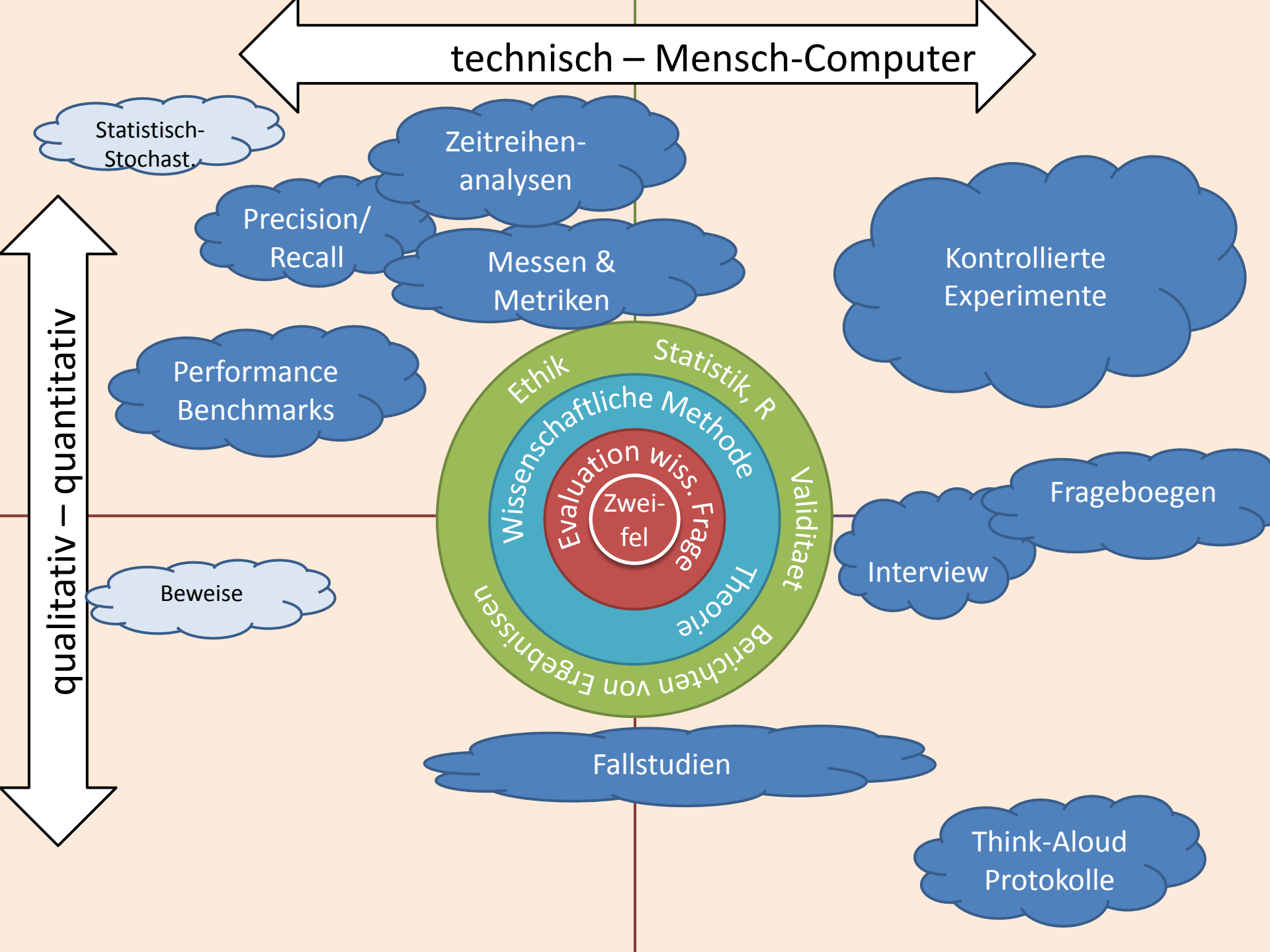
Empirische Methoden für Informatiker
Teil 4: Qualitative Untersuchungen

Christian Kästner


technisch – Mensch-Computer
qualitativ – quantitativ
Statistisch-Stochast.
Zeitreihen-analysen
Precision/ Recall
Messen & Metriken
Kontrollierte Experimente
Performance Benchmarks
Ethik
Statistik, R
Validitaet
Theorie
Berichten von Ergebnissen
Wissenschaftliche Methode
Evaluation wiss. Frage
Zwei-fel
Frageboegen
Interview
Beweise
Fallstudien
Think-Aloud Protokolle

Agenda

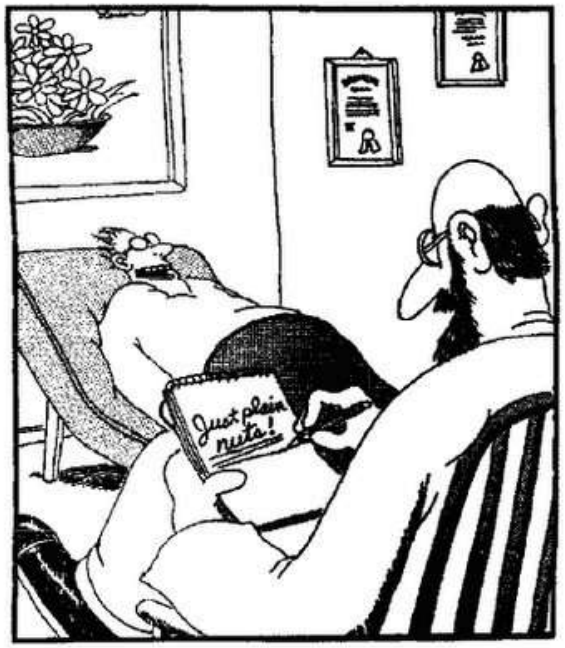
- Fallstudien
- Think-Aloud-Protokolle
- Interviews
- Wert und Probleme Qualitativer Ansaetze

Laboruntersuchung vs. Felduntersuchung

- Konstanthalten von Drittvariablen im Labor
  - “Quicksort ist schneller als Mergesort bei den Daten X auf Computer Y wenn implementiert mit Z von V’.”
  - Zuverlaessige Messung der abhaengigen Variablen (hohe interne Validitaet)
  - Nicht verallgemeinerbar auf andere Belegungen der Drittvariable (geringe externe Validitaet)
  - Aus praktischen und ethischen Gruenden nicht immer moeglich
- Untersuchung im Feld, Drittvariablen nicht immer kontrollierbar
  - Hohe externe Validitaet
  - Geringe interne Validitaet

=> Kompromiss

Qualitative Methoden



- Interpretation von verbalem Material
- Fokus auf Erfahrung
- Offene Befragungen
- “Mehr Details als ein Messwert”
- Realismus statt Laborbedingungen

- Keine statistischen Signifikanztests
- Mehr Zeitaufwand
- Schwer vergleichbar

Oberflaechliche Abgrenzung

Quantitativ

- “Naturwissenschaftlich”
- Labor
- Erklaeren
- “Harte Methoden”
- Messen
- Stichprobe
- Zahlen
- Abstraktion

Qualitativ

- “Geisteswissenschaftlich”
- Feld
- Verstehen
- “Weiche Methoden”
- Beschreiben
- Einzelfall
- Texte, Bilder
- Komplexitaet


schlecht
akzeptabel
gut
sehr gut
das Beste, was
ich je erlebt habe

Qualitative und quantitative Methoden

- Kombination qualitativer und quantiativer Methoden typisch

- Aus Texten/Erfahrungen Daten extrahieren
- Abstraktion
- Quantiative Inhaltsanalyse
- Erfordert ggf. strukturierte Interviews

- Statistische Analyse der gewonnen Daten

Fallstudien (Einzelfallbeobachtungen)

Fallstudie

- Detailierte Untersuchung eines einzigen Beispiels (oder weniger einzelner Beispiele)

- Beispiel
  - Anwenden des neuen Compilers auf ein Beispielprogramm
  - Aendern der Implementierung eines Programms so dass es ein neues Designpattern nutzt
  - Beobachten eines Entwicklers beim Stellen einer Anfrage, beim Interagieren mit der IDE, beim Lesen in der Hilfe, …
- Oft Minimalanforderung an Diplom-/Masterarbeit

Analyse eines Problems

- Beobachten von Entwicklern
  - Lesen von Kommentaren/Hilfe
  - Aendern von fremdem Quelltext
  - Formulieren von Anfragen in neuer Sprache
- Analysieren von Quelltext
  - z.b. Fehler in Open-Source Programm
  - Kommentare in industriellem Quelltext
- Nachgehen von Berichten aus der Praxis
  - “NoSQL ist viel besser”

