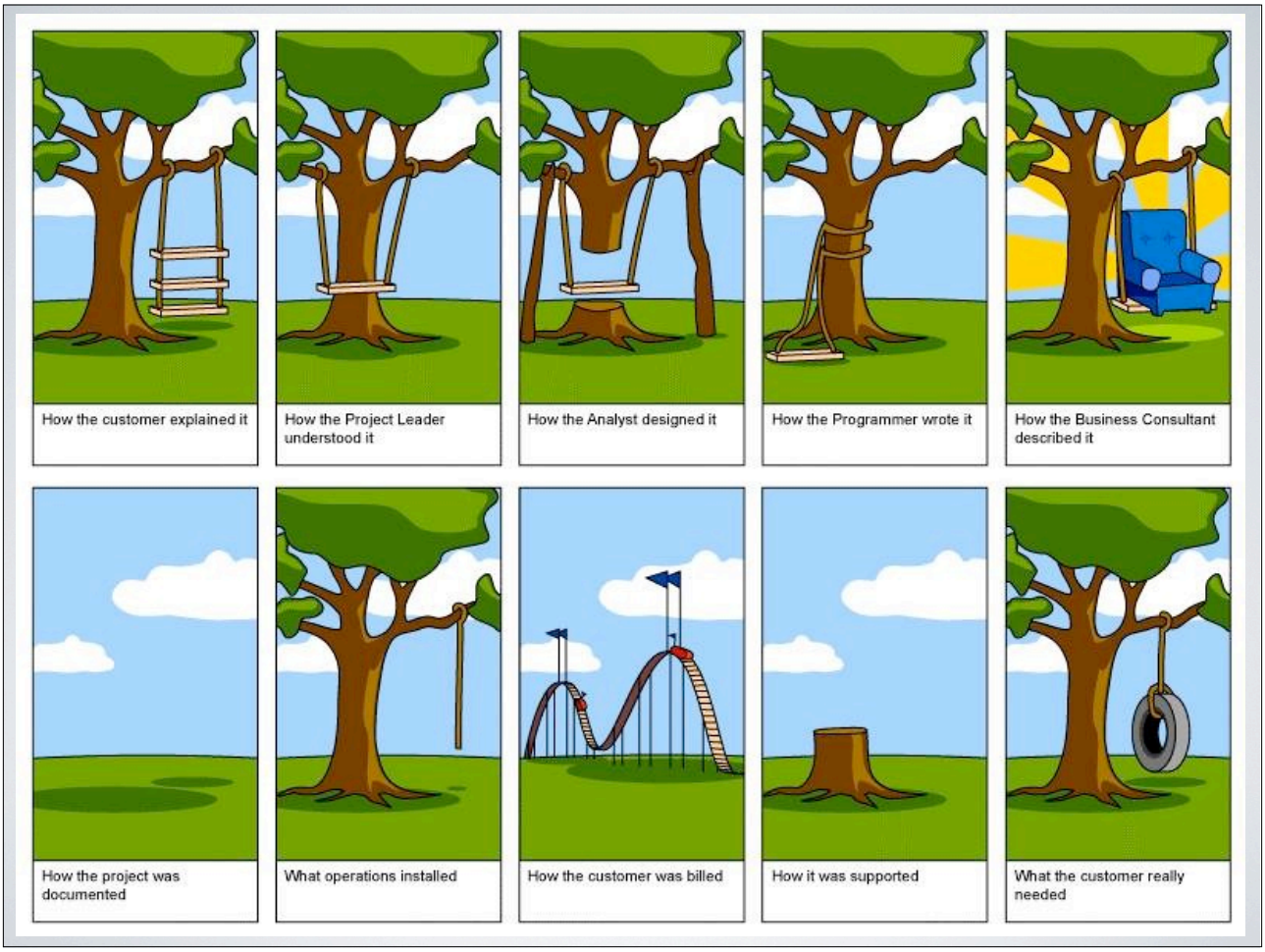
OO- SW-Entwicklung

Objektorientiert Entwerfen und Implementieren

Das Problem...



