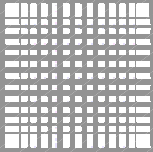


hochschule mannheim

## **Software-Entwicklung in Theorie und Praxis**

Prof. Dr. Peter Knauber

Informatik begeistert – Fachhochschule Heidelberg, 19. Mai 2006



# Vorgehensmodelle

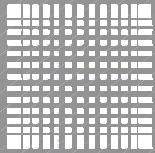
Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

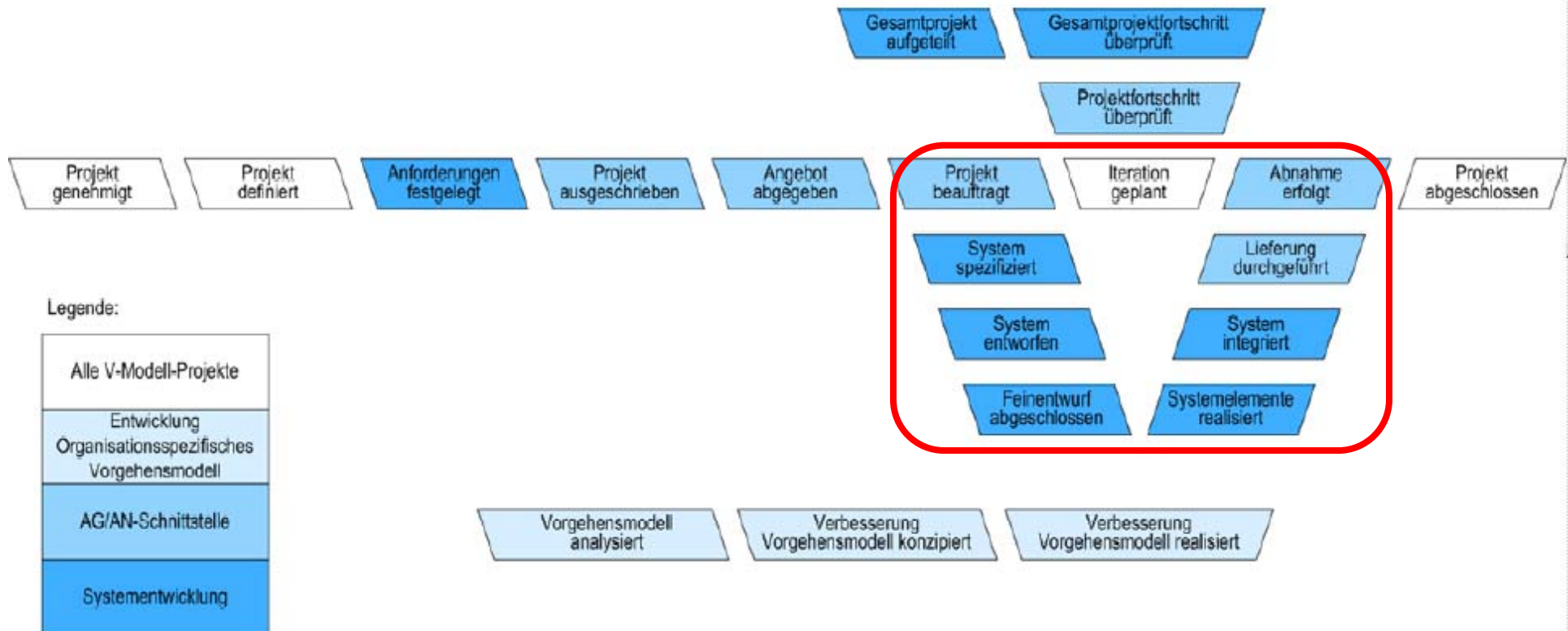
In jedem Strafraum 11 m vor der Torlinie eine strafstoßmarke als schwarze Linie anzubringen.

[FIFA Regelwerk]