Evaluieren neuer Methoden

- Anwenden einer neuen Methode
  - Vom Autor selbst auf eigenem Beispiel
  - Vom Autor selbst auf bestehendem Beispiel
  - Von Drittem auf eigenem Beispiel
  - Von Drittem auf bestehendem Beispiel
  - Von neutralem Drittem auf bestehendem Beispiel
  - Kontrolliertes Experiment

Fallstudien zur Theoriebildung

- Pilotstudie, Erkundungsexperiment
- In fruehen Phasen der Untersuchung
- Zum Bilden von Theorien (die dann z.B. quantitativ untersucht werden)

Fallstudien und Quantitative Methoden

- Innerhalb einer Fallstudie Messungen moeglich
  - z.B. Geschwindigkeitsvorteil durch neuen Datenbankindex
  - Inferenzstatistik fuer Hypothesen ueber diesen Fall

- Kein Schluss auf allgemeine Faelle (externe Validitaet)

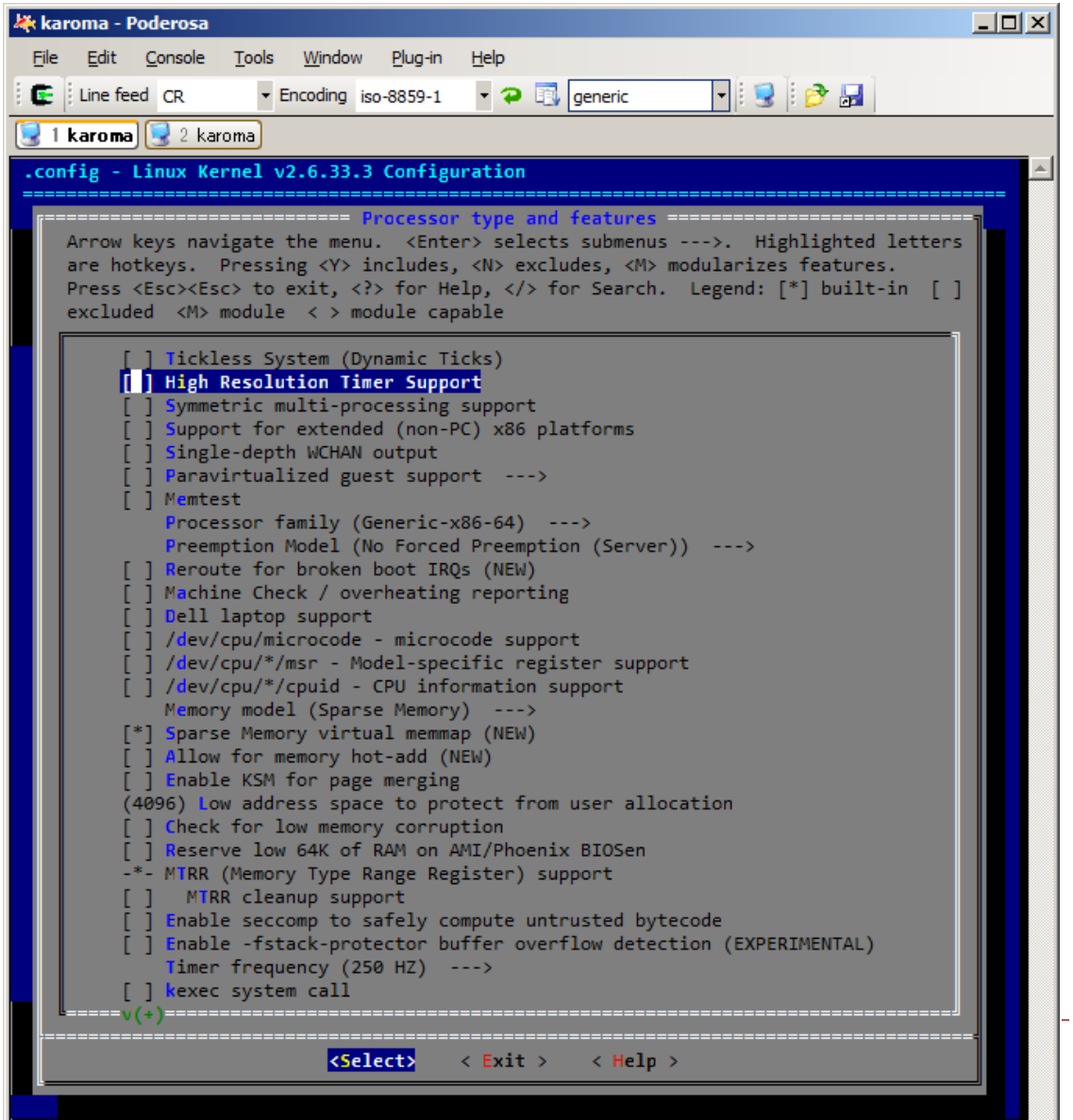
Beispiel: Berkeley DB

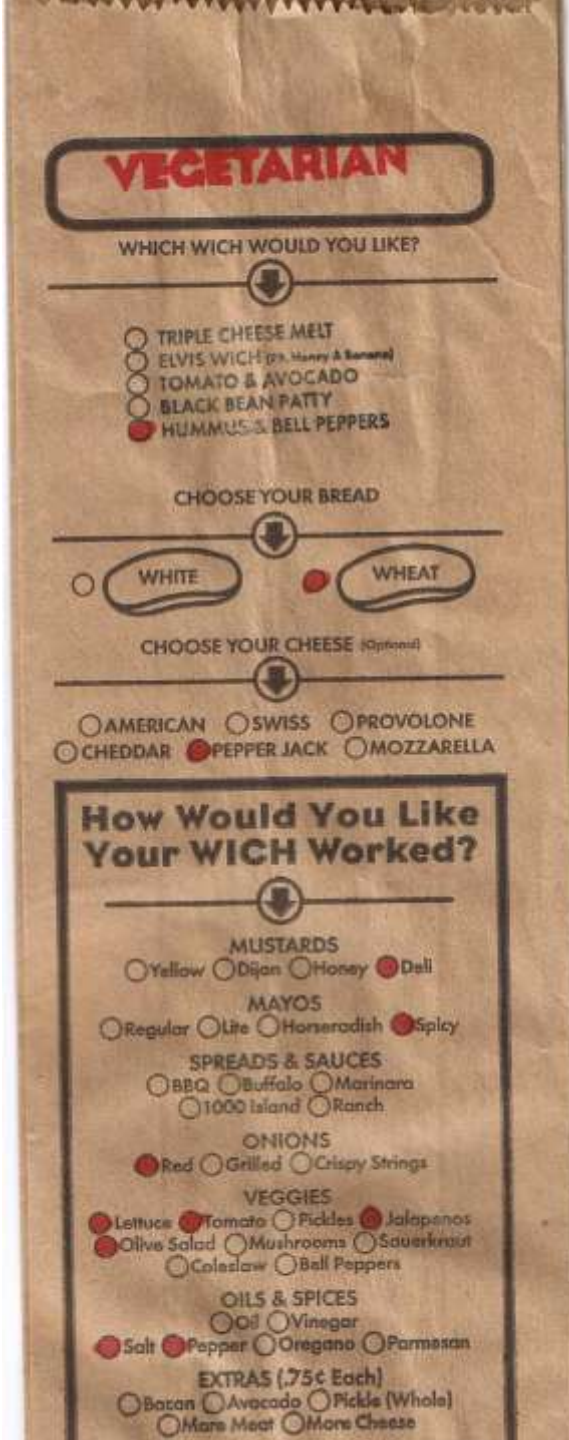
Fallstudie: Aspekte fuer Produktlinien

- Ausgangspunkt
  - Forscher schlugen AOP fuer Produktlinien vor
  - viele Publikationen, wenig Erfahrung
  - keine grossen Beispiele
- Idee
  - Umsetzen einer praktischen AOP Produktlinie
  - Zerlegung eines bestehenden Systems (statt Neuentwicklung)
  - Dadurch Realismus
  - (nur Benutzung bestehender Methoden)

Kästner, Apel, Don Batory. **A Case Study Implementing Features Using AspectJ**. In SPLC, pages 223-232. 2007.

