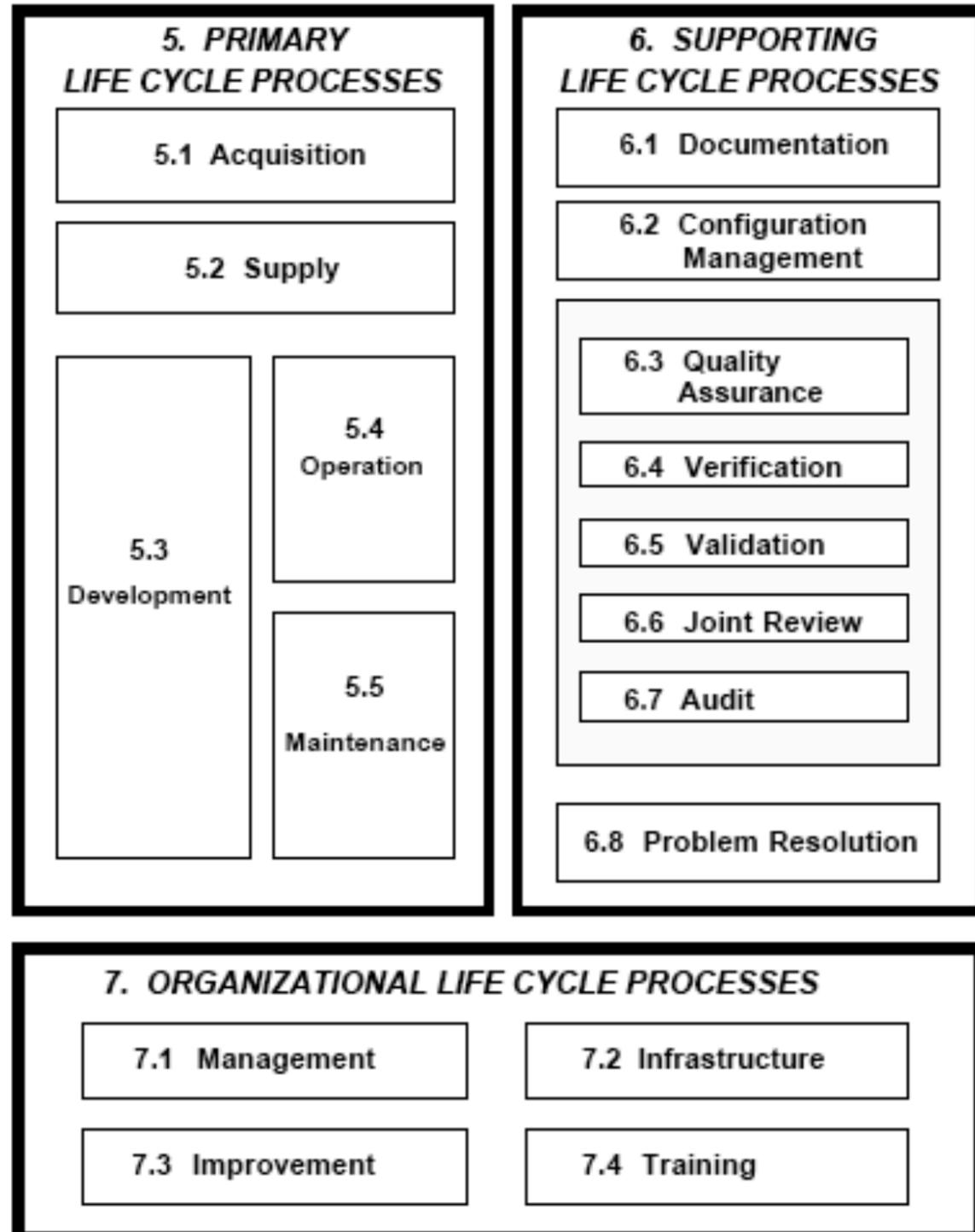




Dimensionen der Software-Entwicklung

Seminarkonzeption

Software-Entwicklung



SE-Lebenszyklus: IEEE 12207

Primary Life Cycle

- 1) Acquisition process;
- 2) Supply process;
- 3) Development process;
- 4) Operation process;
- 5) Maintenance process.

3 Development Activities

- 1) Process implementation;
- 2) System requirements analysis;
- 3) System architectural design;
- 4) Software requirements analysis;
- 5) Software architectural design;
- 6) Software detailed design;
- 7) Software coding and testing;
- 8) Software integration;
- 9) Software qualification testing;
- 10) System integration;

- 11) System qualification testing;
- 12) Software installation;
- 13) Software acceptance support.