**Alle Phasen vermischt, frühe Phasen gekürzt**

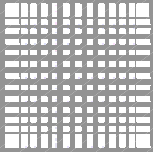


# Bewusste Phasentrennung: zum Beispiel das V-Modell XT

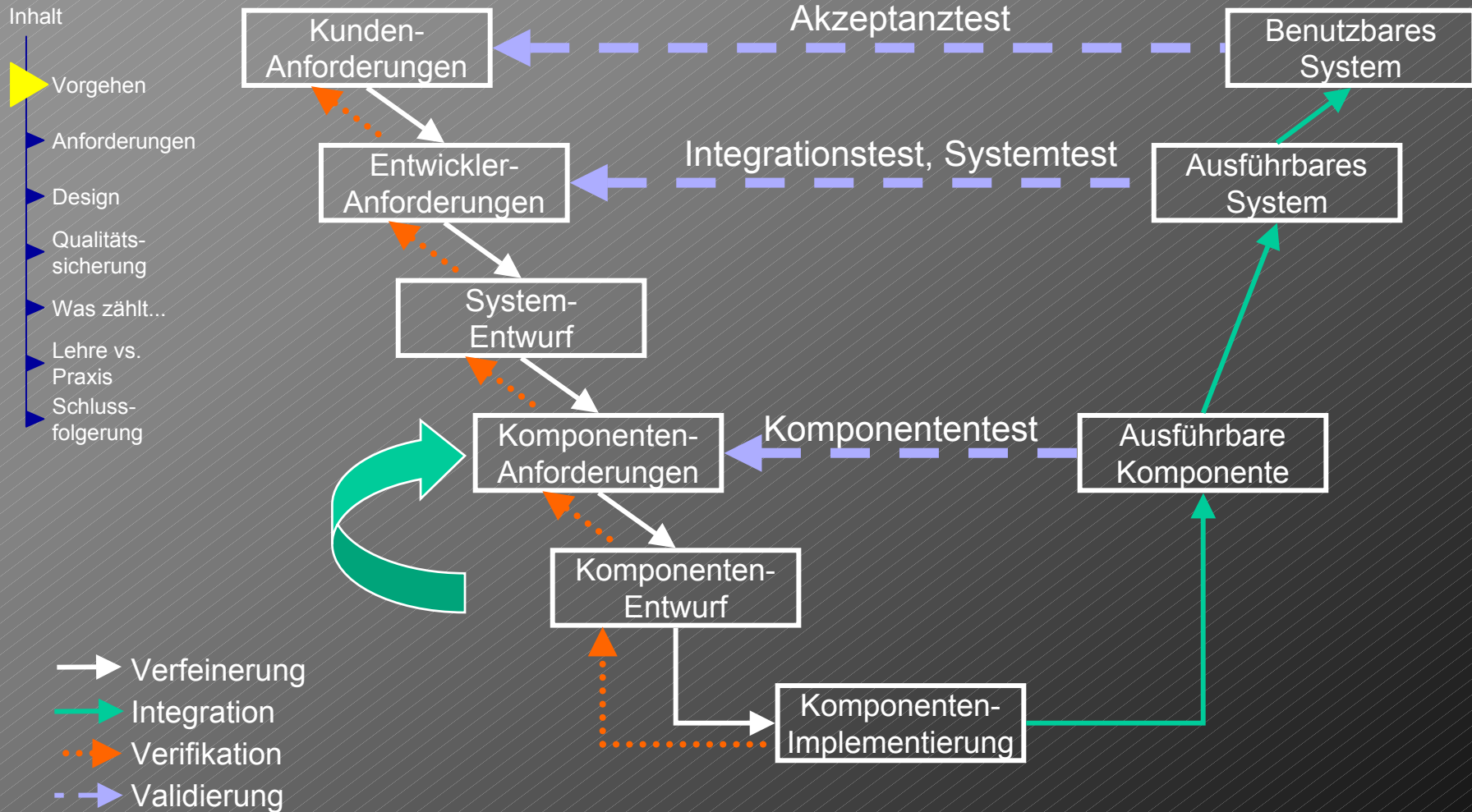


In der Regel wird ein neues V-Modell nach einer Katastrophe mit dem jeweils vorhergehenden V-Modell entwickelt.

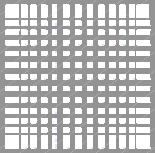
[Wikipedia]



# Entwicklungsaktivitäten des V-Modells



Folie 3

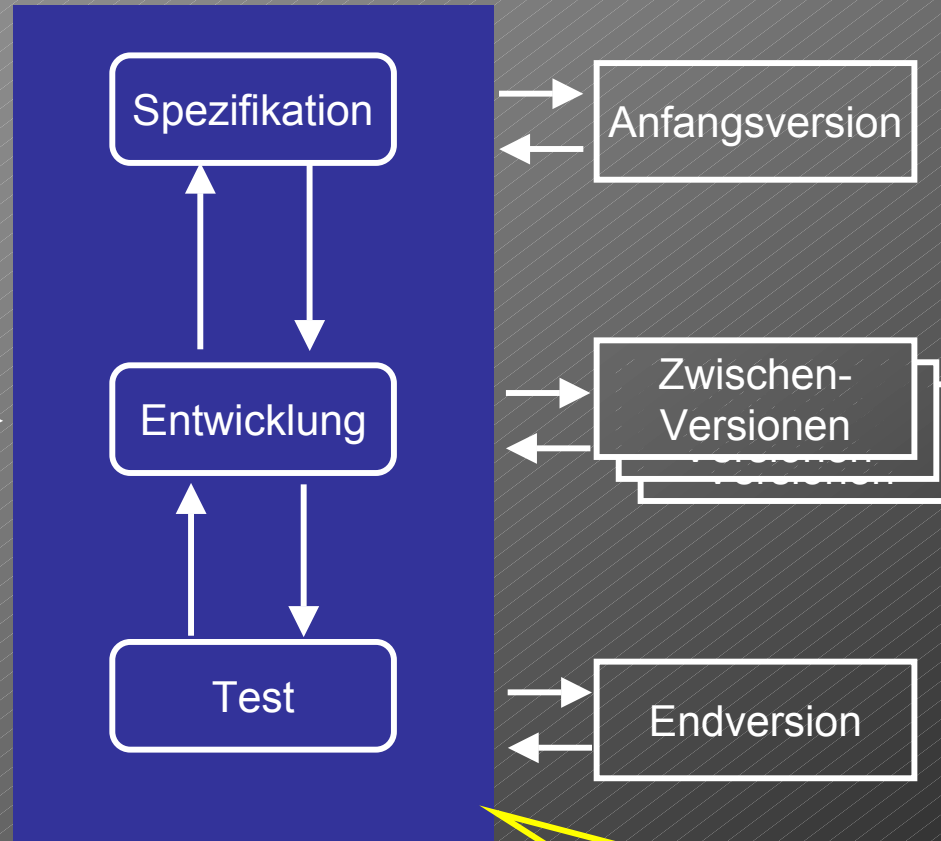


# Alle Phasen vermischt, frühe Phasen gekürzt: evolutionäre Entwicklung

Inhalt

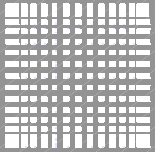
- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