FRÜHER... -> STRUKTURIERT

- Daten sind in Bewegung
  -> Datenflüsse
  -> Datenmodelle
- Funktionen sind statisch
  -> verarbeiten Daten

Daten werden hin und her gereicht

Die Softwarekrise

5 typische Probleme

Problem 1
Produktivität

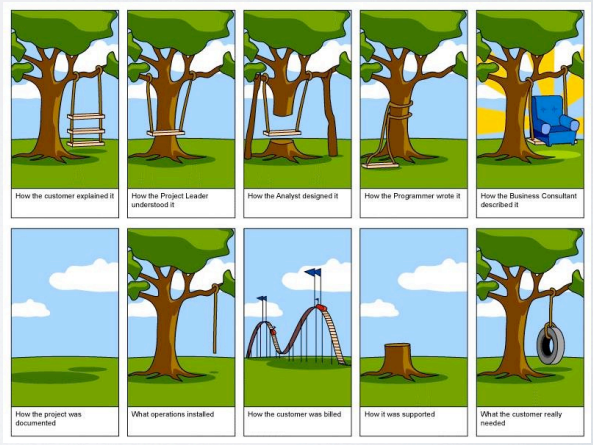
- Geringe Produktivität der Programmierer
  alles wird immer wieder neu geschrieben
- Wenig oder gar kein Code re-use
- Copy / Paste Programmierung

Problem 2
Teurer Unterhalt

- Viel Unterhalt, wenig neu Erstellen
- Bedürfnisse ändern sich -> Programme müssen angepasst werden
  - Das ist zwar gut, aber sehr aufwendig!

Problem 3
Unterschiedliches Vorgehen

- Verwendet verschiedenste Beschreibungen und Modelle
- Unterschiedliche Sprache für Analyse und Entwurf
- Übereinstimmung zw. Datenfluss und Datenmodell sehr aufwendig



Problem 4
Wo ist oben?

- Top-Down Ansatz ist zwar eine gute Sache, aber...
  - Wo ist bei einer interaktiven Applikation eigentlich “oben”?
  - Klassischer top-down-Ansatz nur bei statischen Problemen sinnvoll (Raketenbahn vs. interaktiver Appl.)
- Zerlegen von oben nach unten verhindert Blick aufs Ganze
- -> Spezifikation durchdringt den ganzen Code

Problem 5
Offen oder Geschlossen?

- Zusammengehöriger Code wird zu Modulen zusammengefasst

  Vorteil -> Lokalisierte, bessere Wartbarkeit, dadurch besser wiederverwendbar

  Nachteil -> Bei Wiederverwendung passt es dann doch nicht ganz, also doch abändern und anpassen!

Lösung: Objektorientierung

Prozedurale Programmierung

- Daten und Operationen
- Daten sind statisch, Operationen transient

OO Programmierung

- Zusätzliche Abstraktionsebene
- Objekte kapseln Daten und Verhalten
- Klassen fassen gleiche Objekttypen zusammen
- Erweiterbarkeit durch Vererbung

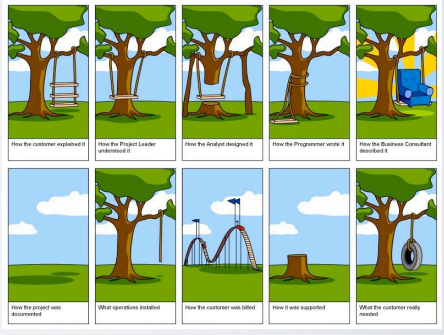
Durchgängige Sprache

Die Objektorientierung erlaubt es...