Exkurs: Produktlinien



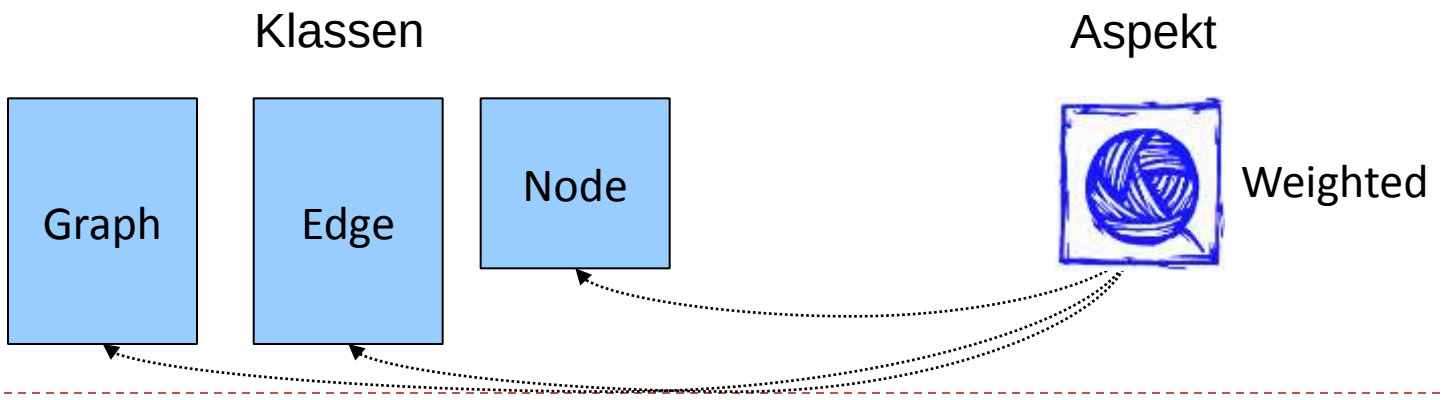


Exkurs: Bedingte Kompilierung

```
static int __rep_queue_filedone(dbenv, rep, rfp)
        DB_ENV *dbenv;
        REP *rep;
        __rep_fileinfo_args *rfp; {
#ifndef HAVE_QUEUE
        COMPQUIET(rep, NULL);
        COMPQUIET(rfp, NULL);
        return (__db_no_queue_am(dbenv));
#else
        db_pgno_t first, last;
        u_int32_t flags;
        int empty, ret, t_ret;
#ifdef DIAGNOSTIC
        DB_MSGBUF mb;
#endif
        // over 100 lines of additional code
}
#endif
```

Exkurs: Aspekt-orientierte Programmierung

- Modularisierung von einem querschneidenen Belang in einem Aspekt
- Dieser Aspekt beschreibt die Änderungen dieses Belangs in der restlichen Software
- Wird mitkompiliert oder nicht



Exkurs: AspectJ

Basic Graph

```
class Graph {
  Vector nv = new Vector();
  Vector ev = new Vector();
  Edge add(Node n, Node m) {
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m);
    ev.add(e); return e;
  }
  void print() {
    for(int i = 0; i < ev.size(); i++)
      ((Edge)ev.get(i)).print();
  }
}
```

```
class Edge {
  Node a, b;
  Edge(Node _a, Node _b) {
    a = _a; b = _b;
  }
  void print() {
    a.print(); b.print();
  }
}
```

```
class Node {
  int id = 0;
  void print() {
    System.out.print(id);
  }
}
```

Color

```
aspect ColorAspect {
  Color Node.color = new Color();
  Color Edge.color = new Color();
  before(Node c) : execution(void print()) && this(c) {
    Color.setDisplayColor(c.color);
  }
  before(Edge c) : execution(void print()) && this(c) {
    Color.setDisplayColor(c.color);
  }
  static class Color { ... }
}
```

Kontext der Fallstudie

- Beginn mit Grundwissen zu AspectJ, aber keine **praktische** Erfahrung
- Urspruenglich Fallstudie fuer andere Forschungsfrage
  - “Inwieweit ist die Reihenfolge von Aspekten relevant und kann geaendert werden?”
- Wenige verfuegbare grosse/praktische AOP-Quelltexte
- Neuschreiben moeglich aber aufwendig und verfaelschbar
- Daher: Refactoring bestehender Anwendung
  - Herausloesen von Funktionalitaet
  - Reimplementierung als optionaler Aspekt

Auswahl der Fallstudie

- Ein einziges Projekt: **Berkeley DB** Java Edition
- Eingebetette Datenbank (Bibliothek)
- Open Source von Sleepycat (heute Oracle)
- Wohlbekannte Domane
- Realistische Groesse (ca. 84000 Codezeilen, 300 Klassen)
  - aber nicht zu gross
- Realistisch als Produktlinie benutzbar (eingebetette Systeme)

Erfahrungen und Beitrag der Fallstudie

- Auf Probleme gestossen
  - Einige Probleme erwartet
  - Ausmass der Probleme hat ueberrascht
- Vorgehen und Erfahrungen (insb. Probleme) dokumentiert
- Fallstudie oeffentlich zugaenglich gemacht

Bericht zum Vorgehen

- Feature-Auswahl:
  - Repraesentative Auswahl von kleinen und grossen Features (ad-hoc)
  - Nach Dokumentation, Configurationsparametern, Domaenenwissen und Quelltext
- Infrastruktur
- Refactorings

Reproduzierbarkeit?

Refactoring	# times used
Extract Introduction (Method)	365
Extract Beginning/End	214
Extract Introduction (Field)	213
Create Hook Method	164
Extract Before/After Call	121
Move Class to Feature	58
Extract Method	15
Move Interface to Feature	4

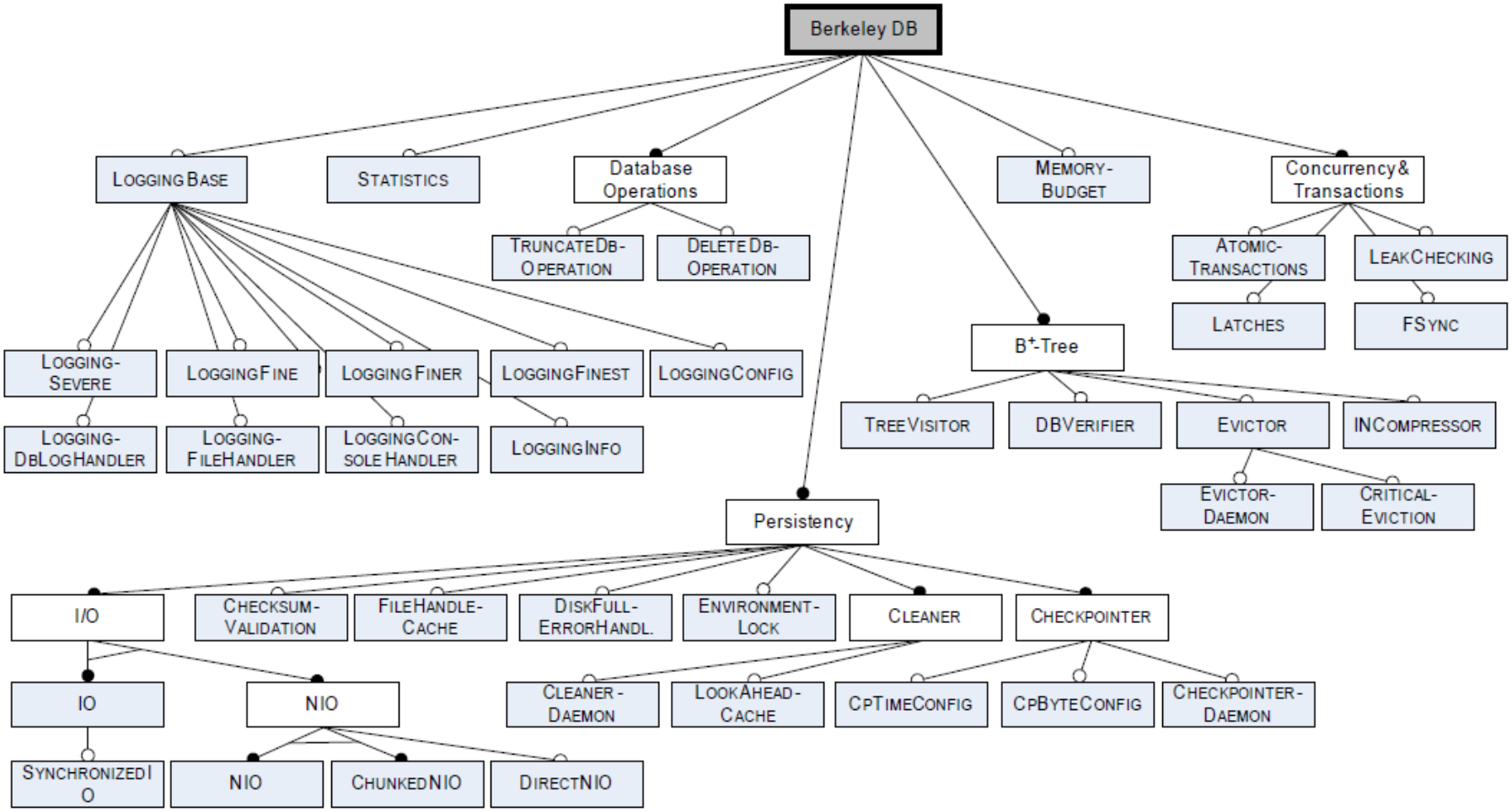
Table 2. Refactorings used in Berkeley DB.

