



University
of Basel

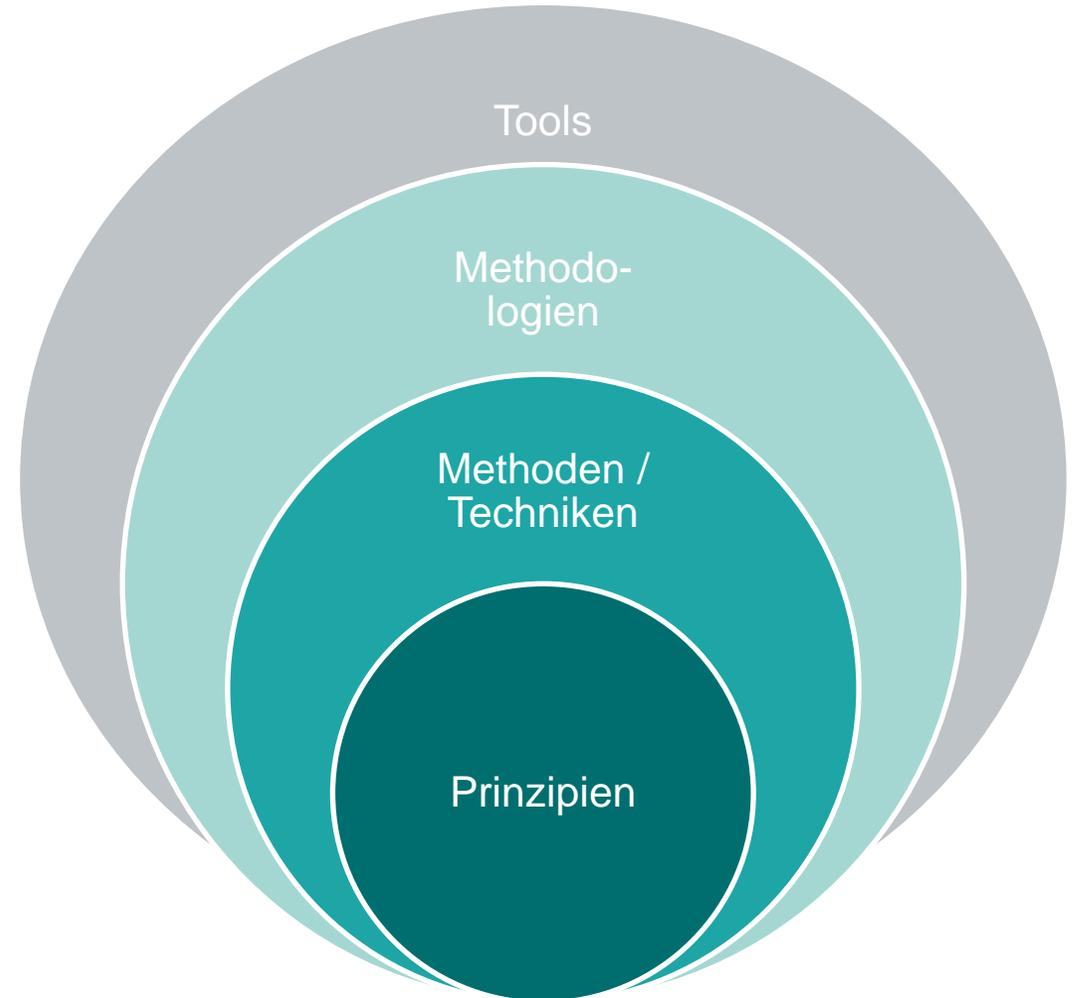
Software Engineering

Marcel Lüthi, Universität Basel

Prinzipien des Software Engineerings

Prinzipien des Software Engineerings

- Technologien und Werkzeuge der Informatik entwickeln sich rasant.
- Grundlegende Prinzipien sind seit vielen Jahrzehnten stabil.



Eigenschaft guter Prinzipien

Gute Prinzipien sollten abstrakt und falsifizierbar sein.

Beispiele

- Benutze JUnit um Tests zu schreiben 
 - Schreibe qualitativ gute Software 
 - Schreibe Tests immer zuerst 
-

Einige wichtige Prinzipien

- Genauigkeit und formales Vorgehen
 - Trennung der Verantwortlichkeiten (Separations of concerns)
 - Modularität
 - Abstraktion
 - Voraussehen von Veränderungen (Design for change)
 - Allgemeinheit
 - Schrittweises Entwickeln (Incrementality)
-

Trennung der Verantwortlichkeiten

- Gesamtproblem wird in kleinere Teilprobleme mit eigenen Verantwortlichkeiten zerlegt.
 - Reduziert Komplexität jeder Task
 - Aufgaben und Verantwortlichkeiten können verteilt und parallelisiert werden

Produkt

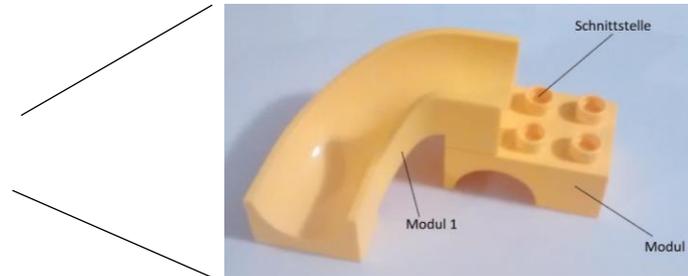
- Businesslogik
- Infrastruktur (Datenhaltung Logging, etc)
- Benutzeroberfläche
- ...

Prozess

- Separiere Entwicklung und Testing
 - Unterschiedliche Teams?
- Unterschiedliche Teammitglieder haben unterschiedliche Rollen (Management / Entwicklung / Quality Assurance)

Modularität

- Komplexes System wird in kleine Teile zerlegt
- Wichtigstes Prinzip in der Softwareentwicklung
- Reduziert Komplexität
 - Erlaubt Trennung der Verantwortlichkeiten



Abstraktion

- Wichtige Aspekte eines Problems werden identifiziert
- Details werden ignoriert
- Spezialfall von "Trennung der Verantwortlichkeiten"
 - wichtig (Funktion)
 - unwichtig (Detail)



Abstraktion



Implementation

Beispiele:

- Programmiersprache (Java)
 - OutputStream
- Betriebssystem
 - Dateien
 - Netzwerk Sockets
- Hardware:
 - Blöcke und Adressen