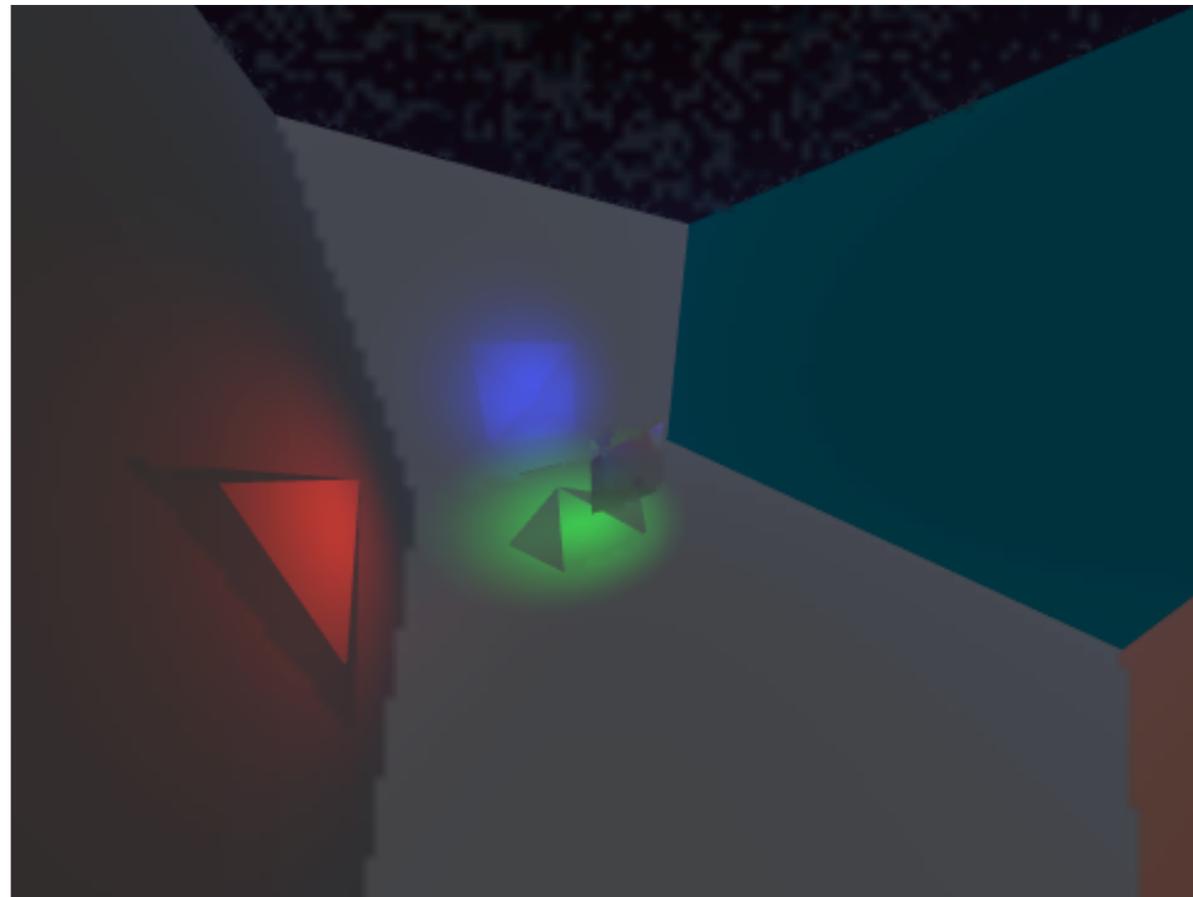
Supporting Life Cycle

- 1) Documentation process;
- 2) Configuration management process;
- 3) Quality assurance process;
- 4) Verification process;
- 5) Validation process;
- 6) Joint review process;
- 7) Audit process;
- 8) Problem resolution process.

Organizational life cycle processes

- 1) Management process;
- 2) Infrastructure process;
- 3) Improvement process;
- 4) Training process.

