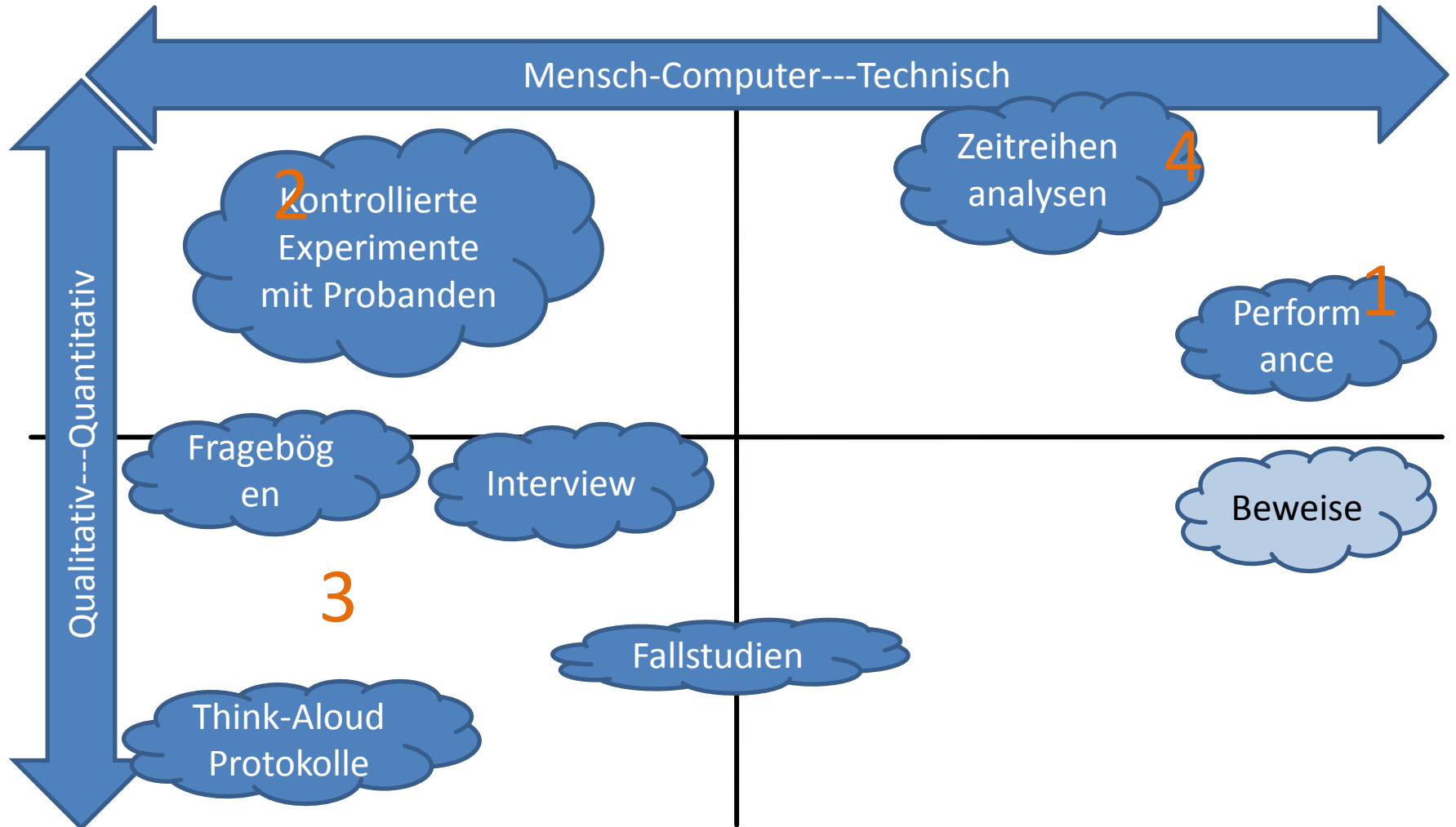


# Kontrollierte Experimente

# Einordnung



# Lernziele

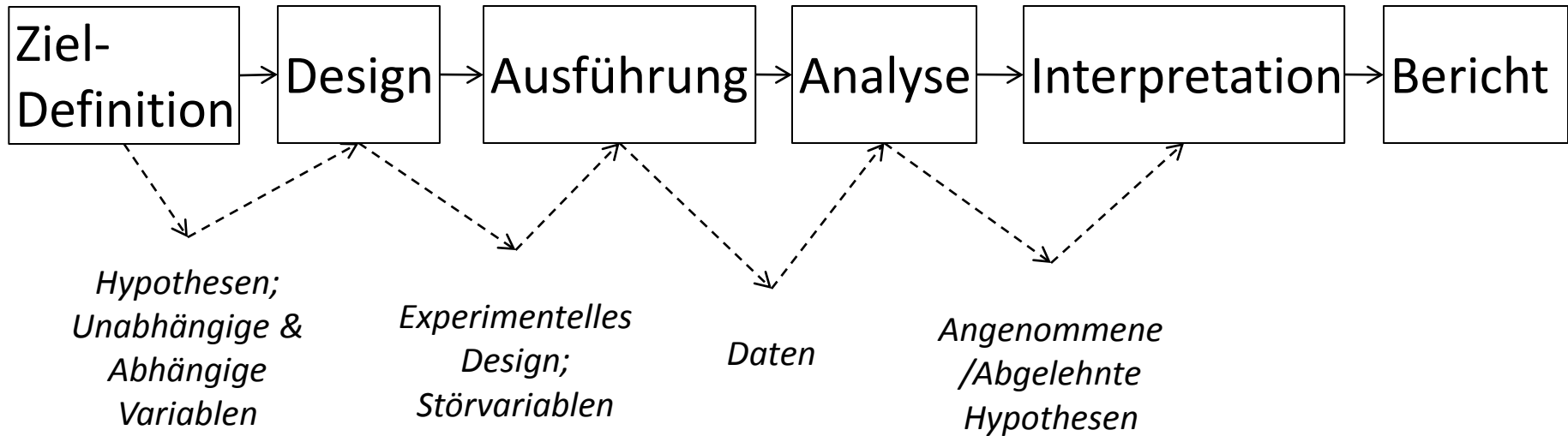
- Gute Hypothesen aufstellen können
- Experiment mit hoher interner oder hoher externer Validität entwerfen können

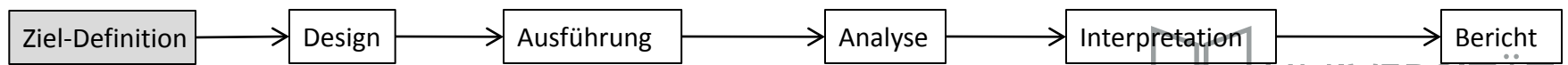


# Definition

- Systematische Studie
- Ein oder mehrere Faktoren werden variiert
- Alles andere konstant halten
- Ergebnis der systematischen Variation wird beobachtet

# Experimentelle Phasen





# Variablen

# Unabhängige Variablen

- Absichtlich, systematisch variiert durch Versuchsleiter
- Faktor, Prädiktor (-variable)
- Alternativen, Level, Stufen, Treatment
- Beispiele:
  - Programmierparadigma
  - Sprache
  - Workload

# Abhängige Variable

- Ergebnis eines Experiments
- Hängen ab von Variation der unabhängigen Variablen
- Beobachtet
  
- Beispiele:
  - Performance
  - Programmverständnis
  - Produktivität von Entwicklern



# Latente Variablen

- Konstrukt
- Nicht direkt beobachtbar
- Beispiele
  - Programmverständnis
  - Intelligenz
  - Performance

