

2. Übungsblatt

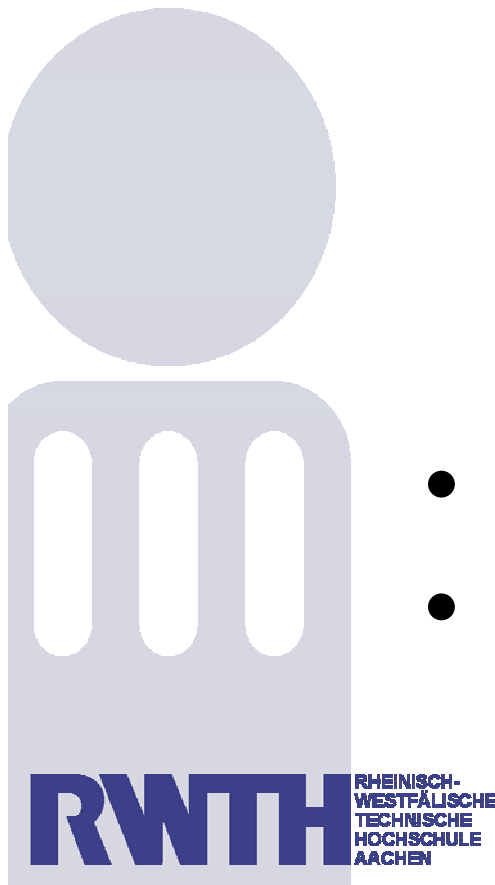
Änderungen am ersten Entwurf und Entwurfsparadigmen

4. Mai 2006

Dipl.-Inform. Christian Fuß

Änderung der Anforderungen

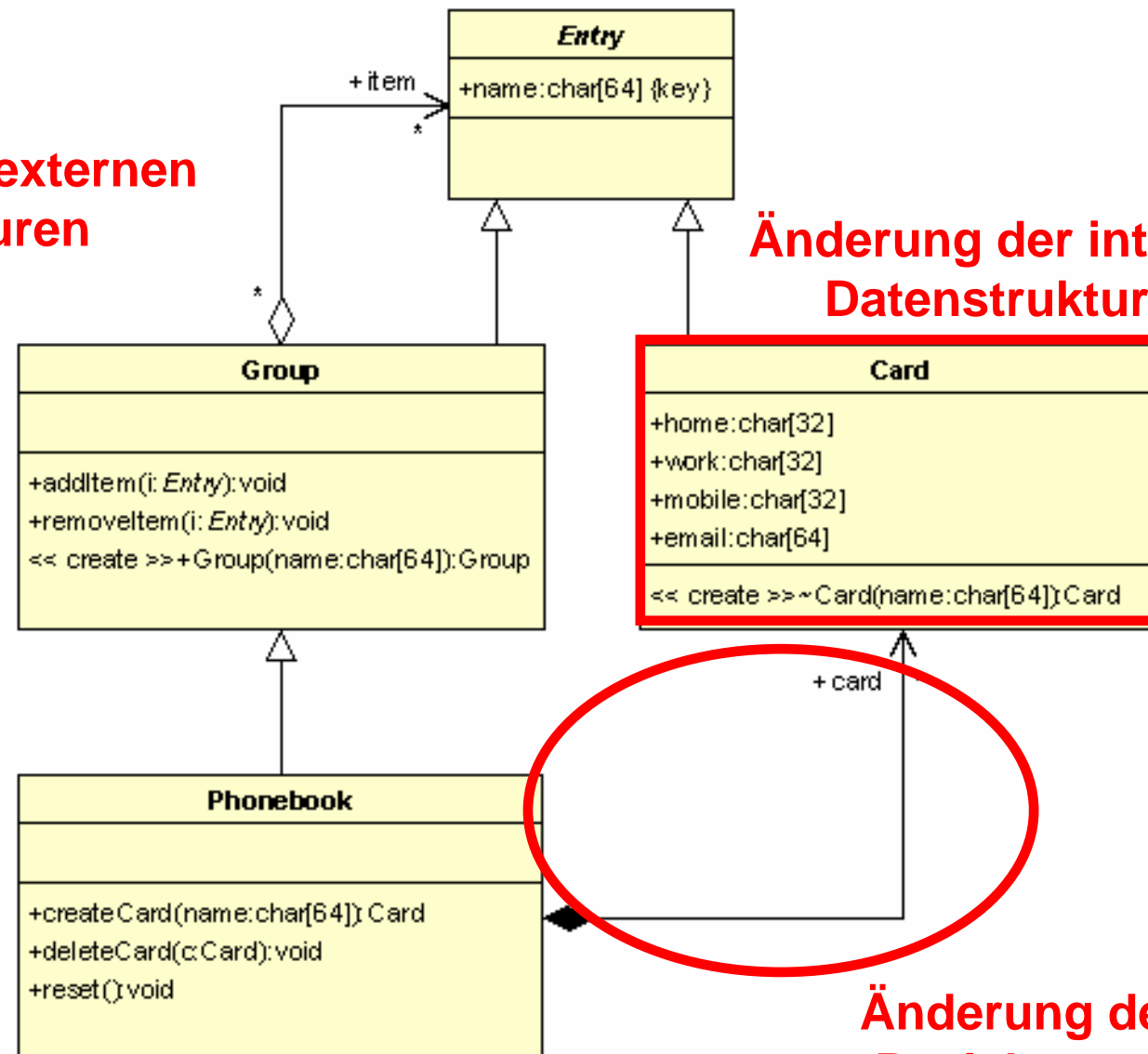
- Bedienung
 - Änderung der Anwendungsfälle
 - Änderung der Aktivitätsdiagramme
 - Änderung des Controllers
 - Änderung der Bedienoberfläche
 - Änderung der Views
- Datenstrukturen (Funktionalität)
- Hardware



