

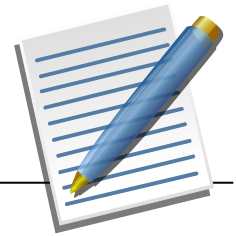


Software-Engineering für langlebige Systeme

VL1

- Softwareerosion
- Systemtypen
- Ziele:
 - Den Inhalt der Vorlesung kennen lernen.
 - Kennenlernen der grundlegenden Probleme durch Softwareerosion
 - Verstehen was langlebige Systeme sind.
 - Kennen lernen der Probleme von langlebigen Systemen.

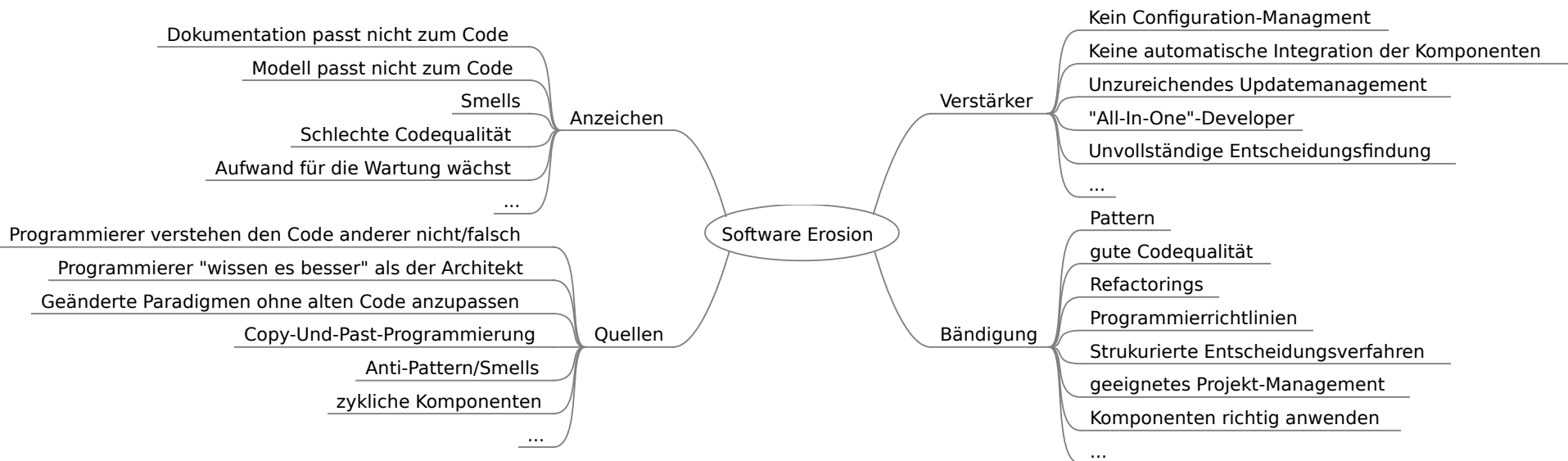
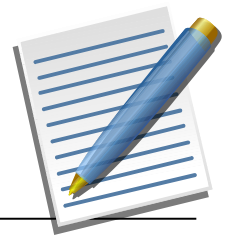
Softwareerosion