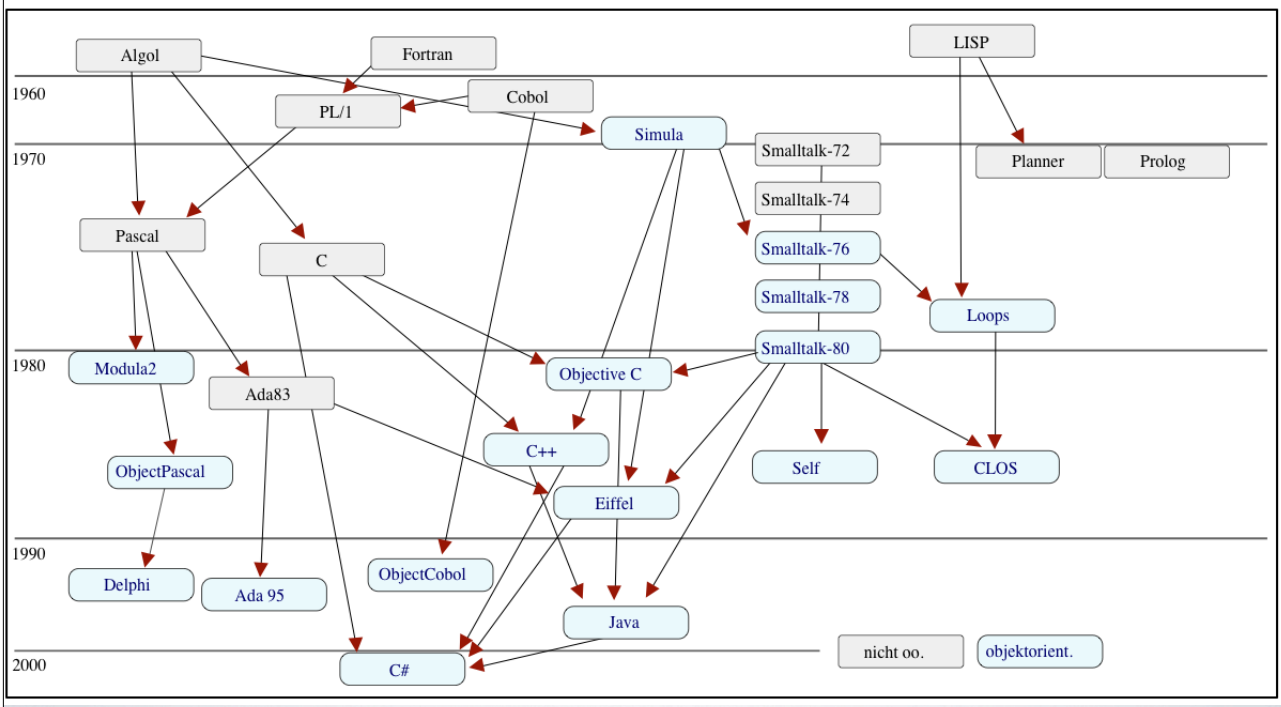
- für Analyse und Entwurf dieselben Konzepte zu nutzen

Dies bedeutet...

- Keine Strukturbrüche, daher bessere Nachvollziehbarkeit
- Durchgängigkeit zwischen Analyse und Entwurf