# Operationalisierung

- Operationen definieren, mit denen man Variablen messen kann
- Darf gesundem Menschenverstand nicht widersprechen

# Aufgabe

- Operationale Definitionen für folgende Variablen
  - Performance
  - Programmverständnis
  - Intelligenz
  - Wartbarkeit

# Hypothesen

- Erwartungen über Ergebnisse
- Erwartungen müssen begründet sein in Theorie oder Praxis
- Hypothesen müssen einfach und klar formuliert sein
- Hypothesen müssen überprüfbar sein
- Falsifizierbarkeit (Hausaufgabe: Logik der Forschung und Falsifizierbarkeit erklären)

# Hypothese-Negativbeispiel

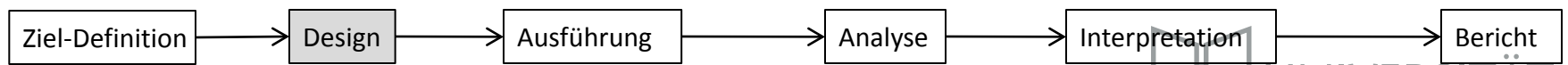
- Schlechte Kommentare sind schlecht für Programmverständnis
- Gute Kommentare sind gut für Programmverständnis

# Besser

- Kommentare, die jedes Statement von Quelltext beschreiben, haben keinen Einfluss auf Antwortzeit beim Verstehen von Quelltext
- Kommentare, die falsche Informationen über Quelltext enthalten, verlangsamen Programmverständnis
- Kommentare, die den Zweck von Statements beschreiben, beschleunigen Programmverständnis

# Aufgabe

- Stellen Sie je eine Hypothese zu folgenden Forschungsfragen auf:
    - Erhöht Objektorientierung die Produktivität von Entwicklern?
    - Ist Java besser als C++?
    - Ist MergeSort schneller als Quicksort?
  - Die Hypothese muss überprüfbar sein, begründet sein; die Variablen müssen operationalisiert sein
- = Beispiel für Prüfungsfrage



# Design



# Validität

- Wird das gemessen was gemessen werden soll?

# Interne Validität

- Maß, in dem Wert der abhängigen Variablen auf Variation der unabhängigen Variablen zurückgeführt werden kann

# Externe Validität

- Maß, in dem Ergebnisse aus einem Experiment auf andere Umstände (Probanden, Material,...) übertragen werden kann  
= Verallgemeinerbarkeit

# Hausaufgabe

- Recherchieren Sie andere Validitätsarten

# Gefahren/Bedrohungen

- Störvariablen:
  - Beeinflussen abhängige Variable zusätzlich zu unabhängiger Variablen
  - Lerneffekte
  - Hawthorne-Effekt
  - Messinstrumente
  - Selektion
  - ...

# Aufgabe

- Messen von Programmverständnis: Welche Störvariablen gibt es?
- Wie könnte man diese Störvariablen kontrollieren?

# Störvariablen

- Es gibt viele Störvariablen
- Sorgfältig identifizieren und kontrollieren
  - Randomisierung
  - Matching/Parallelisierung/Balancing
  - Störvariable als unabhängige Variable definieren
  - Störvariable konstant halten
  - Nachträgliche Analyse

# Randomisierung

- Zufallszahlengenerator
- Münze werfen
- Würfeln
- ...
  
- Probleme:
  - Gruppen müssen groß genug sein
  - 5 pro Gruppe zu wenig, 10 scheint akzeptabel



# Matching/Parallelisierung/Balancing

Proband	Wert
P5	65
P9	56
P3	42
P4	34
P10	24
P6	23
P7	21
P8	16
P2	12
P1	5

Gruppe A	Gruppe B
65	56
34	42
24	23
16	21
12	6

odd-even-even-odd/  
ABBA

# Matching/Parallelisierung/Balancing

- Nachteil gegenüber Randomisierung:
  - Störvariable muss gemessen werden
  - Programmiererfahrung?
  - Intelligenz?
- Vorteil gegenüber Randomisierung:
  - Genauere Kenntniss über Parameter