Grobe Beschreibung



Gleichzeitig  
stattfindende Aktivitäten

1. Problem:  
Wir **behaupten** nur, die  
Phasen klar zu trennen!

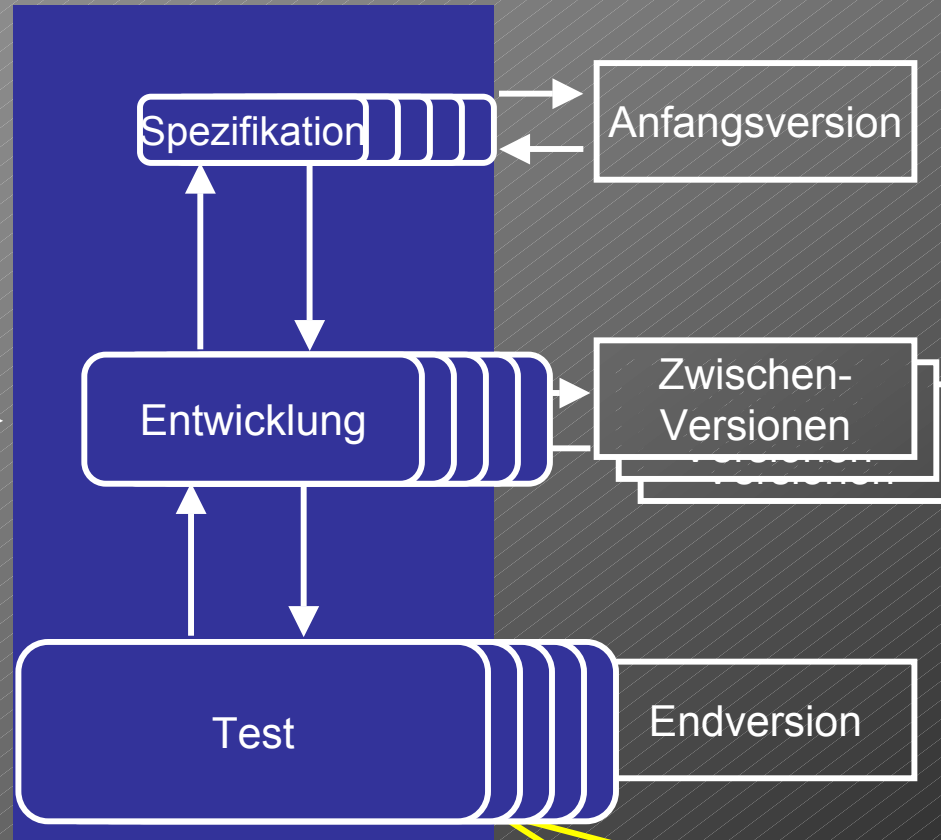


# Alle Phasen vermischt, frühe Phasen gekürzt: evolutionäre Entwicklung

Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

Grobe Beschreibung

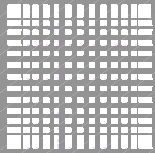


Gleichzeitig  
stattfindende Aktivitäten

1. Problem:  
Wir **behaupten** nur, die  
Phasen klar zu trennen!

# Konsequenzen

[Standish Group 1994]



Abgebrochene  
Projekte: 31,1%

Erfolgreich  
abgeschlossene  
Projekte: 16,2%

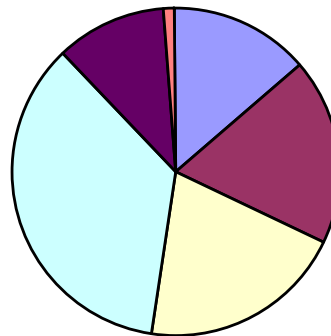
Abgeschlossene Projekte mit  
Zeit-, Kostenüberschreitung,  
weniger Funktionalität: 52,7%



Inhalt

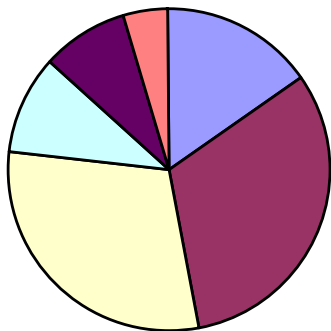
- Vorgehen
- Anforderung
- Design
- Qualitäts-sicherung
- Was zählt...
- Lehre vs. Praxis
- Schluss-folgerung

## Zeitüberschreitung



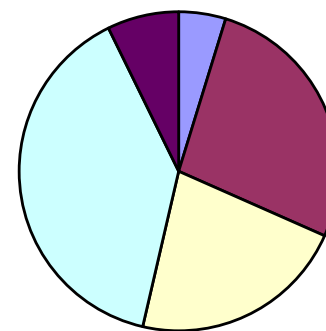
- unter 20%
- 21 - 50%
- 51 - 100%
- 101 - 200%
- 201 - 400%
- über 400%

## Kostenüberschreitung



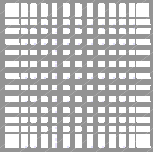
- unter 20%
- 21 - 50%
- 51 - 100%
- 101 - 200%
- 201 - 400%
- über 400%

## Weniger Funktionalität



- w eniger als 25%
- 25 - 49%
- 50 - 74%
- 75 - 99%
- 100%

über



# Anforderungen, Anforderungsanalyse

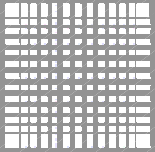
Inhalt

- ▶ Vorgehen
- ▶ **Anforderungen**
- ▶ Design
- ▶ Qualitätssicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schlussfolgerung

Die Te... sein.  
... werden, wenn  
... entsprechen.  
[FIFA Regelwerk]

**Anforderungen auf Zuruf, sporadische Dokumentation**





# Explizite Anforderungsdokumentation: zum Beispiel Pflichtenheft nach DIN 69905

Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

## **PFLICHTENHEFT**

### **1. Zielbestimmung**

- 1.1 **Muss-Kriterien**
- 1.2 **Wunsch-Kriterien**
- 1.3 **Abgrenzungskriterien**

### **2. Produkt-Einsatz**

- 2.1 **Anwendungsbereiche**
- 2.2 **Zielgruppen**
- 2.3 **Betriebsbedingungen**

### **3. Produkt-Bedingungen**

- 3.1 **Software**
- 3.2 **Hardware**
- 3.3 **Orgware**
- 3.4 **Produkt-Schnittstellen**

### **4. Produkt-Funktion**

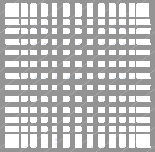
### **5. Produkt-Leistungen**

### **6. Benutzer-Schnittstelle**

### **7. Qualitäts-Zielbestimmung**

### **8. Globale Testfälle**

### **9. Ergänzungen**



# Explizite Anforderungsdokumentation: z. B. Pflichtenheft nach IEEE/ANSI 830-1993

Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

## 1. Einleitung

1. Ziel des Anforderungsdokuments
2. Anwendungsbereich des Produkts
3. Definitionen, Akronyme und Abkürzungen
4. Referenzen
5. Überblick über den Rest des Dokuments

## 2. Allgemeine Beschreibung

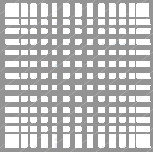
1. Produktperspektive
2. Produktfunktionen
3. Benutzercharakteristika
4. Allgemeine Beschränkungen
5. Voraussetzungen und Abhängigkeiten

## 3. Spezifische Anforderungen

Funktionale, nicht-funktionale und Schnittstellen-Anforderungen

## 4. Anhänge

## 5. Index



# Anforderungen auf Zuruf, sporadische Dokumentation

## Inhalt

Vorgehen

Anforderungen

Design

Qualitäts-  
sicherung

Was zählt...