Feature	LOC	EX	AT	Description
ATOMICTRANSACT.	715	84	19	Part of the transaction system that is responsible for atomicity.
CHECKPOINTERD.	110	14	4	Daemon to create checkpoints in the log.
CHECKSUMVALID.	324	32	8	Checksum read and write validation of persistence subsystem.
CHUNKEDNIO	52	2	1	Chunked new I/O implementations.
CLEANERDAEMON	129	9	2	Daemon to clean old log files.
CRITICALEVICTION	63	10	7	Evictor calls before critical operations to ensure enough memory.
CPBYTESCONFIG	41	7	5	Configuration options for the Checkpointer by size.
CPTIMECONFIG	59	6	5	Configuration options by time.
DBVERIFIER	391	16	10	Debug facility to verify the integrity of the B^+-tree.
DELETEDBOP.	226	31	13	Operation to delete a database.
DIRECTNIO	6	1	1	Direct I/O access.
DISKFULLERRORH.	41	4	2	Emergency operations on a full disc error.
ENVIRONMENTLOCK	61	5	2	Prevents two instances on the same database directory.
EVICTOR	371	20	9	Subsystem that evicts objects from cache for the garbage collector.
EVICTORDAEMON	71	9	3	Daemon thread that runs the Evictor when a memory limit is reached.
FILEHANDLECACHE	101	6	2	File handle cache.
FSYNC	130	5	1	File synchronization for writing log files.
INCOMPRESSOR	425	21	4	Removes deleted nodes from the internal B^+-tree.
IO	38	2	1	Classic I/O implementation.
LATCHES	1835	155	28	Fine grained thread synchronization.
	43	4	2	Debug checks for leaking transactions.
	1115	132	24	Debug logging facilities, separated in 10 features for different logging levels and handlers, not listed here.
LOOKAHEADCACHE	84	6	2	Look ahead cache for read operations.
MEMORYBUDGET	958	118	28	Observes the overall memory usage.
NIO	26	2	1	New I/O implementation.
STATISTICS	1867	345	30	Collects runtime statistics like buffer hit ratio throughout the system.
SYNCHRONIZEDIO	26	2	1	Synchronized I/O access.
TREEVISITOR	138	24	9	Provides a Visitor to traverse the internal B^+-tree.
TRUNCATEDBOP.	131	5	3	Operation to truncate a database.

AD - Pieces of advice; EX - Extensions (advice, introductions);
AT - Number of types affected by the feature.

Table 1. Refactored features of Berkeley DB.

Features in Berkeley DB



Beobachtungen

- Neue Sprachkonstrukte kaum verwendet
- Wenig querschneidende Belange
- Praktische Limitierungen der Sprache
  - Statement extensions, Local variable access, Exceptions, …
  - In Theorie weitgehend bekannt, hier praktische Konsequenzen
- Fragilitaet
- Lesbarkeit und Verstandlichkeit
  - Diverse Argumente, weitgehend subjektiv
- Werkzeuge

	# extended join points				
Category	2	3	4	> 4	$\sum$
Pattern expressions	5	1	1	0	7
Explicit enumeration	15	7	1	2	25
Homog. statement extensions	11	4	2	3	20

Used AspectJ Language Constructs

- Most used language constructs:
  - Static introductions (4 interfaces, 58 classes, 365 methods, and 213 fields)
  - Method refinements (execution, 214)
  - Statement extensions (call && within, 121)
  - Hook methods (164)
- Rarely used:
  - Advanced advice (if, cflow, etc.)
  - Homogeneous extensions

	# join points				
Category	2	3	4	>4	Σ
Pattern expressions	5	1	1	0	7
Explicit enumerations	15	7	1	2	25
Homog. statement extensions	11	4	2	3	20

Limitations of AspectJ

- Statement Extension Problem
  An emulation only, e.g., call && within
  Hook methods
- Local Variable Access Problem
  Unable to access local variables at a join point
- Exception Introduction Problem
  Signatures and exceptions unchangeable
- Scope Problem
  Often needed to publish private/protected classes or methods

Maintainability

- Aspects were fragile and hard to maintain
- Implicit coupling, extensions not visible in code
- Dependence on implementation details
- Pointcuts can break silently
- Tool dependence, but tools not designed for features and SPLs

Readability and Understandability

- Increased code size, repetitive
- Third person perspective
- Large features are hard to read and understand

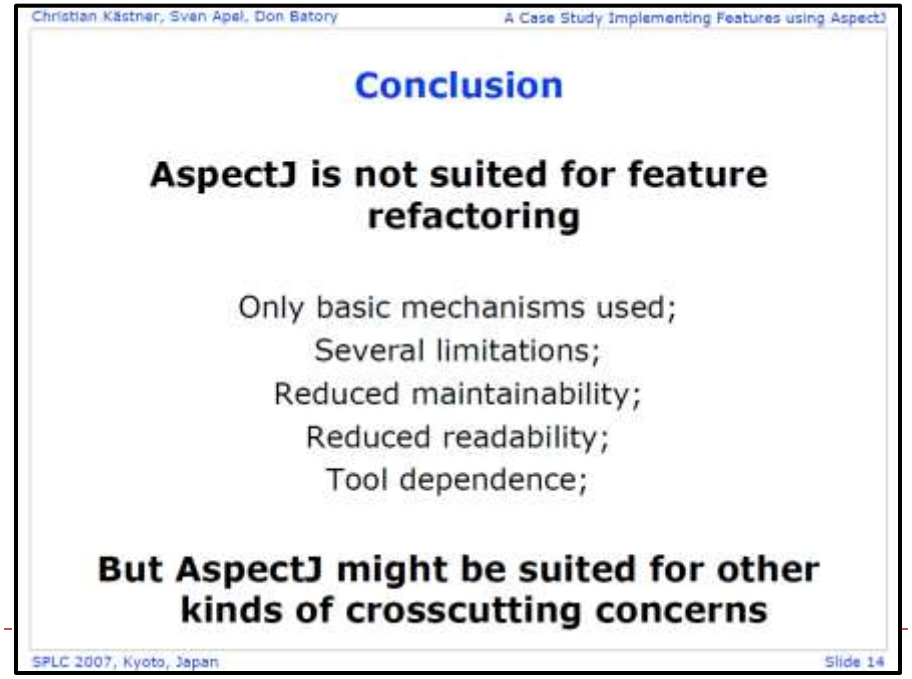
```
public void delete(Transaction txn, DbEntry key) {
  super.delete(txn, key);
  Tracer.trace(Level.FINE, "Db.delete", this, txn, key);
}
```

```
pointcut traceDel(Database db, Transaction txn, DbEntry key):
    execution(void Database.delete(Transaction, DbEntry))
    && args(txn, key) && within(Database) && this(db);

after(Database db, Transaction txn, DbEntry key):
  traceDel(db, txn, key) {
  Tracer.trace(Level.FINE, "Db.delete", db, txn, key);
}
```

Diskussion

- AspectJ geeignet? Alternativen?
  - Das passende Werkzeug fuer ein Problem
- Werkzeugunterstuetzung kritisch
- Inherente vs. Zufaellige Komplexitaet



Reflektion

- Ein einziger Fall
- Relativ aufwendig
- Realismus
- Viel gelernt, Erfahrung spaeter nuetzlich
- Keine statistischen Tests, keine Vergleiche
- Provokativ, widerlegte Hypothese
- Sehr spezieller Kontext
  - Zerlegung von bestehendem Quelltext
  - Beibehalten des Verhaltens
- Teils subjektiv (insb. Lesbarkeit, Wartbarkeit)

Bericht

- Konferenzpublikation
  - ca. 1 Seite Motivation und Kontext
  - ca. 2 Seiten Vorgehensbeschreibung
  - ca. 3.5 Seiten Erfahrungen und Probleme
  - ca. 1 Seite Diskussion
  - ca. 2.5 Seiten Zusammenfassung, Verwandte Arbeiten und Literatur
- Diplomarbeit nochmal ausfuehrlicher
- Viele Quelltextbeispiele und Tabellen

Diskussion
Vor- und Nachteile von Fallstudien

Kritik an Fallstudien

- Unkontrolliert und subjektiv -> unzuverlaessig
- Tendenz zur Bestaetigung bestehender Hypothesen
- Nicht verallgemeinerbar
- Viele Details, schwer zusammenfassbar