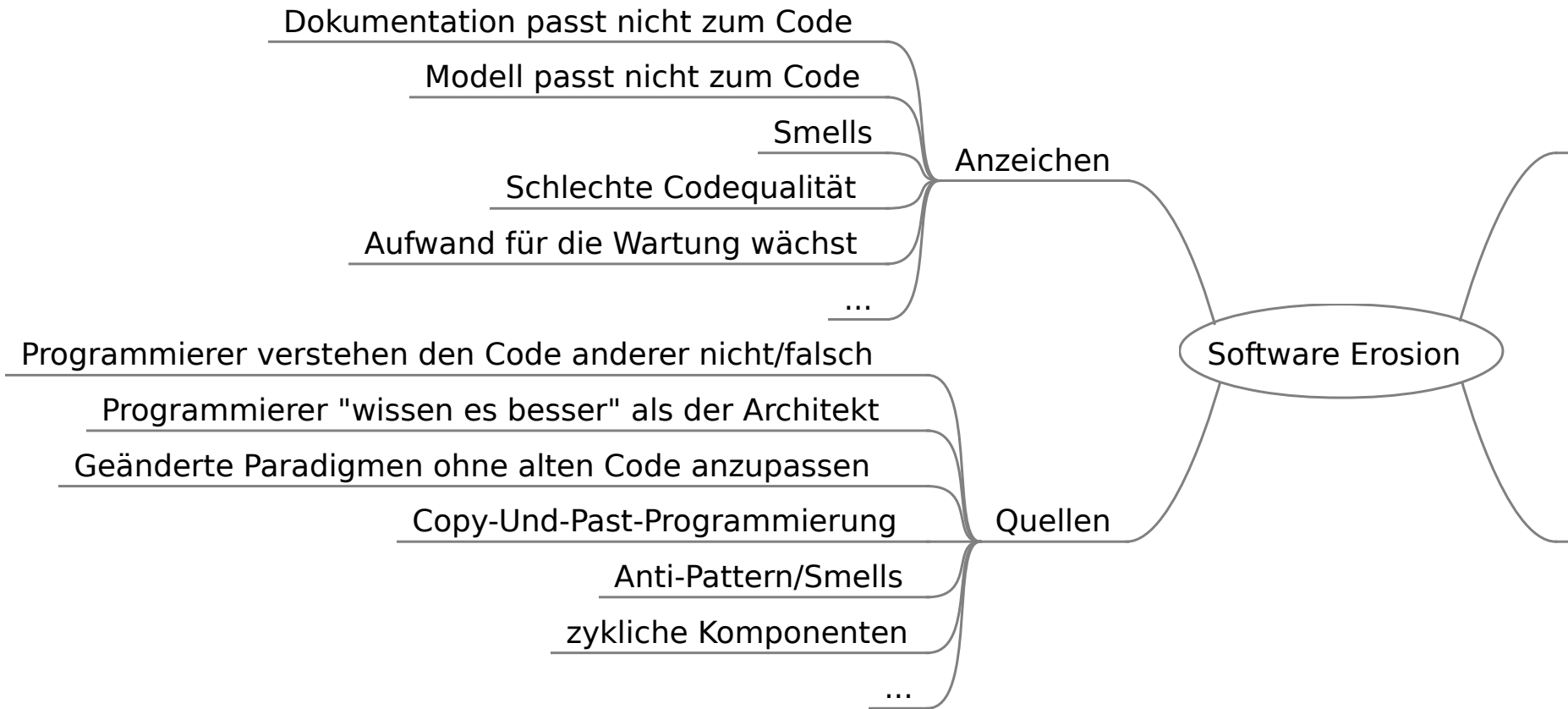
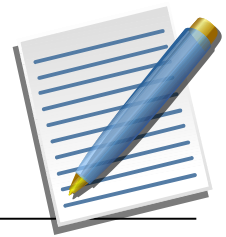