PROGRAMMIERSPRACHEN

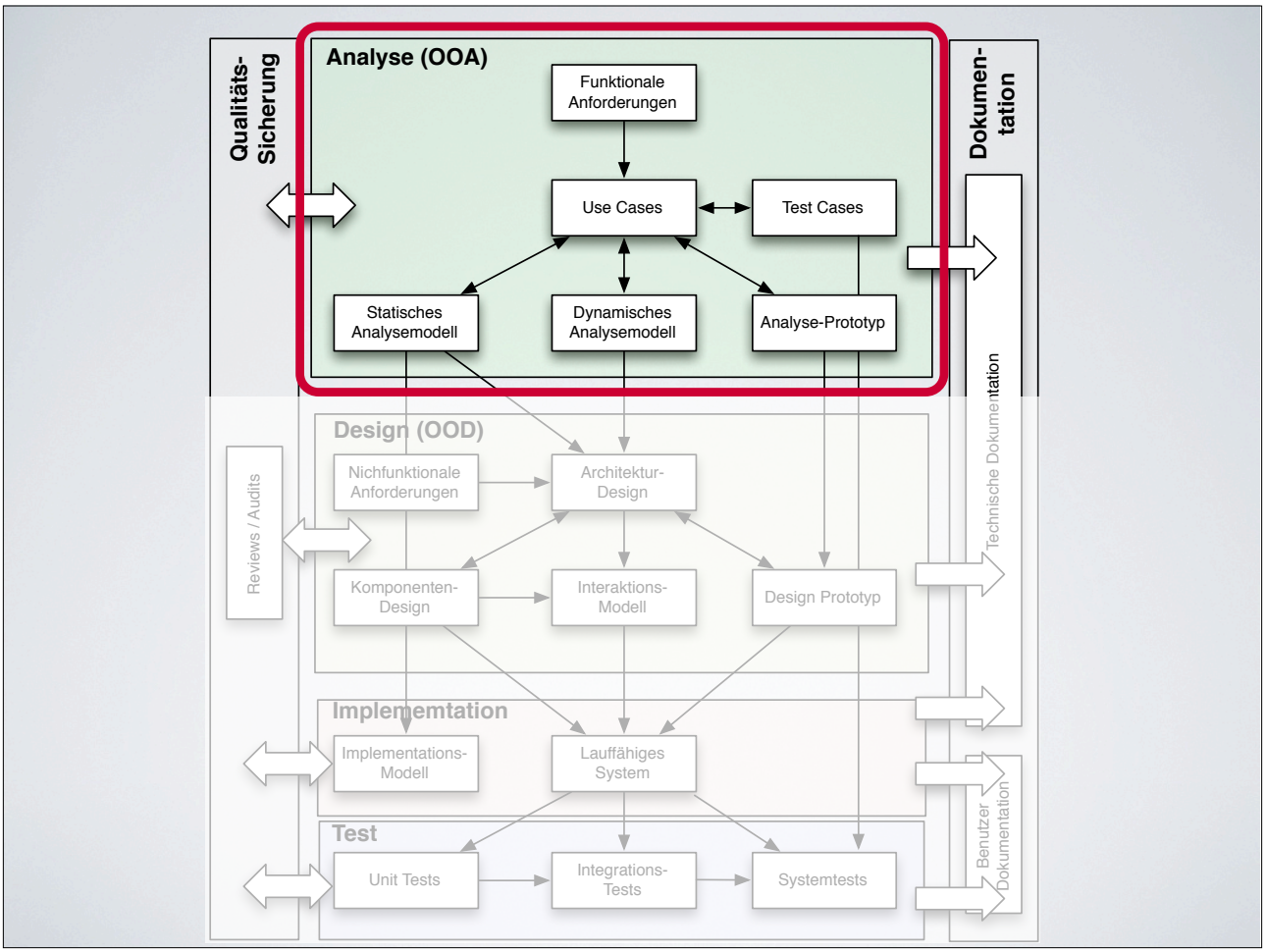


UML

- Unified Modelling Language
  - > Grafische Modelliersprache
- Entstanden Mitte der 1990er Jahre
- Heute Industriestandard

OOA/OOD

The Big Picture...



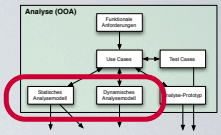
Ergebnisse der OOA

- Pflichtenheft
  - > das Einstiegsdokument
- OOA-Modell
  - > das Fachkonzept
- Prototyp der Benutzeroberfläche
  - > Visualisierung des Fachkonzepts

Pflichtenheft

- Umgangssprachliche Beschreibung dessen, was das zu realisierende System leisten soll
  - > Weniger detailliert als das OOA-Modell
  - > Enthält auch einige Informationen, die nicht im OOA-Modell dargestellt werden
- Zwei Zielsetzungen
  - > Einstiegsdokument in das Projekt für alle, die das System später pflegen und warten sollen
  - > Ausgangsbasis für die objektorientierte Modellbildung
- Das Pflichtenheft ist nicht die Vorlage für den Entwurf bzw. Programmierer, sondern für den OOA-Modellierer

OOA-Analyse-Modell



Assoziation

Statische Konzepte

Vererbung Paket

Das statische Modell beschreibt ...
- die Klassen des Fachkonzepts
- die Assoziationen zwischen den Klassen
- die Generalisierungsstrukturen
- die Attribute der Klassen (Daten des Systems)
- die Bildung von Paketen (Teilsysteme)

Botschaft Zustands-automat

Attribut

Basis

Objekt

Das dynamische Modell zeigt Funktionsabläufe
- Use-Cases beschreiben die durchzuführenden Aufgaben auf einem sehr hohen Abstraktionsniveau
- Aktivitäten beschreiben die Ausführung von Funktionalität bzw. Verhalten
- Szenarien zeigen, wie Objekte miteinander kommunizieren, um eine bestimmte Aufgabe zu erledigen
- Zustandsautomaten beschreiben die Reaktionen eines Objekts auf verschiedene Ereignisse (= Botschaften)