Lehre vs.  
Praxis

Schluss-  
folgerung

## Anforderungen für "██████████"

### 1. Überblick

During the past years, opportunities for applying composite construction techniques have increased very much. The European market urgently needs powerful program systems for the *Calculation* of load-bearing elements. Composite Beams are the most important load-bearing elements for building construction. The existing ██████████ application must be adapted to the most recent evolutions in this area. The purpose of this project is to eliminate known problems or deficiencies and to increase its applicability. In addition to this, ██████████ has to be prepared for possible integration with other ██████████ products as well as with applications of other software vendors, e.g., to enable data input using a CAD system.

### 2. Features

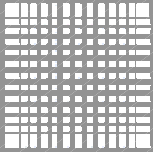
When making *Calculations* for composite Beams, different aspects have to be considered. It is necessary to generate and measure cross sections. After cross section measurements the definition of loads and load combinations is needed to support the *Calculation* of additional measures and deformations in the Structural System. After this, extensive proofs according to *DIN 18 800 Teil 5* or *Eurocode* have to be done. Engineers need the resulting data in the form of tables and graphic. For the construction in factories, blue prints are needed on which the production is based.

This Commonality Analysis will deal with the following questions:

- Which forms of cross sections are possible?
- What kinds of connections (Beam<->Column, Beam<->Beam) are allowed?
- Which standards are supported?
- Which interfaces to external applications are supported?
- Which kinds of load (dynamic, load pull) are possible?
- Which additional proofs (e.g., rest life span) are integrated?
- Which bearing requirements (e.g., gripping device for torsion springs) are allowed?
- Which FEM kernels (own, B&B) are integrated?
- Which combinations of load conditions (automatic, user defined, groups of load) are supported?

• Laden von FEM-Gen  
• Stücklisten

Formate:  
H. Kmo  
fragen!



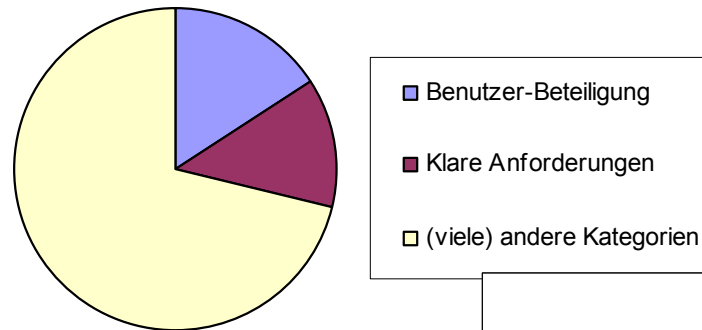
# Anforderungen: das Problem

[Standish Group 1994]

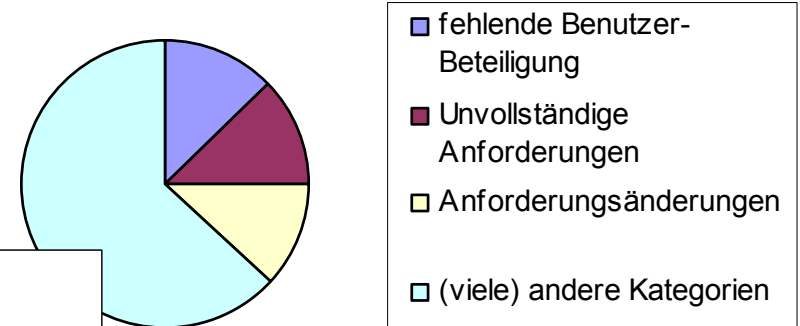
Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

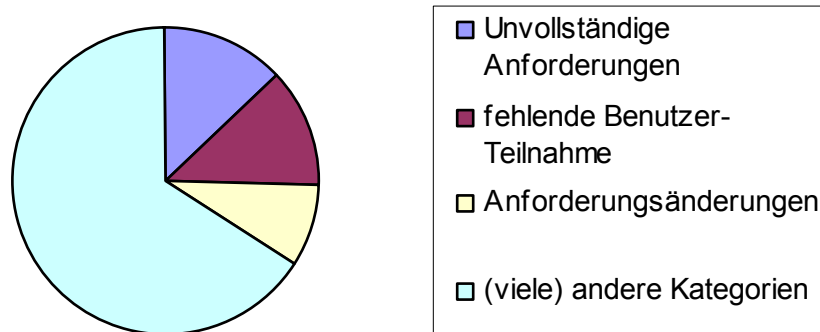
### Schlüsselfaktoren erfolgreicher Projekte



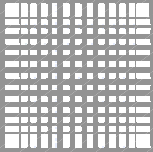
### Problemfaktoren von Projekten mit Überschreitungen



### Probleme bei abgebrochenen Projekten



Folie 11



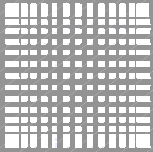
## Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ **Design**
- ▶ Qualitätssicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schlussfolgerung

**Systematisches Vorgehen, Verwendung der UML**

[Die Tore] bestehen in gleicher Höhe aus zwei vertikalen Säulen, die an den Enden der Unterseite verbunden sind. Die Unterkante der Säulen ist 2,44 m vom Boden entfernt.  
[FIFA Regelwerk]

**Ad-hoc Design, nur Klassendiagramme**



# Systematisches Vorgehen beim Design, Verwendung der UML

## Inhalt

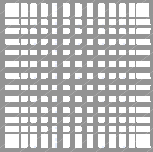
- ▶ Vorgehen
- ▶ Anforderungen
- ▶ **Design**
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

### Unified Modeling Language (UML) 2.1 - Notationsübersicht

The slide provides a detailed overview of UML 2.1 notations, organized into several categories:

- Use Case Diagrams:** Highlighted with a red border, showing actors, use cases, and relationships.
- Class Diagrams:** Shows classes, associations, inheritance, and generalization.
- Sequence Diagrams:** Illustrates the interaction between objects over time.
- Activity Diagrams:** Represents the flow of control and data within an object.
- State Machine Diagrams:** Shows the states of an object and the transitions between them.
- Package Diagrams:** Shows the organization of models into packages.