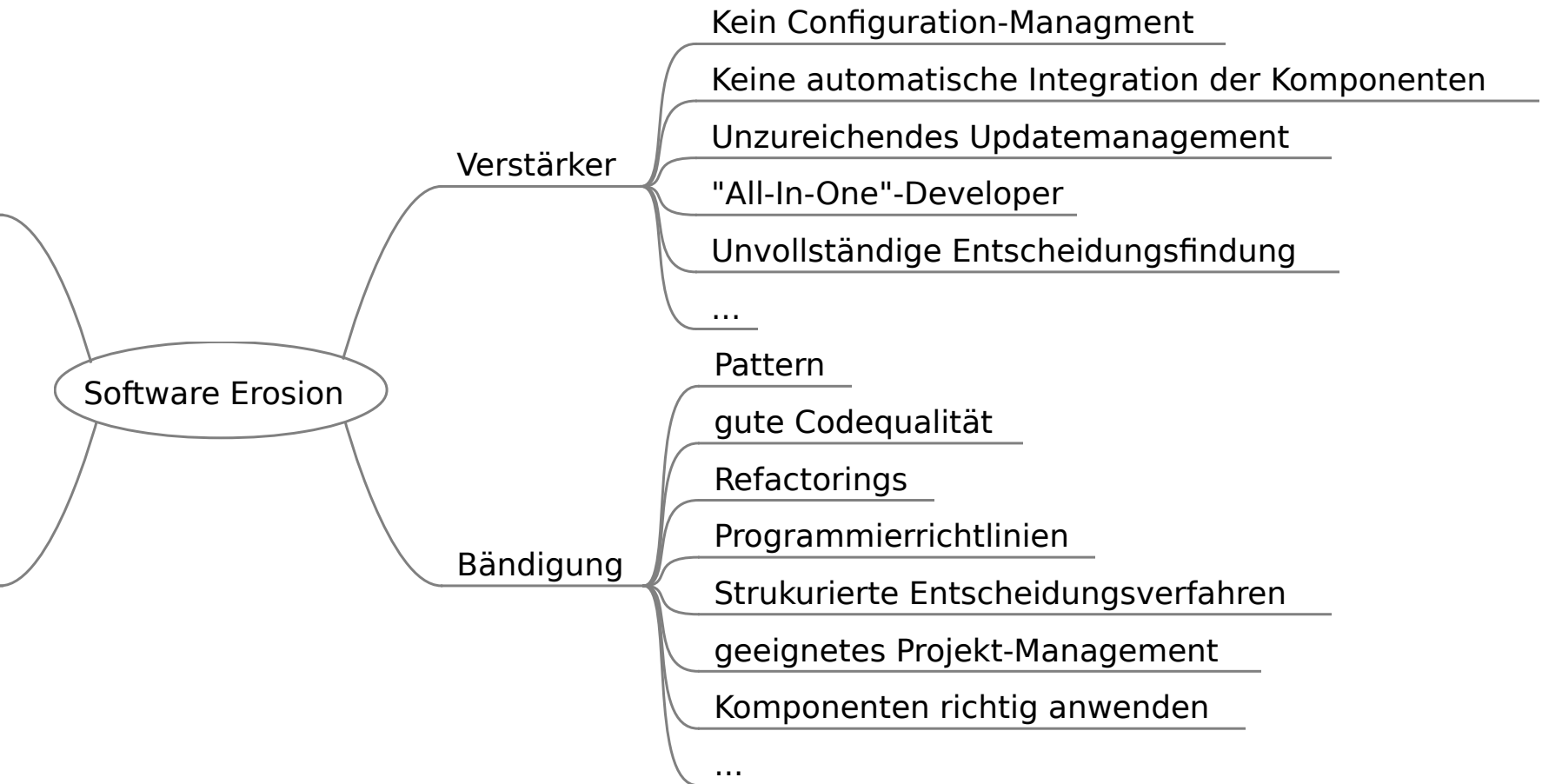
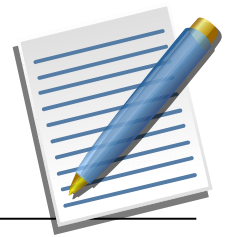
Softwareerosion

Softwareerosion ist der innere Strukturverlust von Software,

- der zu schlechter Wartbarkeit,
 - schlechter Anpassbarkeit,
 - schlechter Performance,
 - Häufung von Fehlern und
 - Häufung von Sicherheitsrisiken
- führt.







Ein Beispiel

<http://structure101.com/blog/2008/11/software-erosion-findbugs/>

Problem im „findbugs“-Beispiel

- Komponentenstruktur geht verloren
- Problem aus dem Code nicht leicht ersichtlich
- Wartung wird immer komplexer
 - Wenn eine Komponente geändert wird, müssen häufig auch andere Komponenten angepasst werden.
 - Wiederverwertung schwierig
 - Fehlereingrenzung schwierig

Wie feststellen?