Änderung der Datenstruktur

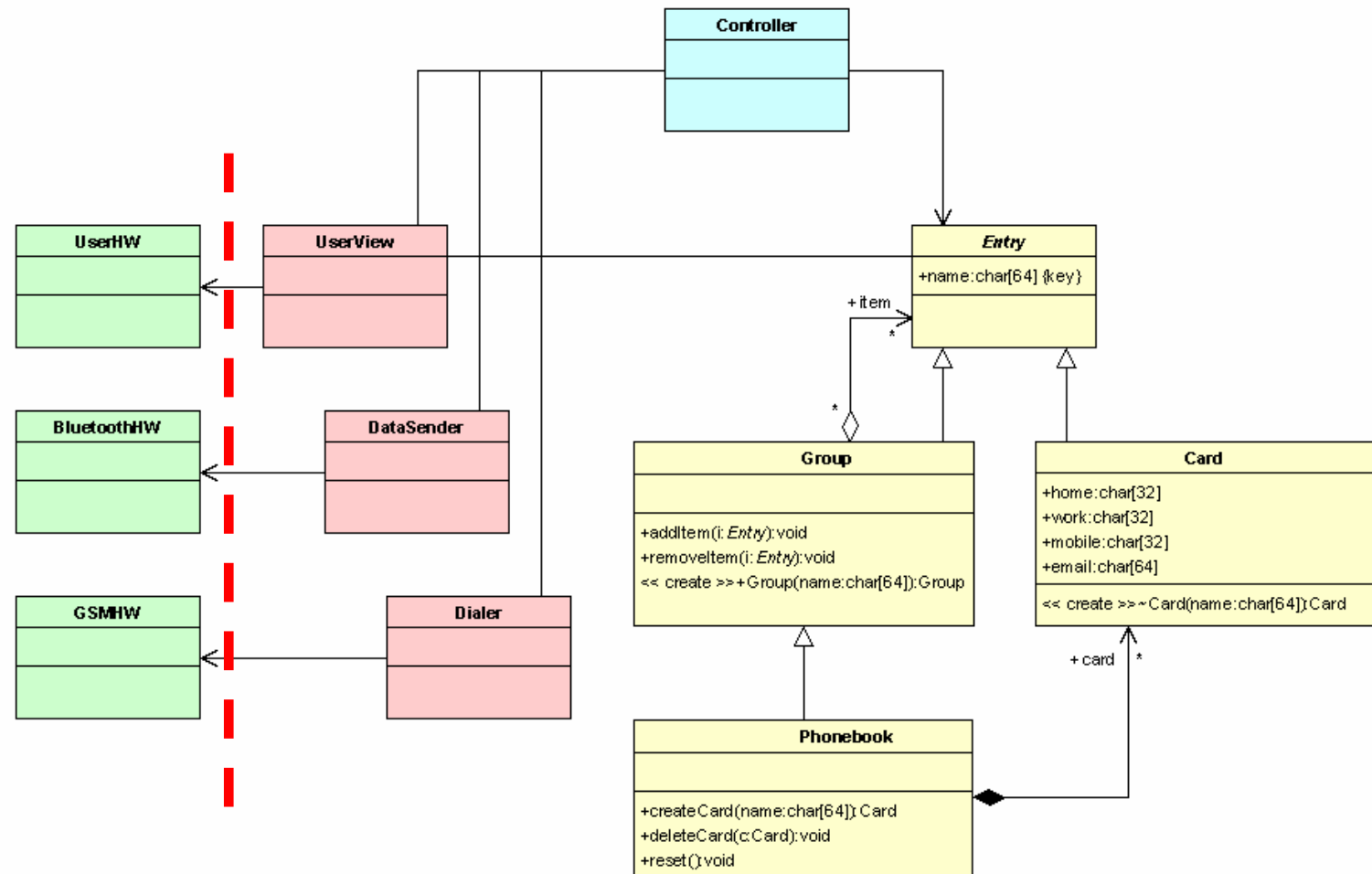
Erweiterung der externen
Datenstrukturen