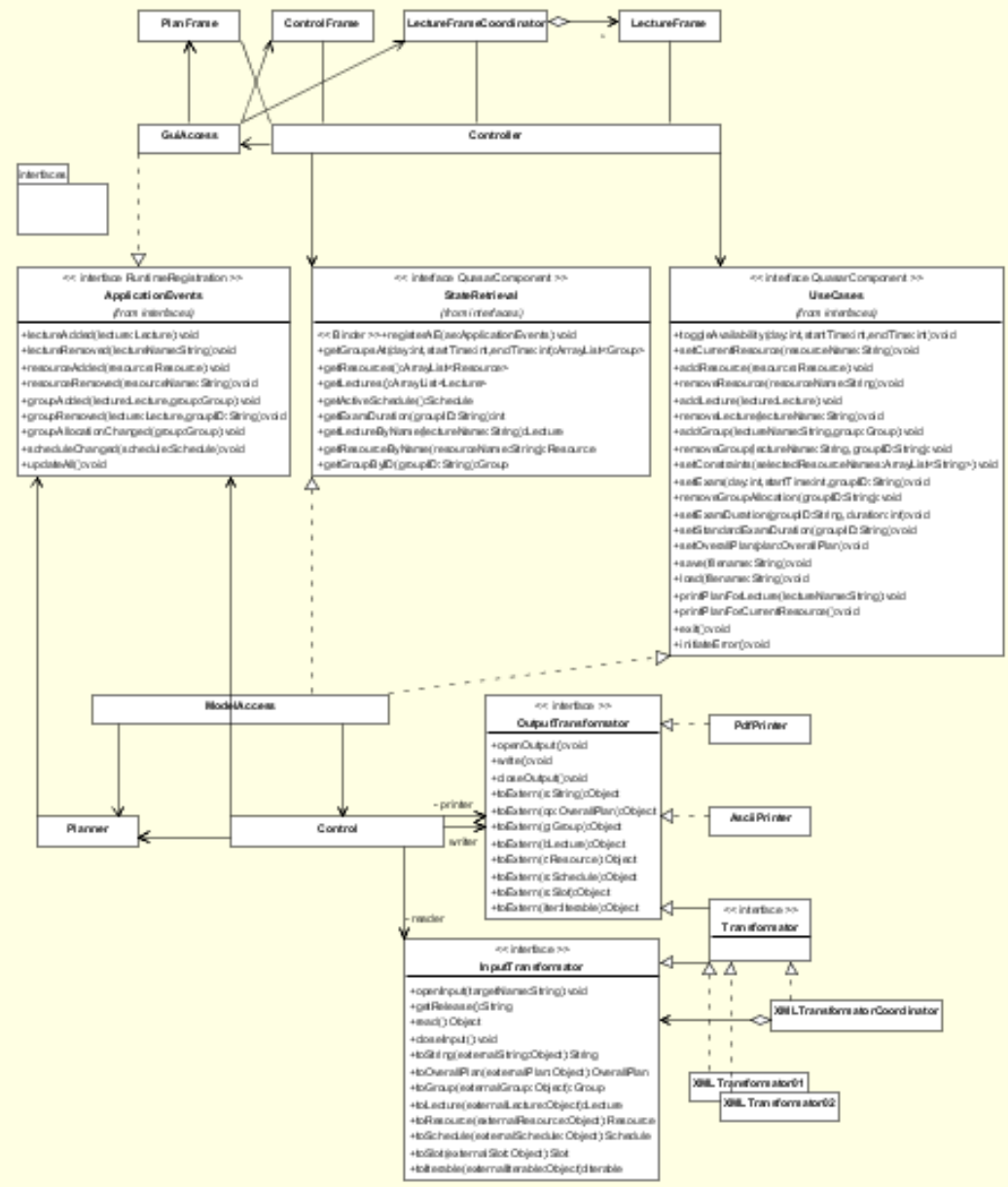
Logos for **oose.** (Innovative Information) and **Oldenburg** (Hochschule Oldenburg) are visible. The footer contains the slogan "Erfahrung nutzen, Ziele erreichen." and the **oose.** logo.

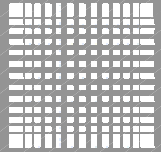


# Ad-hoc Design, nur Klassendiagramme

## Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung



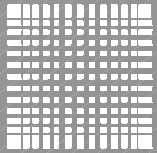


## Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

- Ziele beim Design
  - Divide et Impera: Aufgabe / Lösung durchschauen
  - Arbeitsteilung
- Klassendiagramme
  - Gut geeignet für die Darstellung der Struktur
  - Nicht gut geeignet für die Strukturierung selbst





# Produktaufteilung $\approx$ Projektaufteilung

[DeMarco 1998]

Inhalt

Vorgehen

Anforderungen

Design

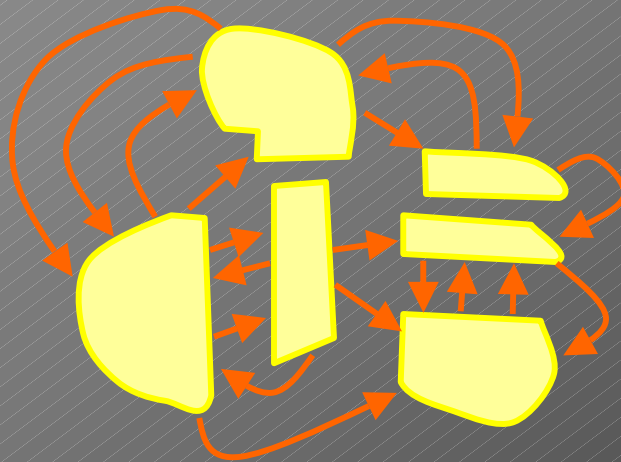
Qualitäts-  
sicherung

Was zählt...

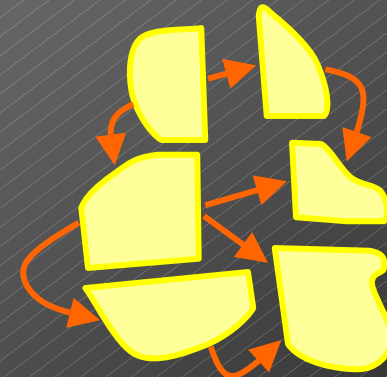
Lehre vs.  
Praxis

Schluss-  
folgerung

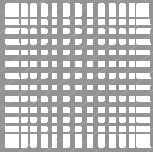
## Diese Unterteilung?