Lernen durch Fallstudien

- Betrachten eines Problems im Kontext
- Lernen aus Einzelfaellen
  - Regel-Lernen fuer Einsteigerlevel
  - Experten durch praktische Erfahrung
  - Probleme wirklich verstehen (learning by doing)
- Realistische Details
- Nicht abstrahiert/simplifiziert auf einfache Modelle
- Verhindert “Elfenbeinturm Forschung”

- Beweis kaum moeglich, aber lernen aus Erfahrungen

Fallstudien zum Falsifizieren

- Fallstudie kann eine Hypothese falsifizieren (-> exhaustion)
- Gut gewaehltes Beispiel kann reichen
- “Wenn schon einfache Beispiele nicht klappen…”
- Diskussion: Probabilistische Hypothesen

- Beispiel
  - Galileo Schwerkraftexperiment mit Fallbeispiel (Feder vs. Blei) statt Experimentserie
  - AOP fuer bekannte nichttriviale querschneidende Belange in Datenbanken

B. Flyvbjerg. **Five Misunderstandings About Case-Study Research**. Qualitative Inquiry. 12(2):219-245. 2006

Auswahl von Faellen

Auswahl	Beschreibung
Zufall	Reduziert Voreingenommenheit; eher Verallgemeinerbar (ggf. zufaellig aus Teilgruppe gewaehlt)
Extremer Fall	Ungewoehnlicher Fall; besonders problematisch oder besonders geeignet Illustriert einen Punkt sehr stark
Maximale Variation	Mehre sehr unterschiedliche Faelle (z.b. drei Faelle die sich durch Groesse/Sprache/Erfahrung unterscheiden)
Kritischer Fall	Erlaubt Schlussfolgerungen wie: “Wenn es hier (nicht) klappt, klappt es in allen Faellen (nicht)” z.B. zur Plausibilitaetspruefung einer Theorie
Paradigmatisch	Allgemeiner typischer Fall der von mehreren Forschern wiederverwendet wird; Theorien basieren auf diesem Fall

“Convenience sampling”

Auswahl von Fallstudien

- Auswahl von guten Fallstudien erfordert Erfahrung
- Abhaengig vom Zweck
  - Machbarkeit zeigen?
  - Maximales Potential einer Methode aufzeigen?
  - Praktische Anwendbarkeit demonstrieren?
  - Bestehende Meinung widerlegen?
  - Methoden vergleichen?

- Gilt auch fuer Auswahl von Benchmarks!

Mit Betreuer / Kollegen beraten!

Aufgabe

- Schlage moegliche Fallstudien fuer die folgenden Forschungsfragen vor:
  - Masterarbeit: Neue Compileroptimierung beschleunigt Java-Programme mit Floating-Point-Operationen
  - Bachelorarbeit: Neue Compileroptimierung beschleunigt Programme mit Floating-Point-Operationen; implementiert ist aber nur ein kleiner Teil von Java (z.B. ohne Strings)
  - Neue Speicherverwaltung in Linux unterstuetzt MultiCore Programme
  - Neue Eclipse-Funktionalitaet: Bereiche die haeufig editiert wurden werden mit Hintergrundfarbe markiert (Idee: sie sind tendentiell Fehleranfaelliger)
  - Webseiten sind einfacher zu navigieren wenn auf allen Seiten ein einheitliches Farbschema verwendet wird
  - Aufgrund der neuen Kantenglaettung sehen die Objekte mit dem 3D-Renderingverfahren realistischer aus
  - UML Modelle sind unnuetz und stoeren den Entwicklungsprozess
  - Ein neuer Datenbank-Index fuer mehrdimensionale Anfragen unterstuetzt Kartenanwendungen/Suche in Videos

Fallstudien erfordern Selbstreflektion

- Gefahr der Verfaelschung und Manipulation
  - Auswahl von sehr vorteilhaftem (trivialen) Fall
  - “Vergessen” von Problemen
  - Vereinfachende Annahmen
- Protokoll fuehren, eigene Arbeit kritisch ueberpruefen
- Erwartungen vor der Fallstudie und Hypothesen transparent machen

- In der Praxis tendieren Fallstudien zum Widerlegen von Hypothesen

Fallstudien zusammenfassen

- Fallstudienbeschreibungen oft lang, subjektiv und anekdotisch
- Oft nicht knapp zusammenfassbar
- Ursache: Komplexitaet von realen Faellen
- Erfahrungen im Kontext weitergeben
  - Aus Erfahrungen anderer lernen
  - Zusammenfassung nicht immer erwuenscht
- Details in Anhang

- Balance finden

Zusammenfassung Fallstudien

- Untersuchung einzelne Faelle
- Praxisnaehe, Lernen mit echten Faellen
- Wenig kontrolliert, bedingt verallgemeinerbar
- Auswahl der Fallstudie wichtig

- Fallstudien koennen praktische Erkenntnisse liefern
- Nur eingeschraenkt verallgemeinerbare Ergebnisse

Brainstorming und Focus Groups

Brainstorming und Focus Groups

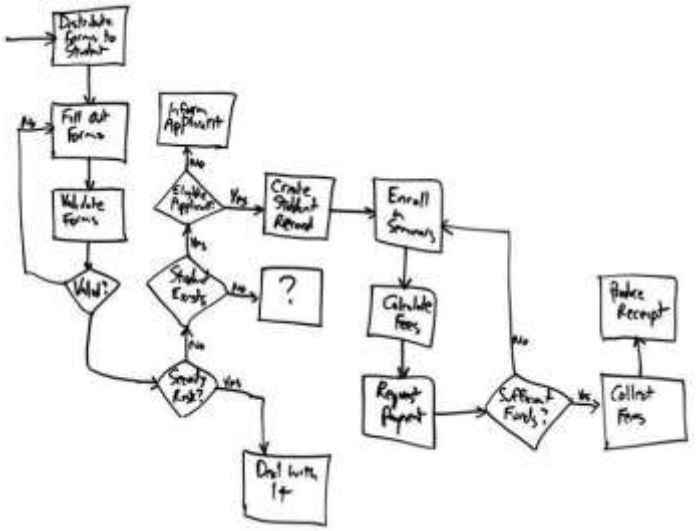
- Moderierte Gruppensitzung
- Beispielthemen
  - Welches sind die Haupttaetigkeiten im Tagegeschaeft?
  - Welche Features wuenschen Sie sich in IDEs?

- Insb. fuer neue Domaenen und fruehe Phasen geeignet
- Erfordert erfahrenden Moderator
- Terminprobleme
- Protokoll

Konzeptionelle Modellierung

- Befragte modellieren ein Konzept (e.g. Flow Chart, Architektur)
- Macht mentale Modelle explizit

- Oft schwer interpretierbar
- Qualitaetsschwankungen



Interviews

Forschungs- und Feldgespraeche

- Vertiefende Literatur lesen!
- Offene Fragen
- wenig Struktur vorgeben
- Reaktion auf Gesagtes, ohne zu beeinflussen
  - Reaktion des Befragten nicht beeinflussen
  - keine eigene Meinung zeigen
- Raum fuer Unerwartetes lassen, aufgreifen

- Muendlich vs. Schriftlich
- Auswahl von Interviewpartnern: Wie Auswahl von Fallstudien (Zufaellig/Begruendet)

Ablauf

- Inhaltliche Vorbereitung
  - Warum, Thema, Personen, ggf. spezifische Fragen
  - Interviewleitfaden erstellen
- Organisatorische Vorbereitung
  - Kontaktaufnahme, Diktiergeraet (+Ersatz)/Kamera/Skype
- Interview
  - Gespraechsbeginn + Aufbau
  - Durchfuehrung und Aufzeichnung
  - Gespraechsende + Nachgespraech + Verabschiedung
  - Gespraechsnotizen anfertigen

Interviewleitfaden (Beispiel)

- Erklärung des Projektes; Klärung von Unklarheiten seitens des Interviewten; Möglichkeit der Anonymität des Interviews erwähnen;
- Start der Tonbandaufnahme; **Projektname**, **Ort**, **Datum**, **Interviewer**, **Start** und **Ende** des Interviews
- Themen auf Interviewpartner abstimmen, aber grundlegende Fragen zu:
  - Aktuelle Aufgabe
  - Verfahren um Informationen zum Problem zu sammeln
  - Benutzte Resourcen (Dokumentation/Personen)
  - Neu Gelerntes in der letzten Woche, neue Werkzeuge
- Zuerst möglichst frei erzählen lassen, anschließend noch nicht erwähnte Themen zur Sprache bringen. Zwischenfragen erwünscht, sollten aber den Erzählfluss nicht zu sehr stören.