Änderung der internen
Datenstrukturen

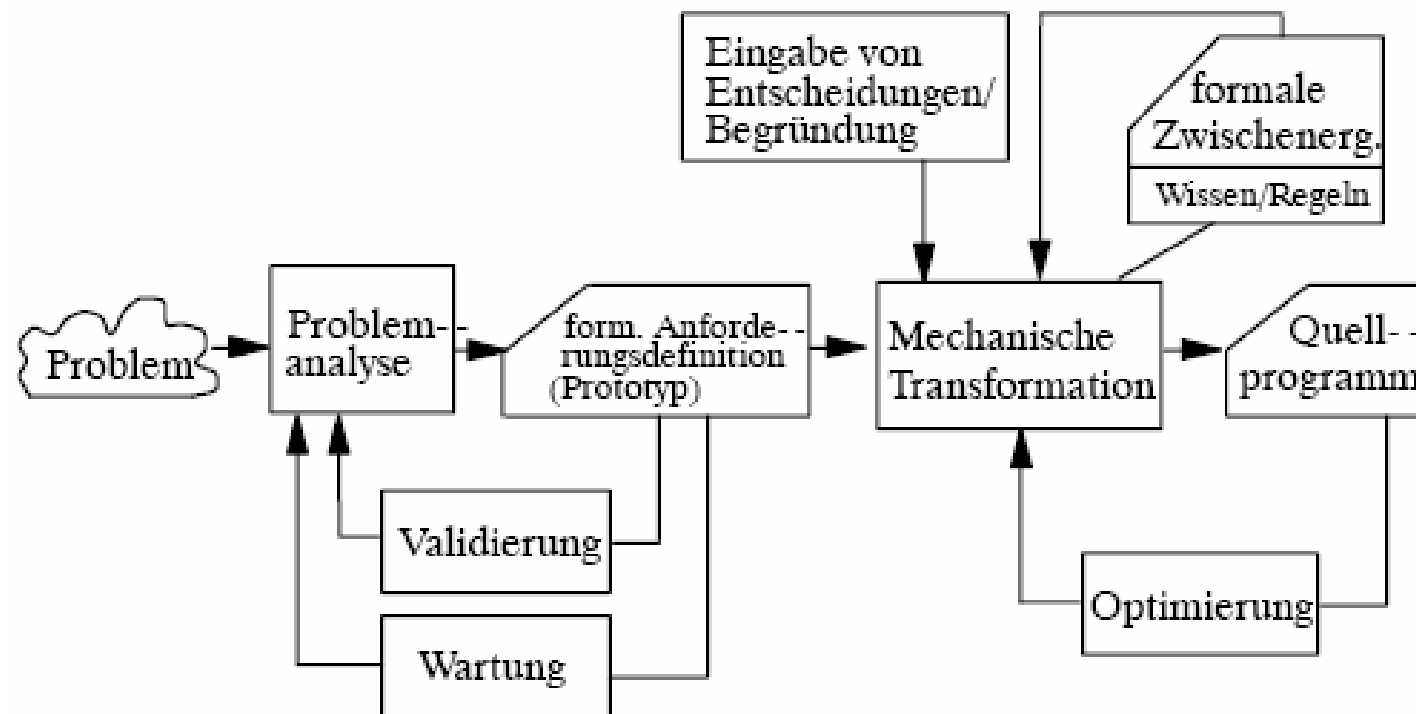


Änderung der
Beziehungen

Änderung der Hardware



Kontinuierliches Paradigma



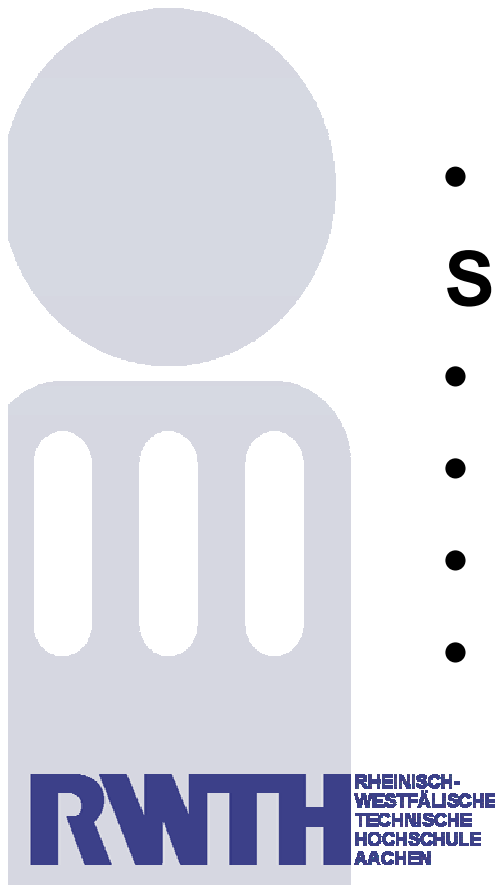
Formale Anforderungsdefinition