## Oder diese?



- Konsequenzen
  - Hoher Kommunikationsaufwand
  - Hoher Integrations- und Testaufwand
  - Teure Wartung



# Qualitätssicherung für die Software

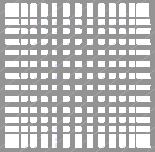
Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitätssicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schlussfolgerung

**Komplementäre Ansätze für den ganzen Lebenszyklus**

...aufzustellen.  
[FIFA Regelwerk]

**Blackbox-Test am Ende der Entwicklung**



# Blackbox-Test am Ende der Entwicklung

Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ **Qualitäts-sicherung**
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

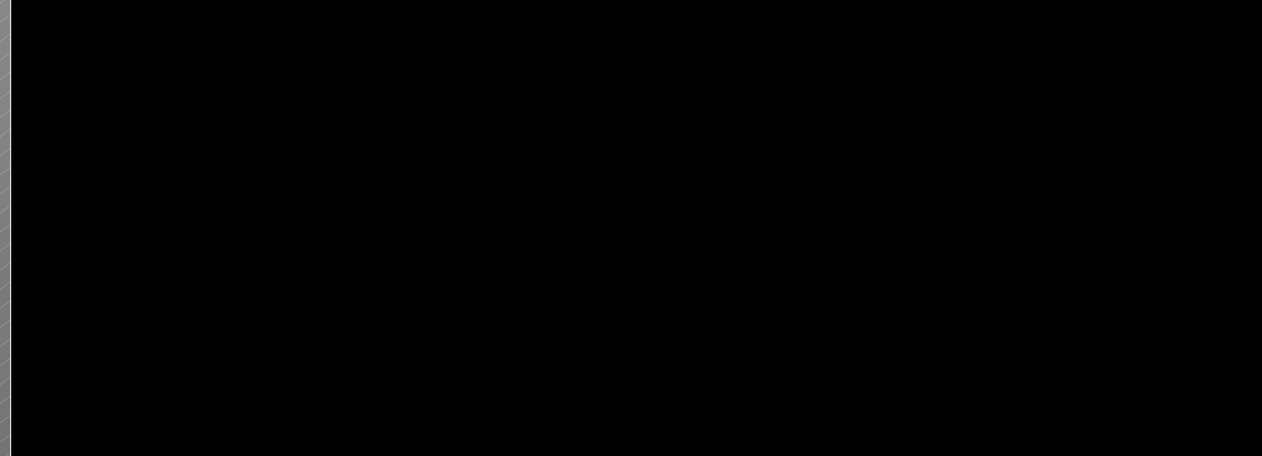
Eingaben



Eingaben



Eingaben



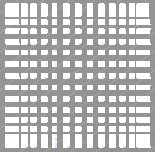
Korrekte  
Soll-Ausgabe?



Korrekte  
Soll-Ausgabe?



Korrekte  
Soll-Ausgabe?



# Blackbox-Test am Ende der Entwicklung

Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ **Qualitäts-sicherung**
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

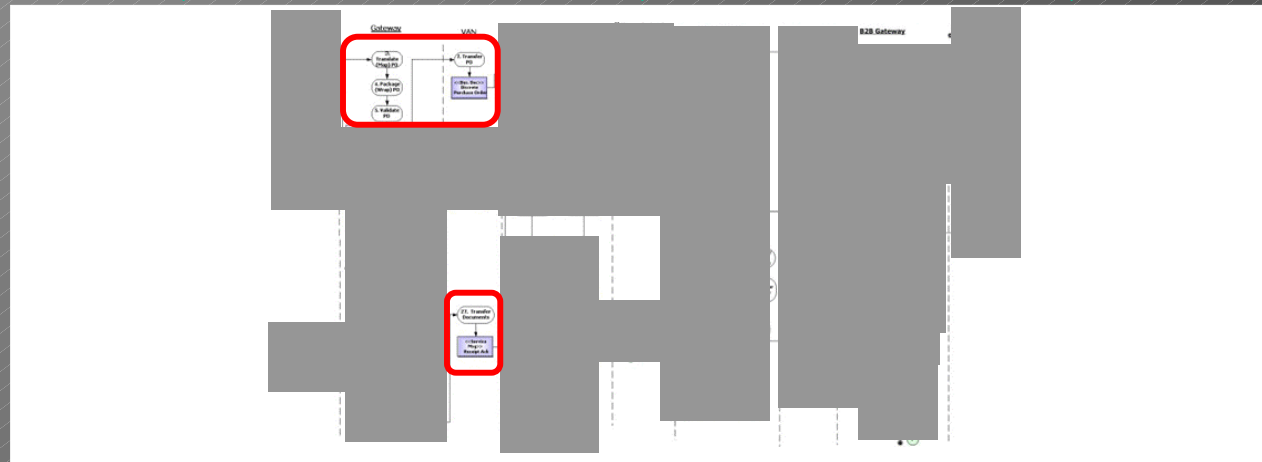
Eingaben



Eingaben



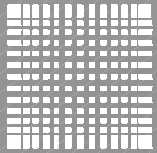
Eingaben



Korrekte  
Soll-Ausgabe?

Korrekte  
Soll-Ausgabe?

Korrekte  
Soll-Ausgabe?