In Anlehnung an: Susan Sim, Richard Holt: *"The Ramp-Up Problem in Software Projects: A Case Study of How Software Immigrants Naturalize"*, 20th Intl. Conf. on Software Engineering, pp.361-370, 1998

Dokumentation

- Transkription
  - Zeitaufwaendig
  - ca. 1 Seite Text pro Minute
- Archivierung des Materials
  - 10 Jahre (DFG Richtlinie)
- Datenschutz
  - Anonymisierung
  - Vernichtung/Rueckgabe des Rohmaterials

I: was würden sie sagen, was so für sie im leben wichtig ist?

D: * gesundheit * dass ich meine arbeit behalte, das ist für mich ganz wichtig, weil, weil, äh erstens mal, bin ich dadurch, dass ich arbeit habe, selbständig, ja, kann mir bestimmte finanzielle wünsche erfüllen, die ich sicherlich nicht könnte, wenn ich fn// wenn ich arbeitslos wäre,

I: hm

D: außerdem ist es mein traumberuf inzwischen jeworden, kindergärtnerin, dass ich sehr was vermissen würde, wenn ich in dem beruf nicht mehr arbeiten könnte,

I: hm

D: mir würde och der kontakt zu=n kollegen und zu den eltern fehlen, weil man ja zusehr im eigenen saft dann sicherlich schmort, ja was, * dass meine kinder * doch recht glücklich aufwachsen in dem land,

I: hm

D: ja, ich äh * bin zum beispiel och, äh, was heißt, meine kinder wissen, dass ich in der pds bin, sie wissen, dass ich 'n bisschen im wohngebiet mitarbeite, die versammlungen besuche, aber dass ich zum beispiel, wenn hier infostände sind in buch, ich da nicht mitmache, sondern,

Andere Interviewformen

- Gruppenbefragungen
- Narratives Interview
- Oral History
- u.v.m

- siehe Literatur

Bortz und Doering. **Forschungsmethoden und Evaluation fuer Human- und Sozialwissenschaftler** (Kapitel 5). 4 Auflage. Springer, 2006.

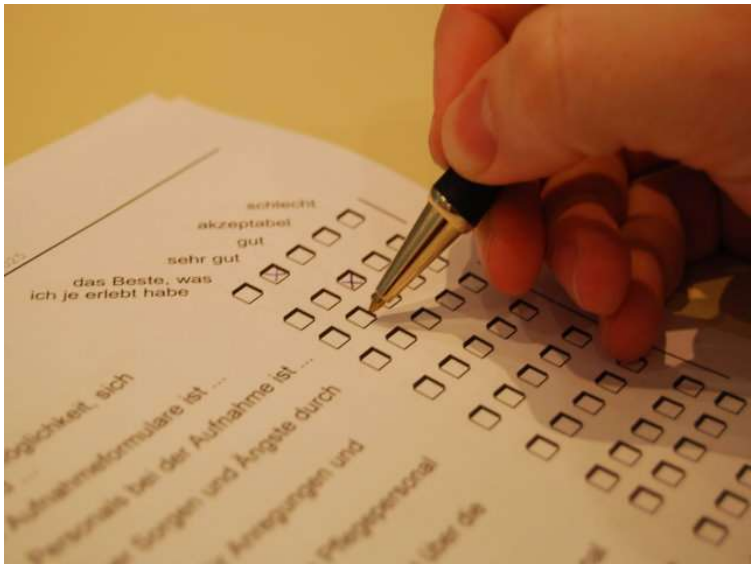
Auswertung

- Text und Quellenkritik (Qualitaet des Materials)
- Auswahl von Teilaspekten oder einzelnen Aussagen (Zufall/systematisch)
- Kategoriensystem, Kodierung (Merkmale identifizieren)
- Vergleich und Zusammenfassung von Einzelfaellen
- Immer methodisch-begruendetes Vorgehen
- Kompakte Repraesentation (vgf. Fallstudien)

- ggf. quantitative Daten gewinnen

Frageboegen

- Geringe Kosten
- grosse Zielgruppen moeglich
- Vorformulierte Fragen, ggf. missverstaendlich



- mehr dazu spaeter (quantitativ)

Selbstbeobachtung von Probanden beim Aufgabenloesen

Tagebuch

- Entwickler protokollieren Erfahrung in Tagebuch
- Time Sheets (teils sowieso vorhanden)
- selbststaendig oder computergestuetzt (z.B. Popups zu zufaelligen Zeitpunkten)

- Fortfuehrung statt Retrospektive
- Genauigkeit?
- Aufwand?

Think-Aloud-Protokolle

- Denkwege bei Aufgabenloesung laut erzaehlen
- Protokoll / Aufzeichnung
- Neutraler Interviewer
  - greift nicht ein
  - fordert zum Reden auf
- insb. Validierung kognitiver Modelle

Shadowing

- Forscher als “unsichtbarer” Beobachter im Raum
- Beobachtet Probanden ueber Zeitraum, folgt ihm

- Keine Interaktion

- Ablaeufe nicht immer extern erkennbar

Ethnografie

- Teilnehmende Beobachtung
- Forscher schliesst sich Entwicklerteam an, aktive Teilname
- Beobachtung “von innen”

- Natuerlichere Interaktion, tiefes Verstaendnis
- Hoher Aufwand
- Gefahr Verlust des Abstands zum Untersuchungsgegenstand

nonparicipating observer -> nonobserving participant

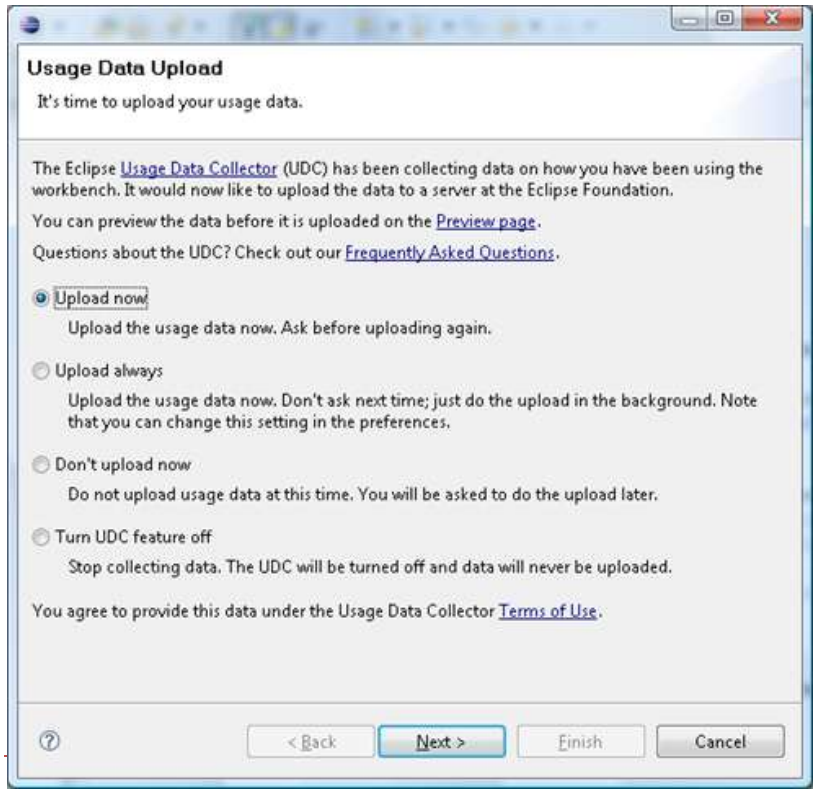
Kritik Selbstbeobachtung

- Selbstbeobachtung aendert das Verhalten (Reaktivitaet)
- “Testsituation”
- Natuerlichkeit/Realismus?

Indirekte Datenerhebung

Instrumentierung

- System protokolliert Benutzung (oder externes Protokollierungsprogramm)
  - Benutzterkommandos, Tastendruecke
  - Detailierte Ausfuehrungslogs
  - Playback-Logs
- Teils koennen vorhandene Logs verwendet werden



- Indirekte Erfassung, neutral
- Intention oft nicht erkennbar

Aufzeichnung (Fliege an der Wand)

- Probanden zeichnen selber ihre Bildschirmaktivitaet auf

- Guenstig, wenig invasiv
- Intention oft nicht erkennbar
- Aufwendige Auswertung

Nicht-Reaktive Verfahren

- Daten erheben ohne Probanden beeinflussen
- Offentliche Daten, verfuegbare Daten
- Ex-Post Analyse von Quelltext (ggf. open source)
  - Kommentare, Muster, Fehler
  - Anteil X in “neuem” Quelltext
- Analyse der Dokumentation
- Literaturanalyse (z.b. Buecher fuer Praktiker)
- Analyse von Commit-Meldungen
- Analyse von Netzwerken (z.b. Github)

Oft quantiativ ausgewertet, mehr dazu spaeter.

