderungen an einem SW-System hauptsächlich durch Anforderungen der Umwelt veranlasst:

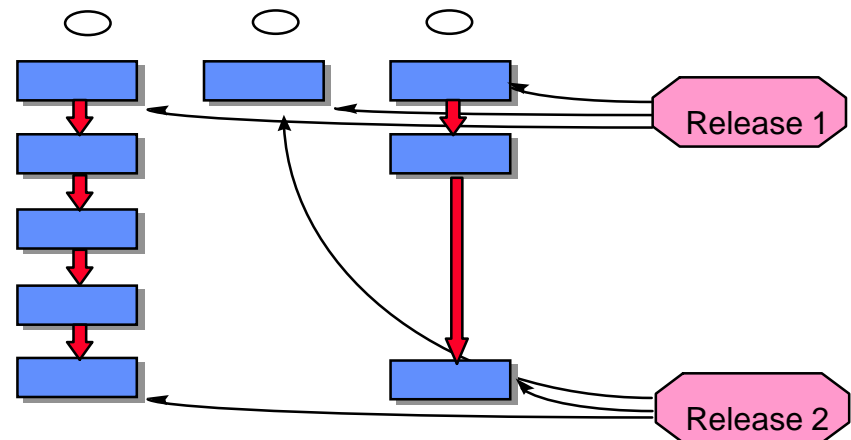
- Fehler treten auf und müssen behoben werden,
- neue oder geänderte Funktionalität wird gebraucht,
- technische Basissysteme, Schnittstellen ändern sich,
- ...

.. ist der geordnete steuernde Umgang mit Anforderungen der Umwelt an das System.

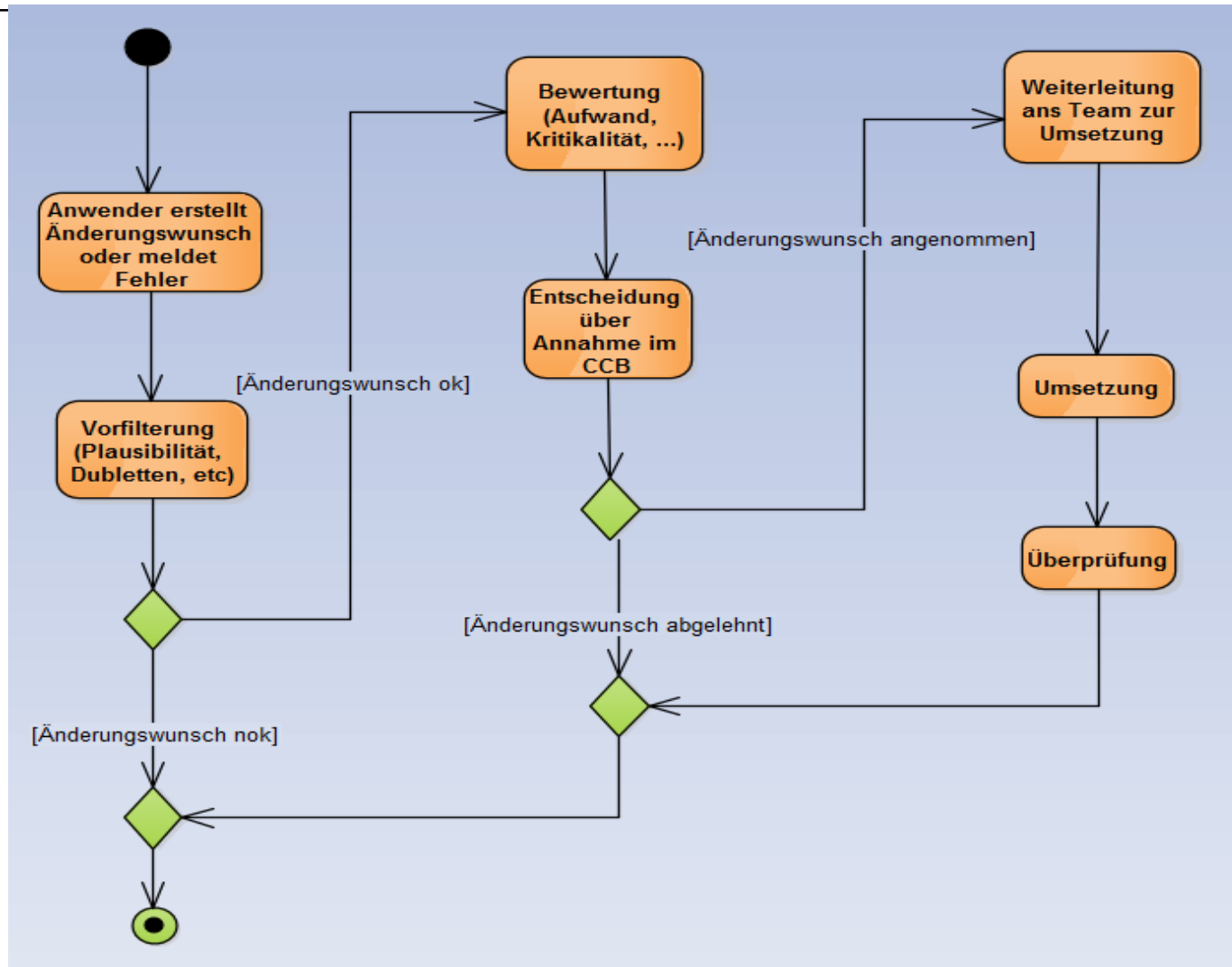
Aufgrund dieser Anforderungen kommt es zur Erzeugung neuer

- Versionen,
- Konfigurationen und
- Releases

des Systems.