Schwierig zu formalisieren

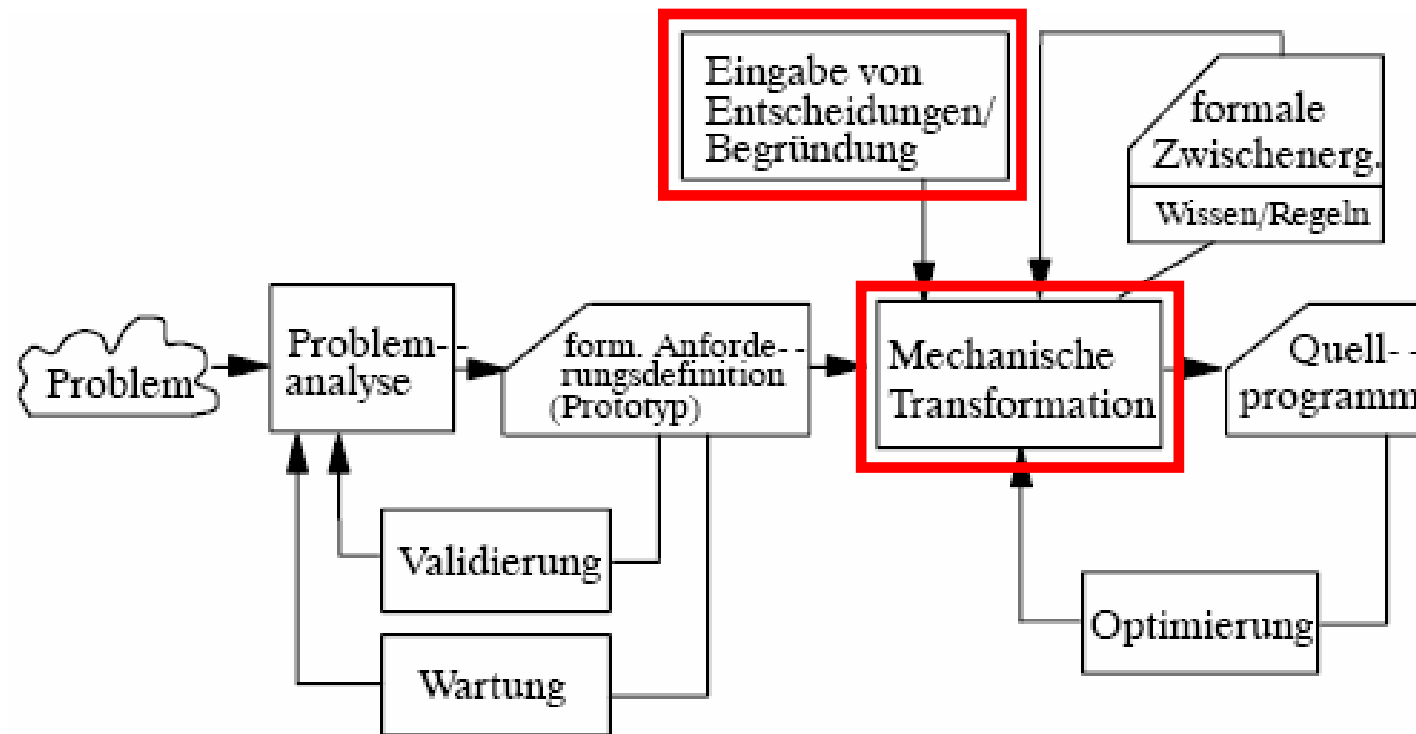
- Weiche Anforderungen
 - Wartbarkeit
 - Erweiterbarkeit
 - Nutzbarkeit, Nutzerfreundlichkeit
- Komplexes Zusammenspiel von Teilsystemen