Beispielfallstudien aus Papern lesen

- Qualitative Analysen/Fallstudien
  - Griswold et al. **Exploiting the map metaphor in a tool for software evolution.** In Proc. ICSE, 2001 (Think aloud protocol)
  - Aldrich et al. **ArchJava: Connecting Software Architecture to Implementation.** In Proc. ICSE, 2002. (Fallstudie, technisch)
  - Mockus et al. **A Case Study of Open Source Software Development: The Apache Server.** In Proc. ICSE, 2000. (Fallstudie, reichhaltige Daten)
  - Cherubini et al. **Let's Go to the Whiteboard: How and Why Software Developers Use Drawings.** In Proc. CHI, 2007. (Fragebogen und Interviews)
- Bonus:
  - Herbsleb and Grinter. **Splitting the Organization and Integrating the Code: Conway's Law Revisited.** In Proc. ICSE, 1999. (Interviews)
- Ueberzeugend? Positives? Kritikpunkte? Validitaet?

Datenanalyse und Bericht

Datenanalyse

- Erhebung qualitativer Daten “befriedigend”, “interessant”

- Rigorose Analyse oft eher “unangenehm”
- “laestig/langweilig/aufwendiger als erwartet”

- Theorie generieren
- Theorie bestaetigen

Shull, Singer, and Sjogberg. **Guide to Advanced Empirical Software Engineering**. (Kapitel 2). Springer 2007.

Theorie generieren (grounded in data)

- Daten aus verschiedenen Perspektiven betrachten
- Constant Comparison Method
  - Offenes Codieren (Markierungen und Stichpunkte am Text)
  - Struktur in Codes erkennen, Subcodes/Kategorien
  - Codes zu Textfragmenten zuweisen
  - Axial Coding: Fragmente eines Codes zusammen lesen
  - Sense Making: Vorlaeufige Theorie als Memo formulieren
- Cross-Case Analyse
  - Vergleich Daten aus unterschiedlichen Faellen
  - Gemeinsamkeiten und Unterschiede aus vielen Perspektiven
  - Codes aus einem Fall in dem anderen verwenden

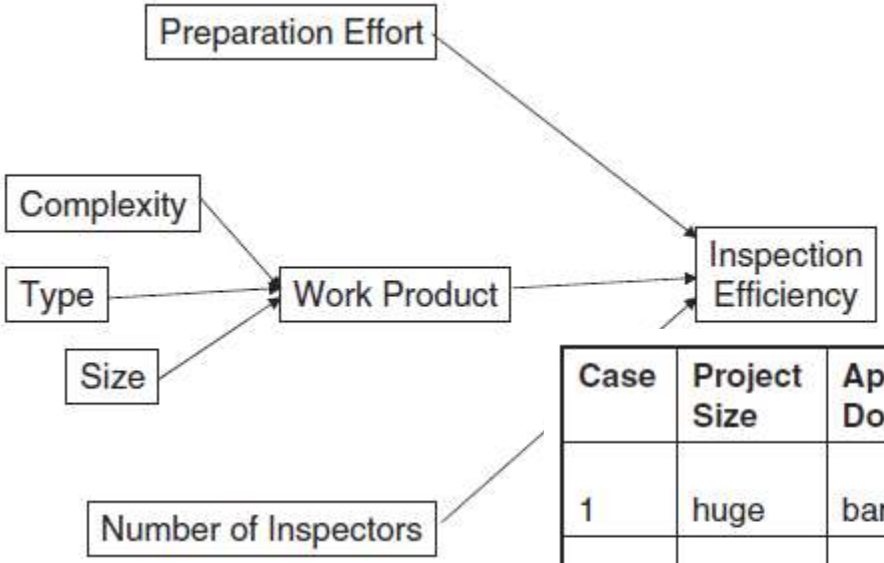
Theorie bestaetigen

- Zusaetzliche Belege sammeln (“weight of evidence”)
- Adressiert Zweifel an Validitaet

- Analyse weiterer “representativer” Faelle (Replikation)
- Analyse eines negativen Falls / extremer Faelle (outlier)
- Ergebnisse mit Subjeken diskutieren (member checking)
- Validitaet diskutieren

Daten visualisieren

- Diagramme, Karten, Tabellen



Case	Project Size	Application Domain	Experience of Developers	Meetings
1	huge	banquing	mixed	"We spend way too much time in meetings"
2	small	banquing		"We try to touch base with the whole team as often as we can"
3	small	aerospace	low	"The daily briefings are really useful, although some people say it interrupts their 'real' work"
4	large		high	"We would all be so much more productive if we could somehow get rid of meetings"
5	medium	communications	high	"People don't like to come to meetings, but I guess most of them are useful"

Validitaet

Guetekriterien qualitativer Datenerhebung

Diskussion: Wie kann man das Messen/sicherstellen?

- Objektivitaet
  - interpersonaler Konsens
  - genaue Beschreibung des methodischen Vorgehens
  - (Objektives Vorgehen verhindert nicht subjektive Meinung der befragten)
- Reliabilitaet strittig
  - Widerholbarkeit teils nicht moeglich/sinnvoll
- Validitaet
  - Interviewausserungen authentisch und ehrlich?
  - Protokolle echtes Abbild der Gespraeche?
  - Relevante Daten fuer Hypothese?
  - interpersonaler Konsens, inkl. Konsens Forscher und Befragte

Guetekriterien qualitativer Datenanalyse

- Gueltigkeit von Interpretationen (interne Validitaet)
  - Interpretation durch Daten gedeckt? Interpersonaler Konsens!
  - Mehere Erklaerungsmodelle? Diskutieren!
- Generalisierbarkeit von Interpretationen (externe Validitaet)
  - exemplarische Verallgemeinerung statt Inferenzstatistik
  - Repraesentative Erfahrungsbeschreibungen?
  - Gezielte statt zufaellige Stichprobe. Faelle nachtraeglich hinzugefuegt/ausgeschlossen…
  - Teils explizit Verzicht auf Verallgemeinerung
  - Ergaenzung durch quantiative Methoden

8 Strategien fuer bessere Validitaet nach Creswell

1. Triangulieren: Daten aus verschiedenen Quellen
2. Member Checking: Rueckfrage bei Probanden
3. Detailierte Beschreibung mit Kontextinformationen
4. Annahmen/Befangenheit/Neigung offenlegen (Bias)
5. Abweichende Informationen berichten
6. Laengerer Kontakt mit Probanden (tiefes Verstaendniss)
7. Praesentation vor bekannten Wissenschaftlern (Peers); fragen stellen lassen
8. Externer Audit

Diskussion und Zusammenfassung

Diskussion Qualitativer Methoden

- Vorteile
  - Realismus statt Labor
  - Mehr Details
  - Tiefes Verstaendnis
- Kritik
  - Vorwurf: Beliebig und Unwissenschaftlich
  - Interpretation von Einzelfaellen, Schluss auf Allgemeinheit
  - “weiche” Methoden
  - (vgl. Interpretation statistischer Ergebnisse)
- Aufwand

Qualitativer Methoden in der Informatik

- Fallstudien typisch
  - Zeigen Erfahrung, Versuch der Praxis
  - “gut genug”
  - Oft geringerer Aufwand
  - Geeignet fuer Abschlussarbeiten
- (Online) Frageboegen, Expertenbefragungen und Think-Aloud-Protokolle typisch
  - Nutzer befragen
  - Grounded Theory als “Trend”
  - Teils geeignet fuer Abschlussarbeiten
- Meist Kombination mit quantiativen Methoden