# Parameter als unabhängige Variable definieren

- Wird systematisch vom Versuchsleiter variiert
- Störvariable wird operationalisiert
- Programmiererfahrung:
  - Statisch/wenig Erfahrung
  - Statisch/viel Erfahrung
  - Dynamisch/wenig Erfahrung
  - Dynamisch/viel Erfahrung

# Rechenbeispiel

- 23 Störvariablen, jede mit 2 Stufen  
= 8 388 608 mögliche Kombinationen
- Wie viele Probanden sind nötig, um jede Kombination abzudecken?
  - min. 10 Probanden pro Gruppe
  - 83 886 080 (ganz Deutschland)

# Konstant halten

- Nur ein Level einer Störvariable
- Programmierfahrung
  - Nur Bachelor-Studenten
  - Nur Programmierexperten
- Intelligenz
  - Nur Studenten mit bestimmter Note

# Nachträgliche Analyse

- Variable wird während des Experiments gemessen
- Einfluss einer Variablen wird nach dem Experiment analysiert
- Probleme:
  - Kann zeigen, dass Ergebnisse unbrauchbar sind

# Verhältnis von Interner und Externer Validität

- Beide verlangen verschiedene Dinge
  - Intern: alles kontrollieren
  - Extern: allgemeines Setting
- Und jetzt?
  - Erst interne Validität maximieren
  - Dann schrittweise externe Validität erhöhen

# Qualitätskriterien Empirischer Studien

- Validität
- Reliabilität
- Objektivität

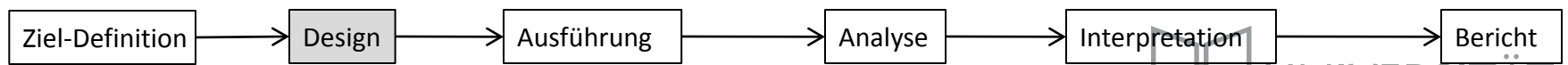


# Qualitätskriterien Empirischer Studien

- Reliabilität:
  - Genauigkeit der Messinstrumente
- Objektivität:
  - Durchführung eines Experiments darf nicht von Person der Versuchsleiter abhängen
  - Dasselbe Experiment, durchgeführt von anderen Versuchsleitern, soll dasselbe Ergebnis liefern

# Beispiel

- Waage zum Messen des Gewichts:
  - Valide
  - Reliabel je nach Qualität
  - Digitale Waage ist objektiver, da jeder die selbe Zahl sieht (analog lässt etwas Spielraum)
- Die selbe Waage zum Messen der Körpergröße:
  - Weniger valide
  - Reliabel je nach Qualität



# Experimentelle Versuchspläne

# Pläne

- Between vs. Within Subject  
= Mit vs. ohne Messwiederholung
- Einfaktoriell vs. Mehrfaktoriell  
= Eine vs. mehrere unabhängige Variablen
- Univariat vs. Multivariat  
= Eine vs. mehrere abhängige Variablen

# Warum Pläne?

- Handlungsanweisung
- Kommunikationserleichterung
- Entscheidungsgrundlage für statistische Auswertung

# Wahl eines Plans

- Abhängig von:
  - Effektstärke
  - Stichprobengröße
  - Je größer beides ist, desto weniger kommen ungewollte Einflüsse zum Tragen; desto weniger ist ein geeignetes Design wichtig
- Da meistens kleine Stichprobe und unbekannte Effektgröße in Softwareengineering, ist richtiges Design sehr wichtig

# Einfaktoriell

# Between-Subjects

- Probanden werden in Gruppen aufgeteilt
- So viele Gruppen wie Stufen der unabhängigen Variablen
- Ergebnisse werden zwischen Gruppen verglichen