Verhalten des Systems im

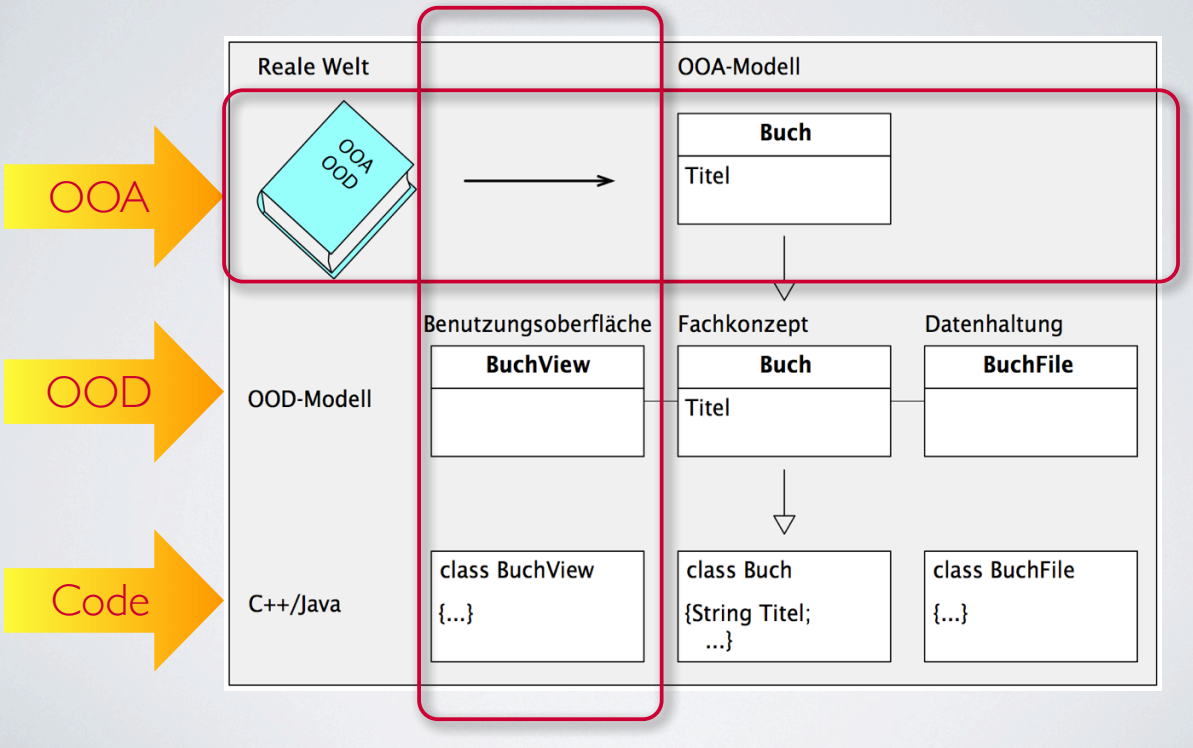
Nicht veränderbare Aspekte
(grundlegende Struktur)

Prototyp der Benutzeroberfläche
- Ablauffähiges Programm, das alle Attribute des OOA-Modells auf der
- Realisiert Trennung von Fachkonzept und Benutzungsoberfläche

Statisches Modell

Dynamisches Modell

Durchgehende Konzepte



Statisches oder dynamisches Modell?

– Ordnen Sie die folgenden Aussagen dem statischen oder dem dynamischen Modell zu

Aussage	Statisches Modell	Dynamisches Modell
Beschreibt das Verhalten des zu entwickelnden Softwaresystems		X
Bildet den stabilen Kern des objektorientierten Modells	X	
Beschreibt die Beziehungen zwischen Klassen (bzw. ihren Objekten)	X	
Zeigt, wie Objekte miteinander kommunizieren, um eine bestimmte Aufgabe zu erledigen		X
Modelliert die Struktur des Softwaresystems	X	
Beschreibt die Reaktionen eines Objekts auf verschiedene Ereignisse		X
Enthält die Aufteilung des Systems in Teilsysteme	X	