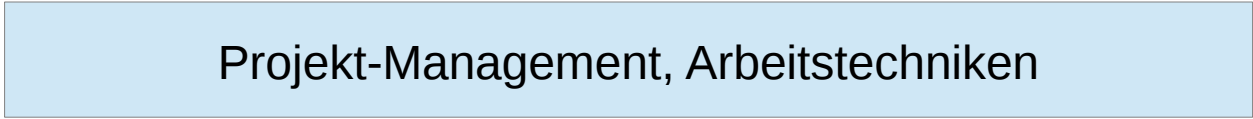
- Metriken
 - Lack of Cohesion of Methods
 - Dependency Structure Matrix
 - ...
- Strukturgraphen (wie im Beispiel)
- Direkte Code-Analyse

Wie beheben und vermeiden?

- Refactoring
- Pattern
- Evolutionsanalysen
- Co-Evolution
- Code-Guidlines
- Systemabstraktionsschichten
- Verbinden von Systemen
- Komponentendesign

Plan

Meta-Ebene



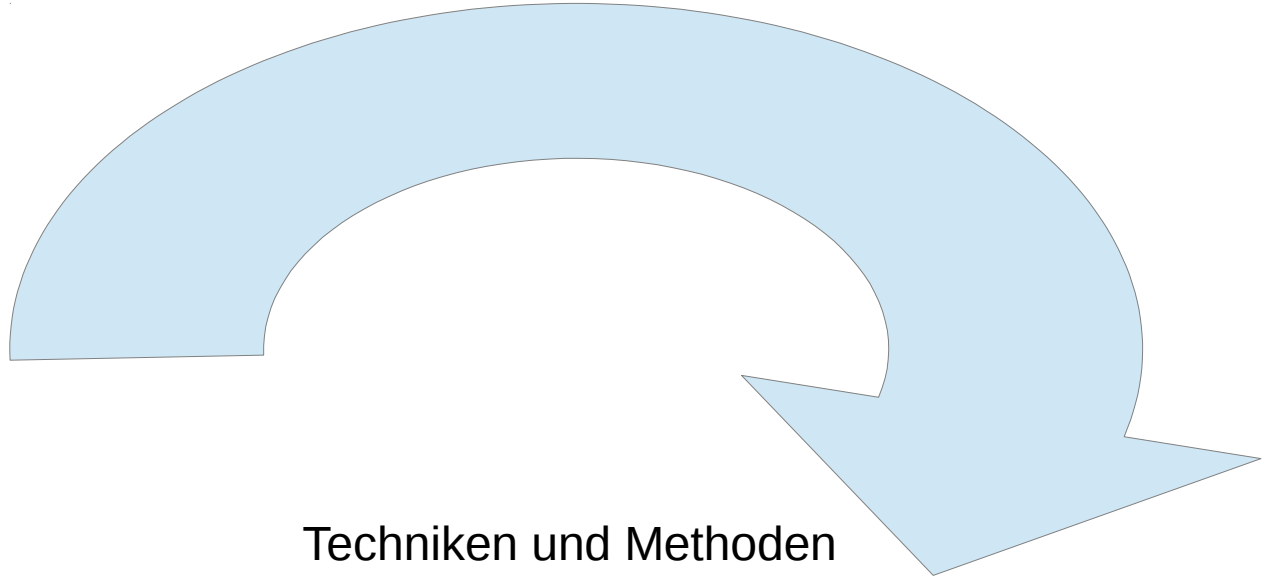
Projekt-Ebene

Requirments

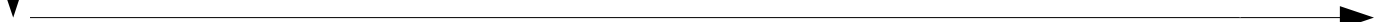
Design

Code

Test



Techniken und Methoden
zur Verlangsamung von Softwareerosion
und zur Softwarerestaurations