Schwierigkeiten durch Formalisierung

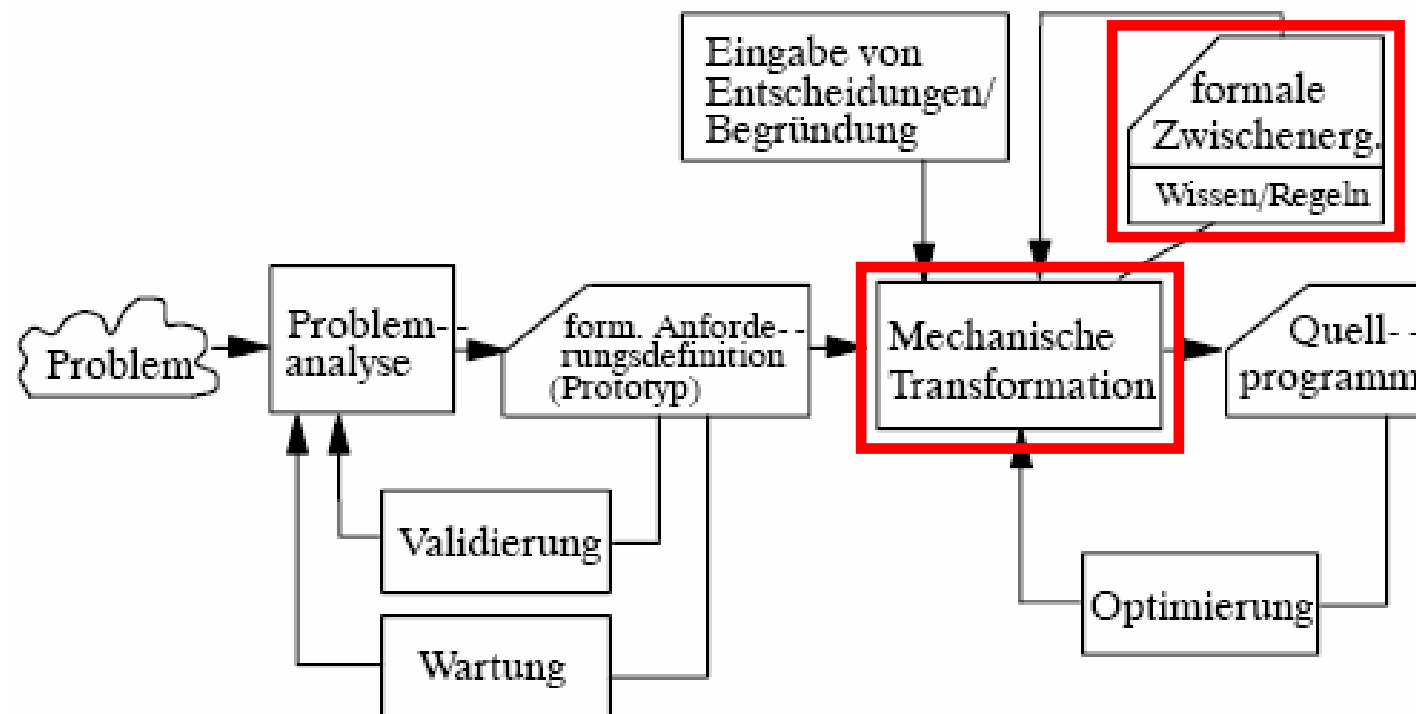
- Tiefe Durchdringung des Problems nötig
- Oft Realisierungsniveau nötig
- Formulierung oft durch Domänenexperten
- Domänenabhängige Sprache nötig