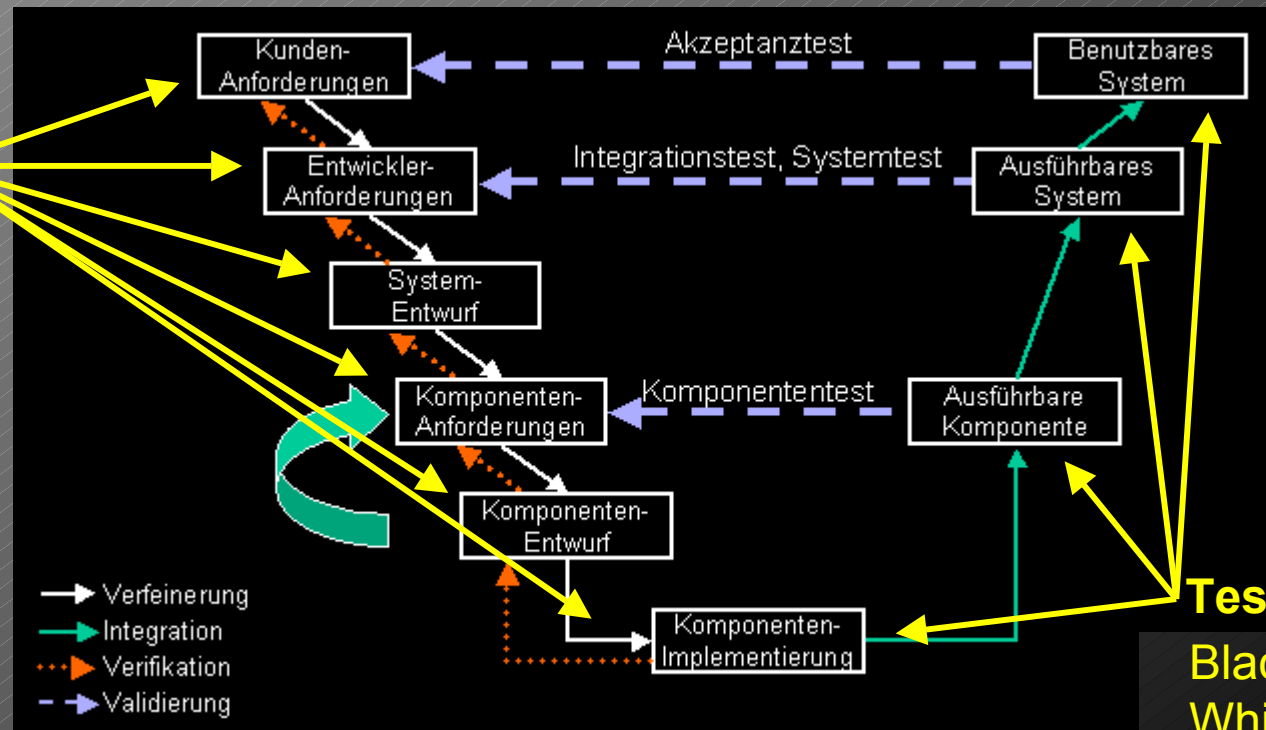
# Komplementäre QS-Ansätze für den ganzen Lebenszyklus

- Projektübergreifende Aktivitäten
  - Organisatorische Maßnahmen
  - Konstruktive Maßnahmen
  - Analytische Maßnahmen

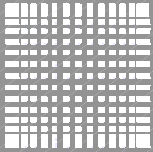
Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ **Qualitäts-sicherung**
- ▶ Was z "..."
- ▶ Lehre Praxis
- ▶ Schlussfolgerung

**Reviews, Inspektionen**



**Test**  
Blackbox  
Whitebox  
Graybox  
...



# Was zählt

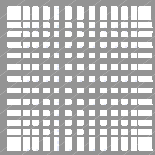
Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitätssicherung
- ▶ **Was zählt...**
- ▶ Lehre vs. Praxis
- ▶ Schlussfolgerung

Ein Tor ist gültig, wenn ein Spieler innerhalb der Spielzeit einen Ball überquert hat.  
[FIFA Regelwerk]