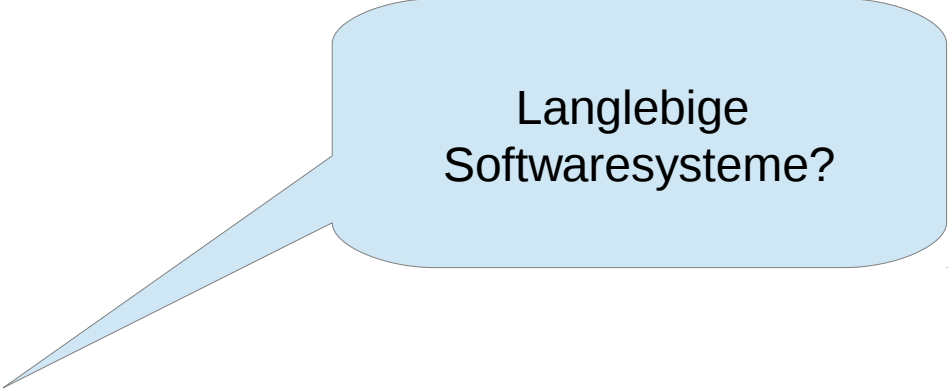


Semester-Verlauf

Softwaresysteme

- Betriebssysteme
- Eingebettete Systeme
- Datenbanksysteme
- Informationssysteme
-

- Definition/Abgrenzung
- Designeinflüsse/Anforderungen



Langlebige
Softwaresysteme?

Betriebssysteme

- Ein Betriebssystem ist eine Sammlung von Computerprogrammen, die die Systemressourcen eines Computers wie Arbeitsspeicher, Festplatten, Ein- und Ausgabegeräte verwaltet und diese Anwendungsprogrammen zur Verfügung stellt. Das Betriebssystem bildet dadurch die Schnittstelle zwischen den Hardwarekomponenten und der Anwendungssoftware des Benutzers.

Andrew S. Tanenbaum: Moderne Betriebssysteme. Pearson Studium, 3., aktualisierte Auflage, ISBN 978-3-8273-7342-7

Betriebssysteme - Designeinflüsse

- Hardware Abstraktion
 - Anwendungen möglichst unabhängig von der Hardware sein.
- Sicherheit
 - Isolation von Daten, Prozessen und Nutzern
 - Authentisierung, Rechtemanagement
- Stabil und Robust
- Ressourcenverwaltung und Konfliktauflösung
 - Drucker, Festplatten etc.

Eingebettete Systeme

- Definition 1:
Ein Rechensystem, das eine Funktion oder einen Funktionsbereich überwacht: wissenschaftliche, technische und/oder industrielle Steuerung
- Definition 2:
Jedes in einem Produkt versteckte Rechensystem, das selbst kein Rechner ist : Konsumgüter, Haushaltegeräte, Fahr- und Flugzeuge, . . . , Waffen
- Definition 3:
Ein zur Durchführung einer spezifischen Aufgabe entwickeltes Rechensystem, wenn auch mit Auswahl und Optionen

Datenbanken

- Geschwindigkeit
- Transaktionssicherheit
- Datensicherung
- Abfrageoptimierung

Eingebettete Systeme - Designeinflüsse

- Integration
 - Mikrocontroller
 - Peripheriebausteine, ...
- Begrenzte Ressourcen
 - Wenig Speicher
 - Wenige Anschlüsse, ...
- Stromverbrauch
- Echtzeitanforderungen
 - Hohe Verfügbarkeit
 - Definierte Antwortzeiten
- Betriebssicherheit
- Stückpreis
- Entwicklungsumgebung
 - Entwicklungsarchitektur ungleich Laufzeitarchitektur