Änderungsmgmt - Prozess



Daten eines CRs

- Id
- Name des Autors
- Datum der Erstellung
- Aktueller Status (vorgelegt, angenommen, umgesetzt, freigegeben, ...)
- Beschreibung
- Beschreibung der Auswirkungen, falls der CR nicht umgesetzt wird.
- Bewertung hinsichtlich Kritikalität und Aufwand
- Priorisierung durch das CCB
- Begründung der Entscheidung (insb. bei Ablehnung)

Verwaltung der CRs und Fehlermeldungen durch einen **Änderungsmanager**

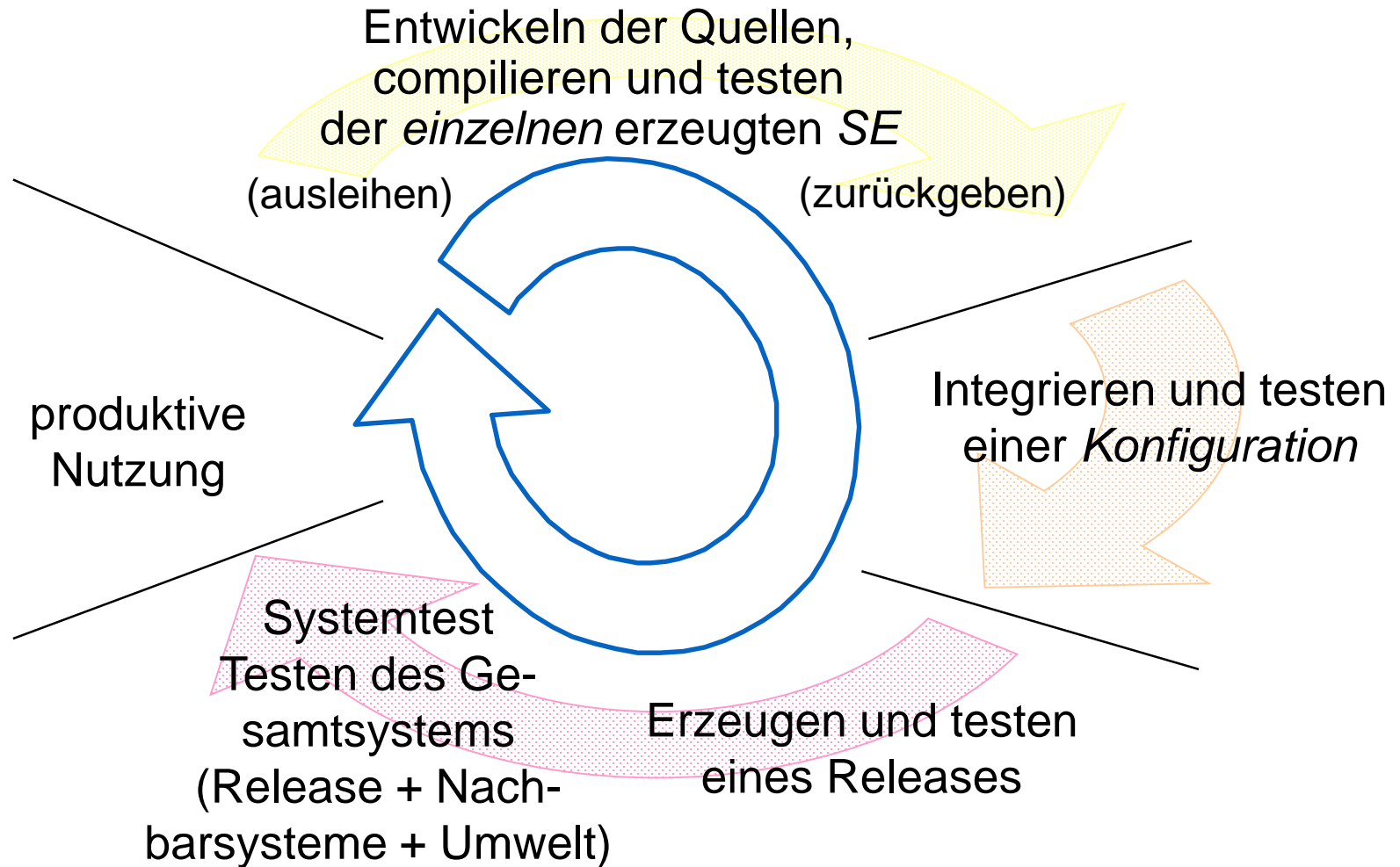
- Vorfilterung
- Bewertung (Kritikalität und Aufwand)