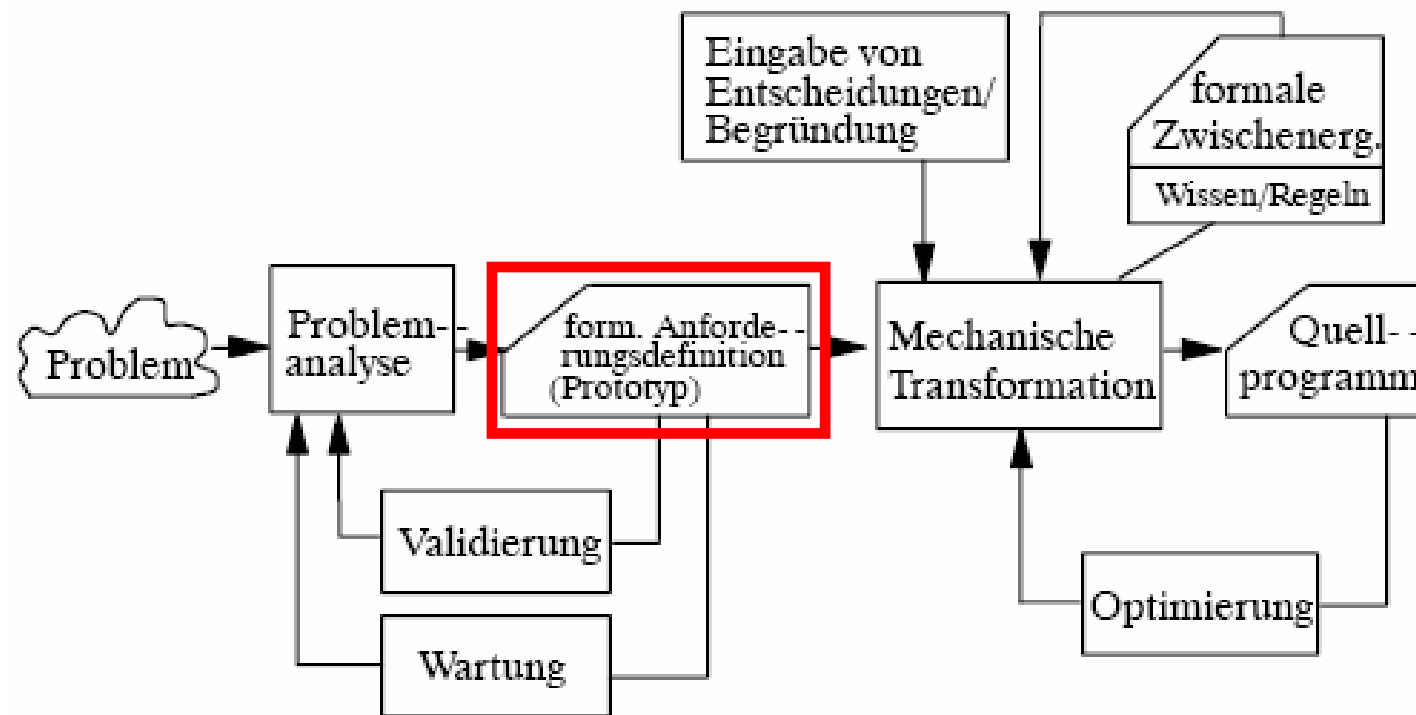
Programmieren ist Transformieren



Programmieren ist Erstellung von Regeln



Programmieren ist Spezifizieren



Domänen mit High-Level-Sprachen

- Compilerbau
- Mathematische Programme
- Simulationen
- Datenflussorientierte Programme
(z.B. Filterbau für Ton und Bild)
- Graphische Benutzeroberflächen
- Model Driven Architecture (QVT)
- Graphtransformationssysteme
(PROGRES, FUJABA)



Top-Down-Entwurf

- Verfeinerung von groben Bausteinen
- nur Modulebene ohne Implementierung, aber Realisierungsüberlegungen nötig
- Lokalitätsprinzip
- oft baumartige Architekturen
- viele funktionale Bausteine
- allgemeine Bausteine werden oft nicht erkannt
- Änderungen später schwierig



Bottom-Up-Entwurf

- Zusammensetzen von Bausteinen
- allgemeine Bausteine früh definieren
- Modellierung mit allgemeiner Benutzbarkeit
- Datenabstraktionsmodule bevorzugt
- besser änderbar
- Gefahr der frühzeitigen Detaillierung