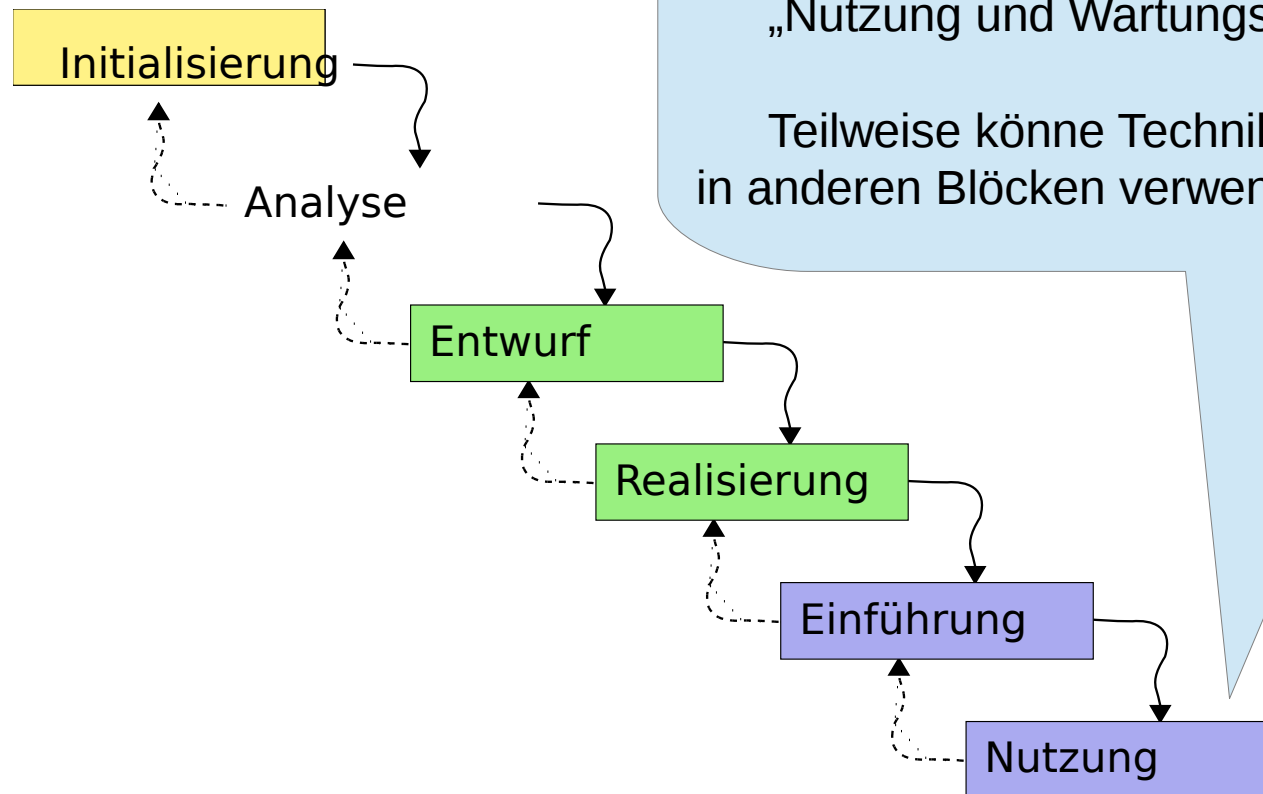
Langlebige Softwaresysteme

- Langlebigkeit ist eine Eigenschaft in verschiedenen Systemklassen
 - Betriebssysteme
 - Steuerungssoftware
- Meist nicht langlebige Systeme
 - Simulations-Prototypen

Langlebige Softwaresysteme

- Langlebige Softwaresysteme sind Softwaresysteme, die solange eingesetzt werden, dass diese regelmäßig und über einen langen Zeitraum an sich ändernden
 - Hardware,
 - Softwareumgebungen (einschl. BS und Libs) und
 - Benutzungsanforderungenangepasst werden müssen und über diesen Zeitraum gefundene Fehler gefixt werden müsse.

Wasserfall-Modell



Die meisten Probleme, die wir betrachten sind im „Nutzung und Wartungs“-Block.

Teilweise könne Techniken auch in anderen Blöcken verwendet werden.

Nutzungs- und Wartungsphase - Softwareänderungen

- Bugfix
 - Beseitigt **nicht-kritische** Fehler
 - häufig kleine Änderung
 - **zügige** Auslieferung
- Hotfix
 - Beseitigt **kritische** Fehler
 - **Sofortige** Auslieferung
- Update
 - Hinzufügen und Ändern von Features/ Systemumgebungsunterstützung
 - Häufig definierte Zyklen, meist mittel- bis langfristig
- Relaunch
 - Größere Umbauten/ Featureänderungen/ Systemumgebungsänderungen

Fragen zur Inhaltsübersicht/Einleitung?