User Evaluations in Software Engineering Research

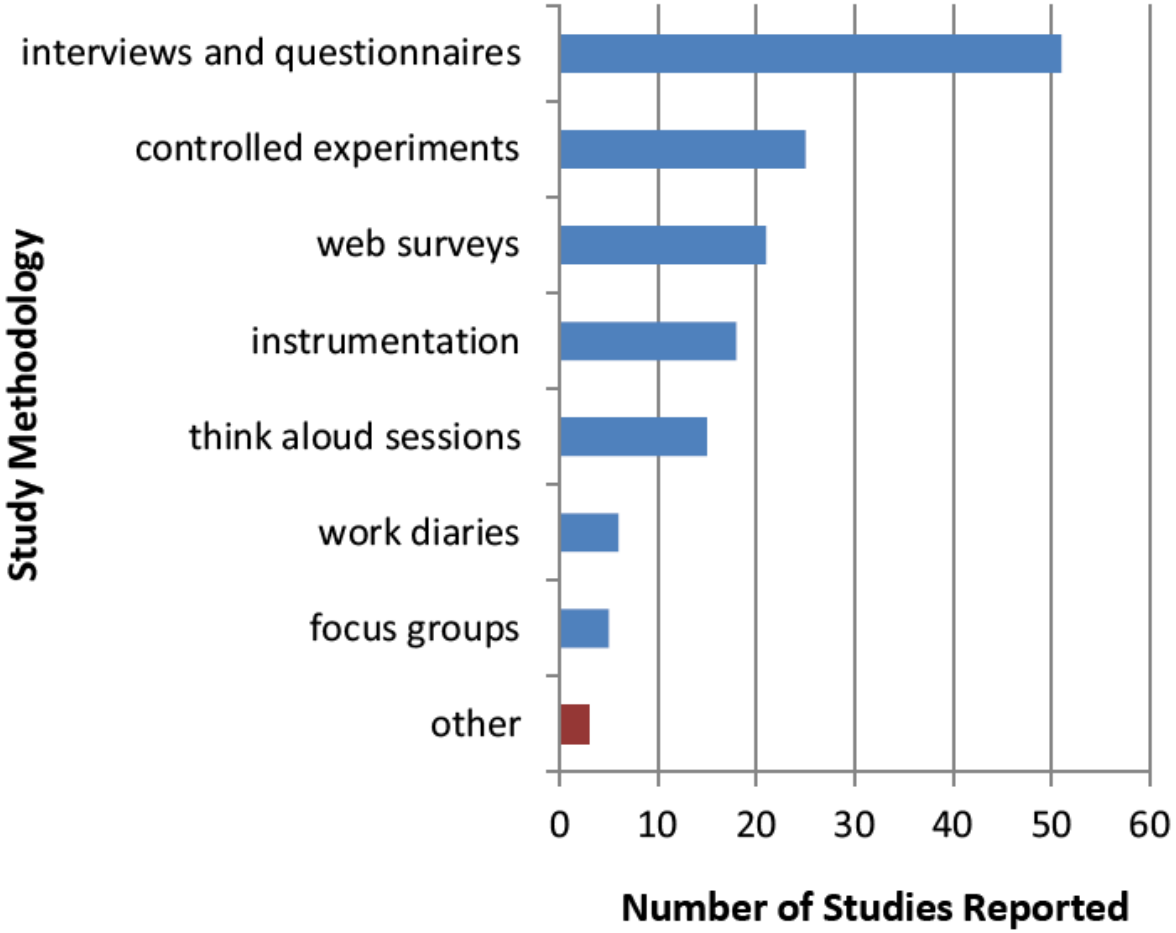


Figure 14. Methodologies employed in the last user evaluation.

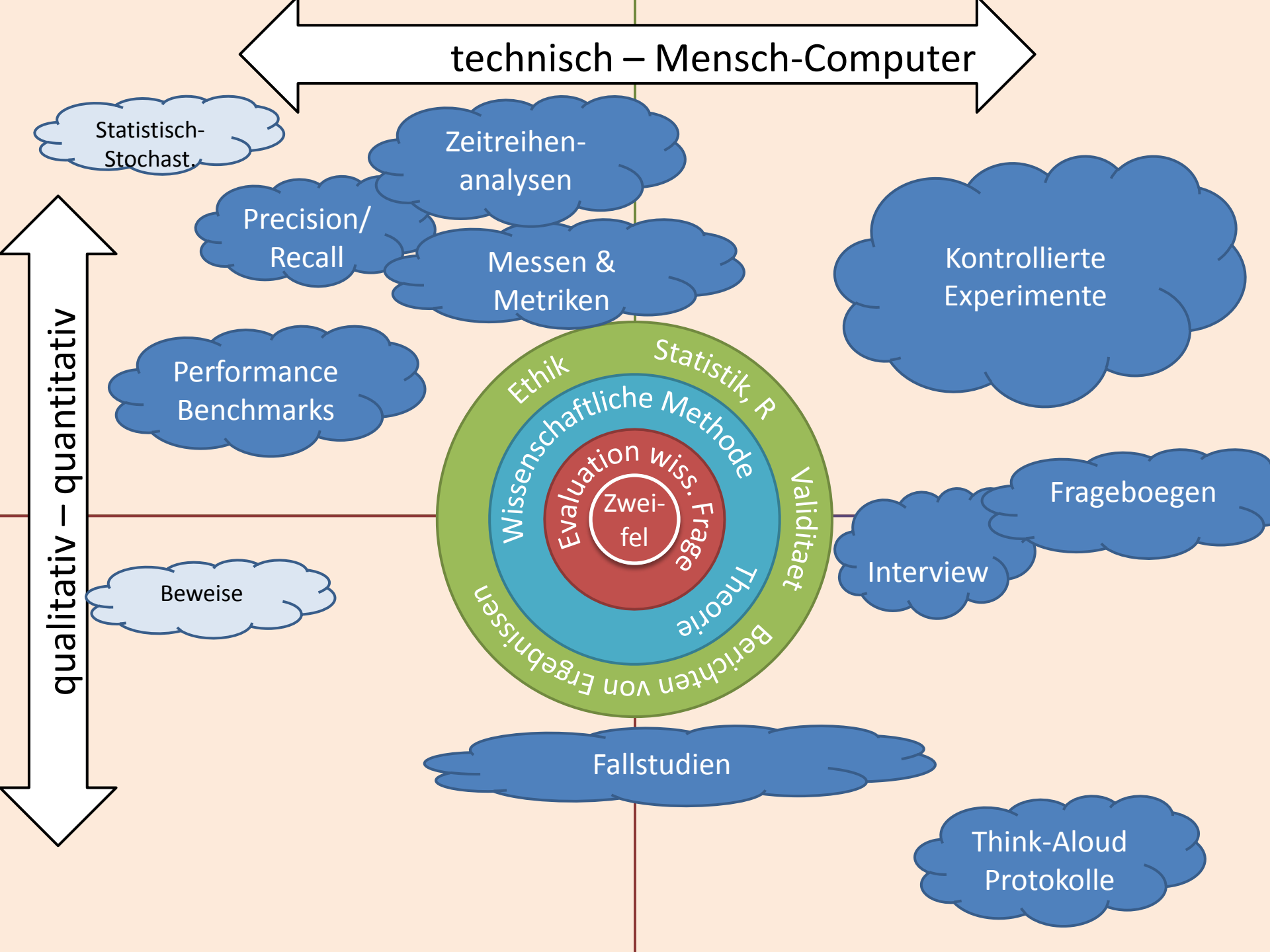
Buse, Sadowski, and Weimer. **Benefits and barriers of user evaluation in software engineering research**. In OOPSLA. ACM. 2011

Aufgaben

- Entwerfen Sie eine qualitative Evaluierungsstrategie fuer folgende Hypothesen
  - Sind Quelltextkommentare, externe Dokumentation, oder Methoden- und Variabilenbenennung wichtiger fuer das Verstaendniss bei der Wartung von Software
  - Die neue Multi-Touch Benutzeroberflaeche ist intuitiv bedienbar
  - Die Spracherweiterung Generics/Closures von Java wird von Entwicklern akzeptiert
  - Quelltextfragmente aus dem Internet werden oft direkt uebernommen
  - C Entwickler schreiben mehr Zeilen Quelltext am Tag als Ruby Entwickler. Woran liegt das?
  - Welche Konzepte haben sich bei der Gestaltung von Grafikprogrammen bewaehrt?
  - Wie organisieren erfolgreiche Entwicklerteams ihren Quelltext/ihre Zusammenarbeit?
- Diskutieren Sie die Validitaet ihrer Untersuchung
- Welche Vorteile/Grenzen bieten qualitative Methoden gegenueber quantitativen?

Literatur

- Bortz und Doering. **Forschungsmethoden und Evaluation fuer Human- und Sozialwissenschaftler** (Kapitel 5). 4 Auflage. Springer, 2006.
- Shull, Singer, and Sjogberg. **Guide to Advanced Empirical Software Engineering**. (Kapitel 1-4). Springer 2007.
  - Viele Beispiele und weiterfuehrende Referenzen
- B. Flyvbjerg. **Five Misunderstandings About Case-Study Research**. Qualitative Inquiry. 12(2):219-245. 2006
- Kästner, Apel, Don Batory. **A Case Study Implementing Features Using AspectJ**. In SPLC, pages 223-232. 2007.


technisch – Mensch-Computer
qualitativ – quantitativ
Statistisch-Stochast.
Zeitreihen-analysen
Precision/ Recall
Messen & Metriken
Kontrollierte Experimente
Performance Benchmarks
Ethik
Statistik, R
Validitaet
Theorie
Berichten von Ergebnissen
Wissenschaftliche Methode
Evaluation wiss. Frage
Zwei-fel
Frageboegen
Interview
Beweise
Fallstudien
Think-Aloud Protokolle