Gruppe	Stufen
A	Textuelle Annotationen
B	Hintergrundfarben

```
21 public class PhotoListScreen extends List {
22
23     //Add the core application commands always
24     public static final Command viewCommand = new Com
25     public static final Command addCommand = new Com
26     public static final Command deleteCommand = new C
27     public static final Command backCommand = new Com
28
29     public static final Command editLabelCommand = ne
30
31     // #ifdef includeCountViews
32     public static final Command sortCommand = new Com
33     // #endif
34
35     // #ifdef includeFavourites
36     public static final Command favoriteCommand = new
37     public static final Command viewFavoritesCommand
38     // #endif
39
40     /**
41     * Constructor
```

```
21 public class PhotoListScreen extends List {
22
23     //Add the core application commands always
24     public static final Command viewCommand = new
25     public static final Command addCommand = new C
26     public static final Command deleteCommand = ne
27     public static final Command backCommand = new
28
29     public static final Command editLabelCommand =
30
31     public static final Command sortCommand = new
32
33     public static final Command favoriteCommand =
34     public static final Command viewFavoritesComm
35
36     /**
37     * Constructor
38     */
39     public PhotoListScreen() {
40         super("Choose Items", Choice.IMPLICIT);
41     }
```



# Probleme

- Varianzen zwischen Probanden (=interindividuelle Unterschiede) sind groß
- 10x (What does 10x Mean? Measuring Variations in Programmer Productivity. Steve McConnell.)
- Ausreichend Probanden
- Balancierung der Gruppen

# Within-Subjects

- Interindividuelle Differenzen sollen berücksichtigt werden
- Jeder Proband erfährt alle Stufen der unabhängigen Variablen

Eine Gruppe	Session 1	Session 2
	Hintergrund-farben	Textuelle Annotationen

# Probleme

- Lerneffekte
  - Besonders bei kreativen Aufgaben problematisch
  - Möglichst unterschiedliche, gleichzeitig ähnliche Aufgaben notwendig
- Reihenfolge-Effekte
- Intraindividuelle Unterschiede:
  - Müdigkeit
  - Motivation
- Mortality

# Crossover

- Jeder Proband erfährt alle Stufen
- Vergleich zwischen Gruppen und innerhalb von Gruppen möglich

Gruppe	Session 1	Session 2
A	Hintergrund-farben	Textuelle Annotationen
B	Textuelle Annotationen	Hintergrund-farben

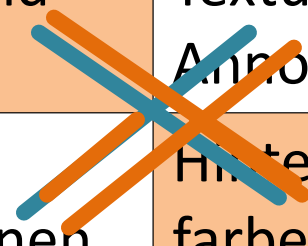
# Probleme

- Intraindividuelle Unterschiede
- Interindividuelle Unterschiede
- Mortality

# Vorteile

- Lerneffekte überprüfen:
  - Unterschied zwischen beiden Sessions für beide Stufen
- Reihenfolge-Effekte überprüfen:
  - Unterschied zwischen beiden Sessions für eine Stufe

Gruppe	Session 1	Session 2
A	Hintergrund-farben	Textuelle Annotationen
B	Textuelle Annotationen	Hintergrund-farben



# Vergleich

Eigenschaft	Between-Subjects	Within-Subjects	Cross-Over
Probandenzahl	2	1	2
Gruppenbalancierung	2	1	2
Lerneffekte	2	3	1
Reihenfolge-Effekte	2	3	1
Mortality	1	2	2
Motivation, Müdigkeit	1	2	2
Experimentdauer	1	2	2
Interne Validität	2	2	1
Externe Validität	2	2	1

# Mehrfaktorielle Pläne



# Latin Square

Group	Aufgabe 1	Aufgabe 2
A	Hintergrund-farben	Textuelle Annotationen
B	Textuelle Annotationen	Hintergrund-farben

- Ähnlich zu/Spezialform von Cross-Over
- Aber unterschiedliche Aufgaben in Sessions
- Aufgabe ist hier 2. Faktor