**Qualität der Entwicklung**

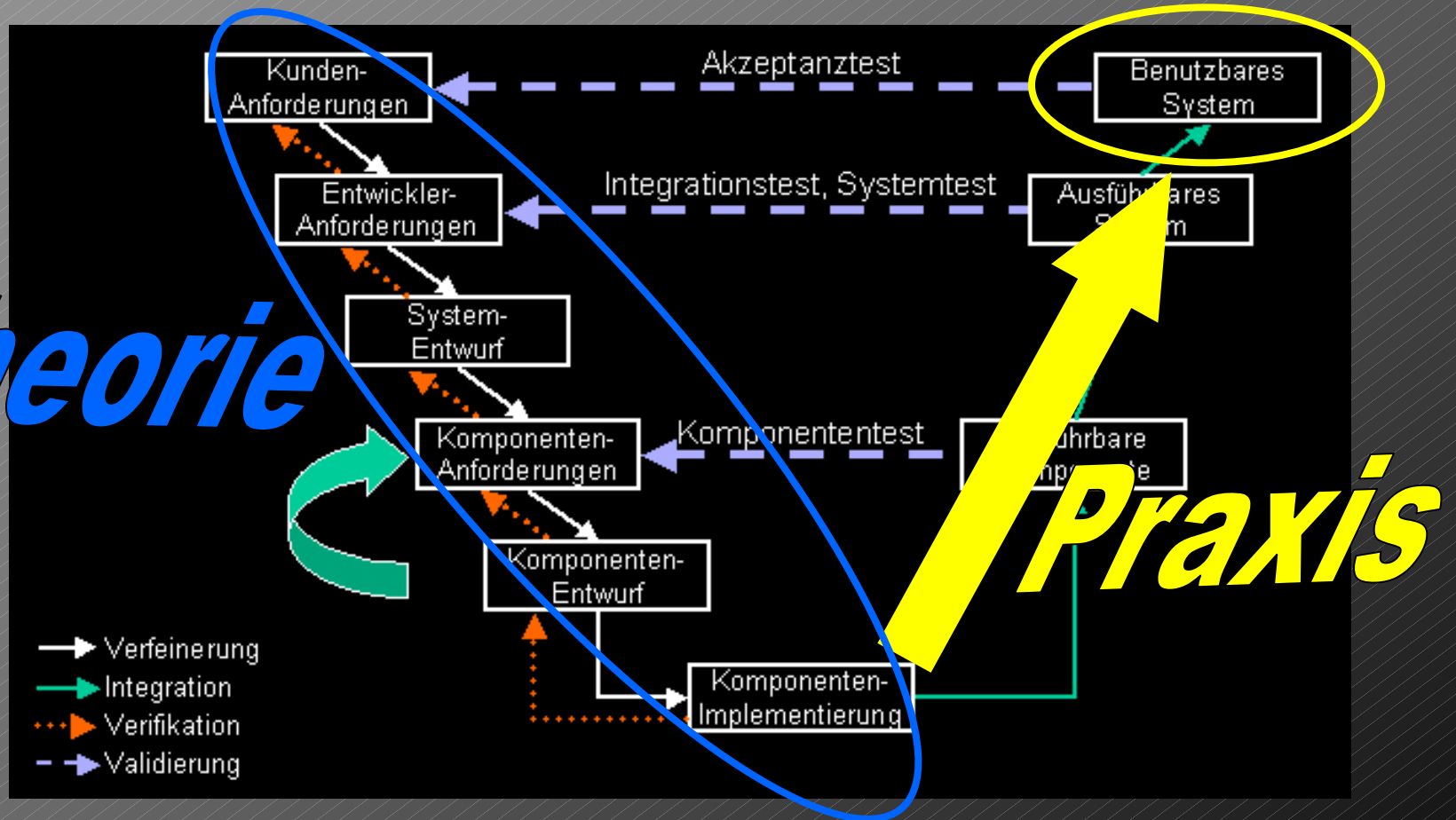
**Der Kunde**

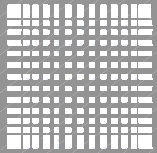


# Theorie: Form und Qualität der linken Seite; Praxis: (nur) das, was der Kunde verlangt

Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zu... **Theorie**
- ▶ Lehr- u. s. Praxis
- ▶ Schluss-folgerung





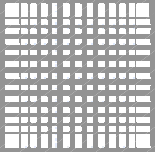
# Korrespondenz von (theoretischer) Lehre und Praxis

## Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

Das Spielfeld **muss** rechtwinklig sein.  
[FIFA Regelwerk]





# Korrespondenz von Lehre und Praxis

## Inhalt

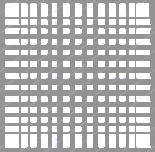
- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

## Lehre

- Wegwerf-Prototypen, Machbarkeitsstudien, z.B. Diplomarbeiten
- Software Engineering-Lehre endet (unmittelbar) vor der Übergabe / dem Akzeptanztest
  - Keine Erfahrungen mit Kunden
  - Keine Erfahrungen mit Weiterentwicklung, mit fremdem Code

## Praxis

- Weiterverwendung/Ausbau
  - "Es gibt keine Fehlschläge", nur positive Bewertungen
  - Keine Zeit für Experimente...
- Wartung = 50 bis 75 Prozent des Tagesgeschäfts
  - Fehlerkorrekturen
  - Ergänzungen
  - Anpassungen
  - Strukturverbesserungen



# Korrespondenz von Lehre und Praxis

## Inhalt

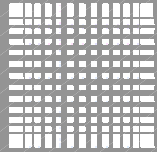
- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

## Lehre

- Dokumentation soll existieren für
  - Anforderungen
  - Design
  - Implementierung
  - Test
  - ...
  
- Dokumentation soll sein
  - Vollständig
  - Konsistent
  - Klar
  - Aktuell

## Praxis

- Dokumentation existiert für
  - Bestenfalls Klassendiagramme aus dem Code generiert, sonst ist Vieles veraltet
  
- Dokumentation ist oft
  - Unvollständig
  - Nicht aktuell



## Die Lösung...

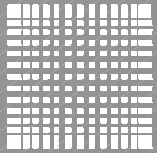
### Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

Seine [des Schiedsrichters] Entscheidung ist eine Entscheidung über die Tatsachen, die vor ihm liegen, und über die Anwendung der Regeln, die er einschließen, ob er das Spiel eines Mannes oder seiner Umgebung oder unter bestimmten Umständen ein Spiel zulassen oder nicht.  
[FIFA Regelwerk]

**Lehre muss praktische Gegebenheiten zur Kenntnis nehmen**

**Praxis sollte wissenschaftliche Erkenntnisse nutzen**



# Lehre muss praktische Gegebenheiten zur Kenntnis nehmen

## Inhalt

Vorgehen

Anforderungen

Design

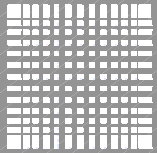
Qualitäts-  
sicherung

Was zählt...

Lehre vs.  
Praxis

Schluss-  
folgerung

- Flexible Prozessmodelle akzeptieren und konsequent lehren: RUP, agile Vorgehensmodelle
- Realistische Anforderungsdokumentation lehren: Testcode dokumentiert den Produktivcode (**zusammen mit** testgetriebener Entwicklung, "nur getesteter Code existiert")
- Umgang mit Kunden lehren, Lehrprojekte mit realistischen Kunden(-Beziehungen) anbieten
- Analyse und Weiterentwicklung von Code lehren



# Praxis sollte wissenschaftliche Erkenntnisse nutzen

## Inhalt

Vorgehen

Anforderungen

Design

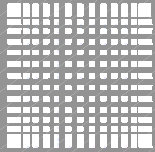
Qualitäts-  
sicherung

Was zählt...

Lehre vs.  
Praxis

Schluss-  
folgerung

- Schrittweise Veränderungen (Verbesserungen?) mit kontinuierlichen Messungen des jeweiligen Nutzens: empirisches SE, GQM
- Ist-Prozess dokumentieren, daran verbessern
- Anforderungen: (nur) die wichtigen Dinge (aber die alle!)
- Qualitätssicherung ist mehr als (Blackbox-)Testen: organisatorische und konstruktive Maßnahmen, Inspektionen, testgetriebene Entwicklung
- Kunden einbeziehen, an das Projektteam heranlassen
- Design ist mehr als Klassendiagramme (= Struktur-Entwürfe): Use Cases, Dynamik!
- Ergebnis von Verbesserungen wieder investieren: über das aktuelle Projekt hinaus denken, Produktlinien?



## Inhalt

- ▶ Vorgehen
- ▶ Anforderungen
- ▶ Design
- ▶ Qualitäts-sicherung
- ▶ Was zählt...
- ▶ Lehre vs. Praxis
- ▶ Schluss-folgerung

- Neville Gabie: Tore, Sanssouci im Carl Hanser Verlag, 2006
- GfK Marktforschung GmbH, Fraunhofer IESE, Fraunhofer ISI: Analyse und Evaluation der Softwareentwicklung in Deutschland – Eine Studie für das Bundesministerium für Bildung und Forschung, 2000
- Tom DeMarco: Der Termin, Carl Hanser Verlag, 1998
- The Standish Group International, Inc.: The CHAOS Report (1994), [http://www.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www.standishgroup.com/sample_research/chaos_1994_1.php)

**Und nicht vergessen:  
nur noch 21 Tage bis zum Anpfiff!**