Change Control Board

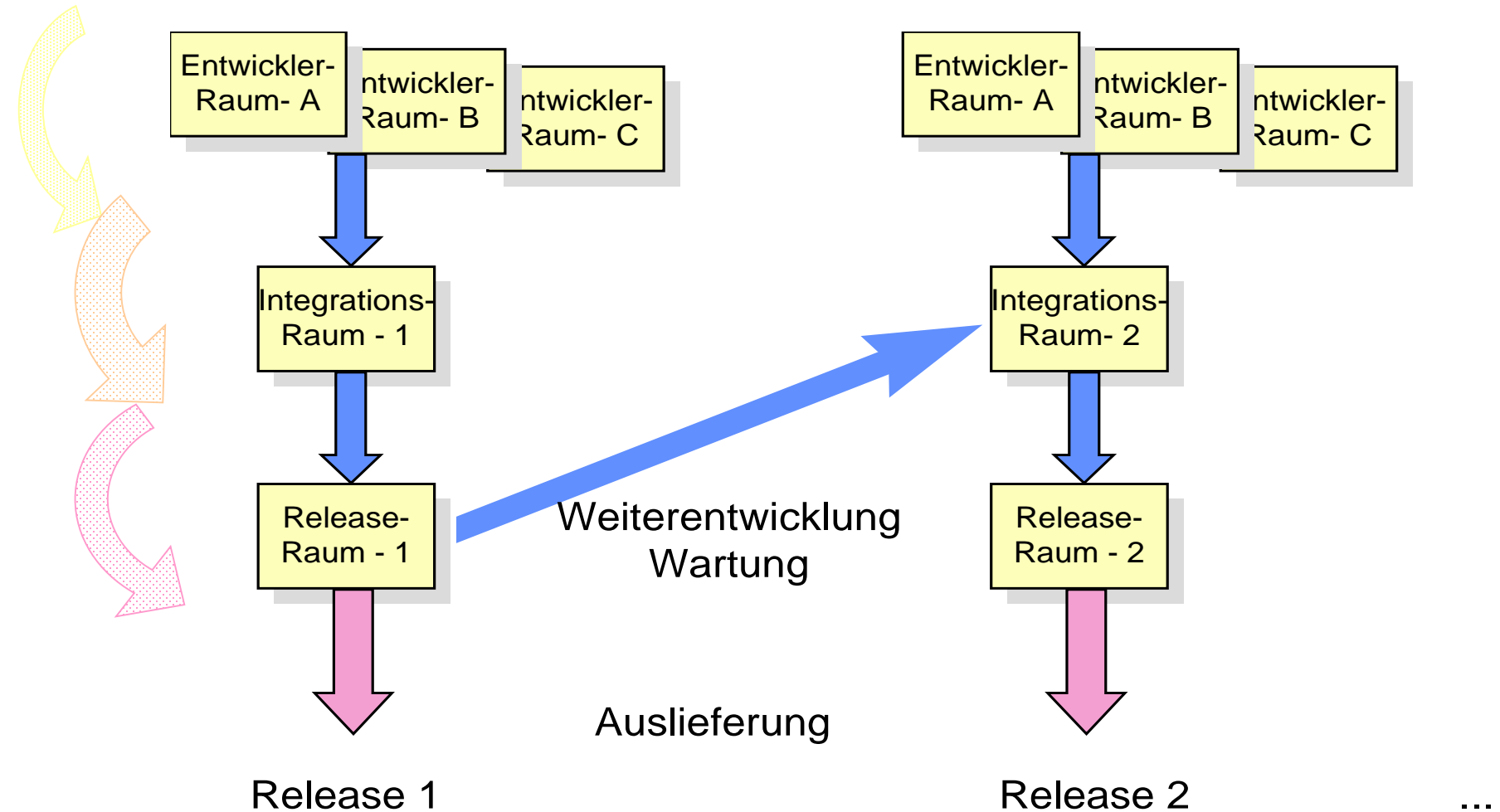
Entscheidung über Annahme oder Ablehnung des CRs/Fehlers

- Motivation
- Konfig Management – Überblick/Definition
- Aufgaben des Kernprozesses
 - Auswahl der Konfig Elemente
 - Erstellung des KM Handbuchs
 - Beschreibung der Konfig Elemente
 - Festlegung der Projektstruktur
 - Verwaltung der Konfig Elemente
 - Projektautomatisierung
 - Änderungs- und Fehlermanagement
- Workflow- und Raumkonzept
- Arbeiten mit speziellen Tools

Workflow und Raumkonzept



Umgebungs (Raum-) Konzept



Der Weg durch die Räume

- Die Übergänge von SW bzw. Konfigurationen in den nächsten Raum stellen wichtige Planungseinheiten für ein Entwicklungs- oder Wartungsprojekt dar.
- Hier spielen KM und QM (Qualitätsmanagement) zusammen: Für den Übergang müssen definierte Qualitätskriterien erfüllt sein.
- Der Übergang in den nächsten Raum ist mit Aktivitäten zur Dokumentation und des KM verbunden
 - z.B. Erstellung von Release-Notes
 - z.B. Archivierung eines Release

- Releasemanagement
- Audits
- Metriken
- Berichte