# Zweifaktoriell, Between-Subjects

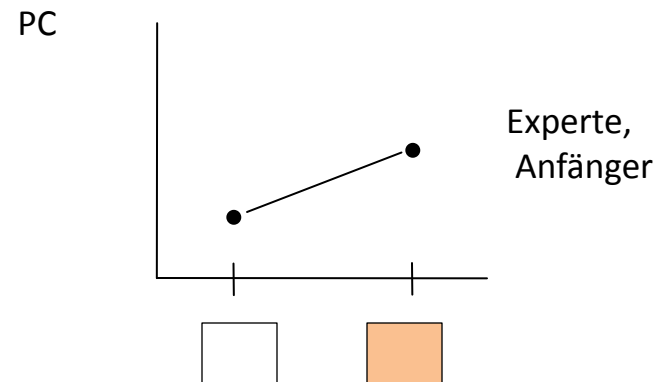
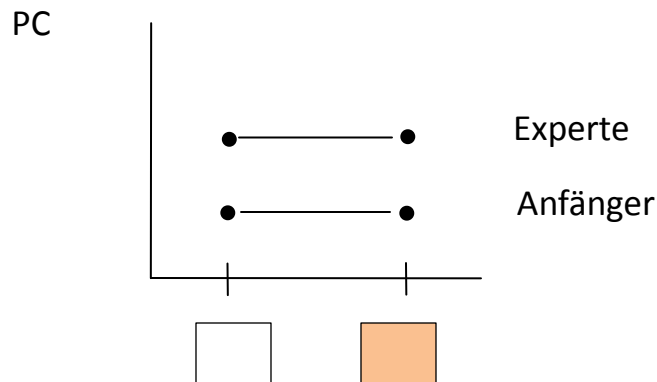
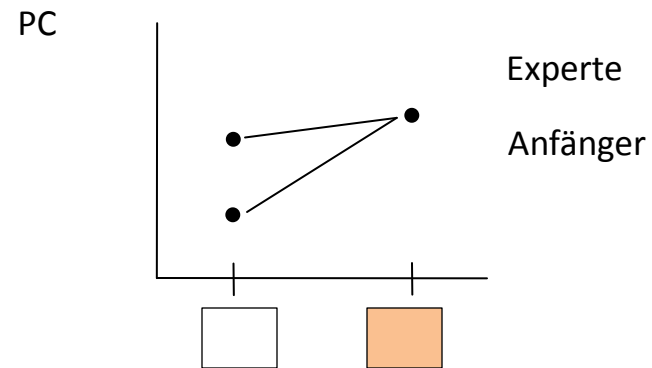
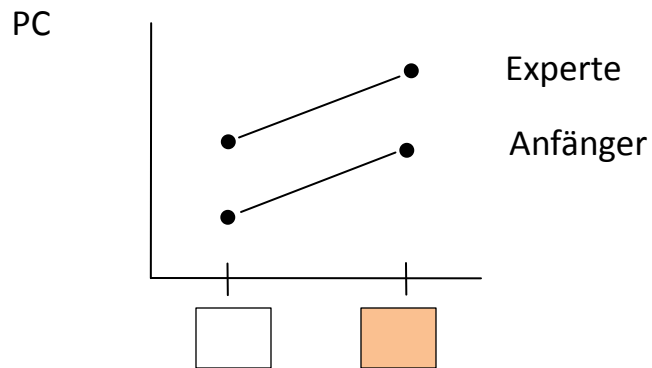
- Programmiererfahrung, Intelligenz

Variablen	Gruppen
Hintergrundfarbe/ Anfänger	Group A
Hintergrundfarbe/ Experte	Group B
Textuell/ Anfänger	Group C
Textuell/ Experte	Group D

# Zweifaktoriell, Within-Subjects

Group	Session 1	Session 2	Session 3	Session 4
Hintergrundfarbe/ Anfänger	Group A	Group D	Group C	Group B
Hintergrundfarbe/ Experte	Group B	Group A	Group D	Group C
Textuell/ Anfänger	Group C	Group B	Group A	Group D
Textuell/ Experte	Group D	Group C	Group B	Group A

# Haupt- und Interaktionseffekte



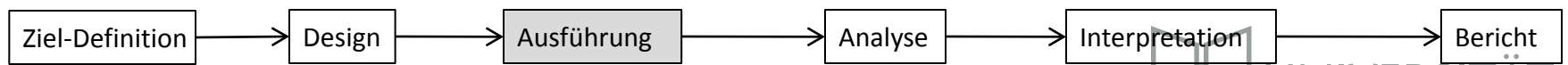
# Mehrfaktorielle Versuchspläne

- Wenn die bisher gezeigten Pläne nicht ausreichen...
- 4-faktorieller Plan (2x2x3x2)
- Interaktionen höherer Ordnung

		$C_1$		$C_2$		$C_3$	
		$B_1$	$B_2$	$B_1$	$B_2$	$B_1$	$B_2$
$A_1$	$D_1$						
	$D_2$						
$A_2$	$D_1$						
	$D_2$						

# Wahl eines Plans

- Möglichst einfachen Plan auswählen
- Vor-und Nachteile sorgfältig abwägen
- Ressourcenbeschränkung beachten



# Ausführung

# Was kann jetzt noch schief gehen?

- Alles!
- Pilottests:
  - Material und Werkzeuge testen
  - Datenspeicherung testen
  - Instruktionen für Probanden testen
  - ...
- Probanden **genau** sagen, was sie machen sollen
- Beobachten, dass Probanden **genau** das machen, was sie machen sollen
- Aufwärm Aufgabe, in der Probanden Ablauf lernen



# Ethik

- Aufwand der Probanden muss Erkenntnisgewinn gerecht werden
  - Evaluierung von Lehrmethoden
  - Evaluierung von Medikamenten
- Anonymität der Probanden sicherstellen
- Seid nett zu euren Probanden, denn sie investieren freiwillig ihre Zeit

# Lernziele

- Gute Hypothesen aufstellen können
- Experiment mit hoher interner oder hoher externer Validität entwerfen können



# Aufgabe

- Folgende Aussagen:
  - Programme in Java lassen sich leicht debuggen
  - Programmieren lernen geht am besten mit Haskell
  - Anfänger beginnen am besten mit Objektorientierung
- Legen Sie fest:
  - Hypothese
  - Abhängige und unabhängige Variablen und deren Operationalisierung
  - Störvariablen und deren Kontrolle
  - Experimentelles Design

# Hausaufgaben

- Warum brauchen wir Hypothesen?
- Was bedeutet Falsifizierbarkeit von Hypothesen und warum ist das wichtig?
- Recherchieren Sie andere Validitätsarten

# Literatur

- Jutta Markgraf, Hans-Peter Musahl, Friedrich Wilkening, Karin Wilkening, and Viktor Sarris. *Studieneinheit Versuchsplanung*, 2001. FIM-Psychologie Modellversuch, Universität Erlangen-Nürnberg.
- Natalia Juristo and Ana Moreno. *Basics of Software Engineering Experimentation*. Kluwer, 2001.
- Claes Wohlin. *Experimentation in Software Engineering*. Springer, 2000.
- William Shadish, Thomas Cook, and Donald Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin Company, 2002.
- James Goodwin. *Research in Psychology: Methods and Design*. Wiley Publishing, Inc., 1999.
- Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. *Selecting Empirical Methods for Software Engineering Research*. In *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer, 2008.
- Steve McConnell. *What does 10x Mean?* In *Making Software*, O'Reilly, 2010.
- Urban Wiesing. *Die Ethik-Kommissionen – Neuere Entwicklungen und Richtlinien*. Deutscher Ärzte-Verlag, 2003.