Dimension



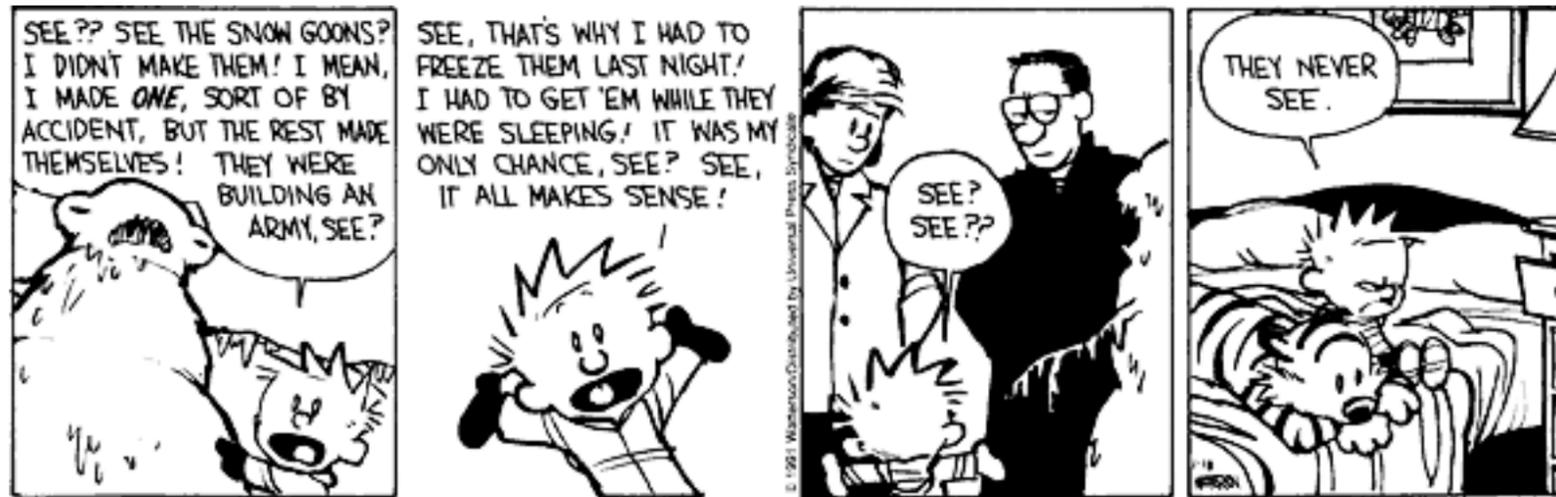
Das Vorhandensein einer Dimension im Horizont eines Weltbildes äußert sich **kognitiv** in Begriffen und Urteilen, **affektiv** in Motivation und Volition und **pragmatisch** in Handlungen.

Urteilskraft



»Urteilskraft überhaupt ist das Vermögen, das Besondere als enthalten unter dem Allgemeinen zu denken. Ist das Allgemeine (die Regel, das Prinzip, das Gesetz) gegeben, so ist die Urteilskraft, welche das Besondere darunter subsumiert bestimmend. Ist aber nur das Besondere gegeben, wozu sie das Allgemeine finden soll, so ist die Urteilskraft bloß reflektierend«

»Gedanken ohne Inhalt sind leer, Anschauungen ohne Begriffe sind blind.«

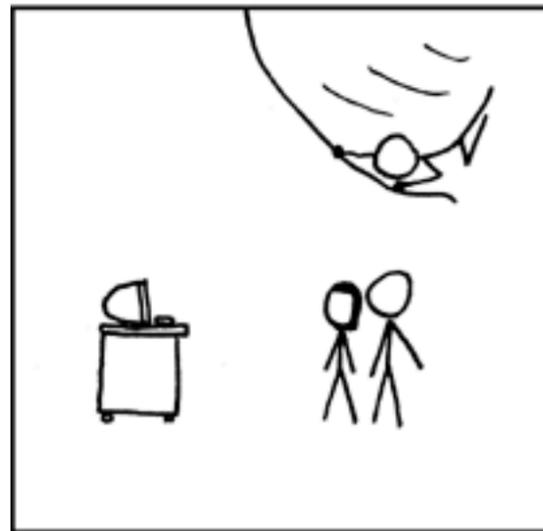
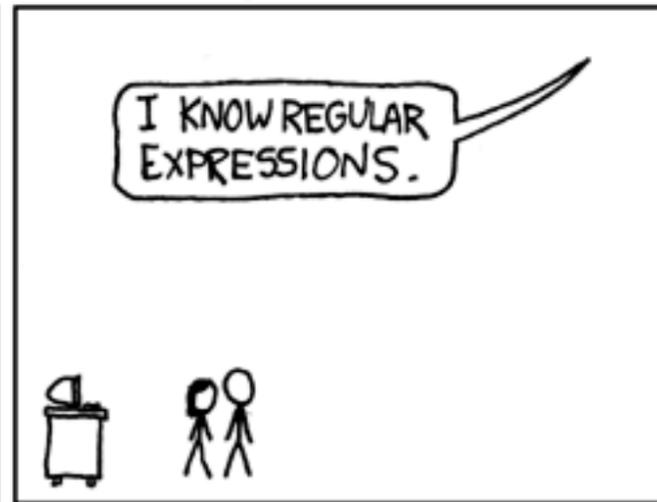


Vernunft

Rationalität ist »die Fähigkeit des Begründens (rationem reddere), wobei mit „Begründung“ nicht mehr als das Beantworten von Warum-Fragen gemeint ist und die Antworten in der Form von „Weil ...“-Sätzen erwartet werden; in diesem Sinne ist Rationalität die Fähigkeit, einem Faktum irgendeinen Grund hinzuzufügen (facti aliquam rationalem afferre).«

H. Schnädelbach

WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.



Handlungskompetenz

Dimensionen der Software-Entwicklung

Ästhetik

Bildung

Didaktik

Erkenntnistheorie

Ethik

Gender

Geschichte

Gesellschaft

Kultur

Ökonomie

Organisation

Politik

Psychologie

Raum/Zeit

Recht

Kontexte

Outsourcing

Projektmanagement

Software-Ergonomie

Sicherheit



Didaktik

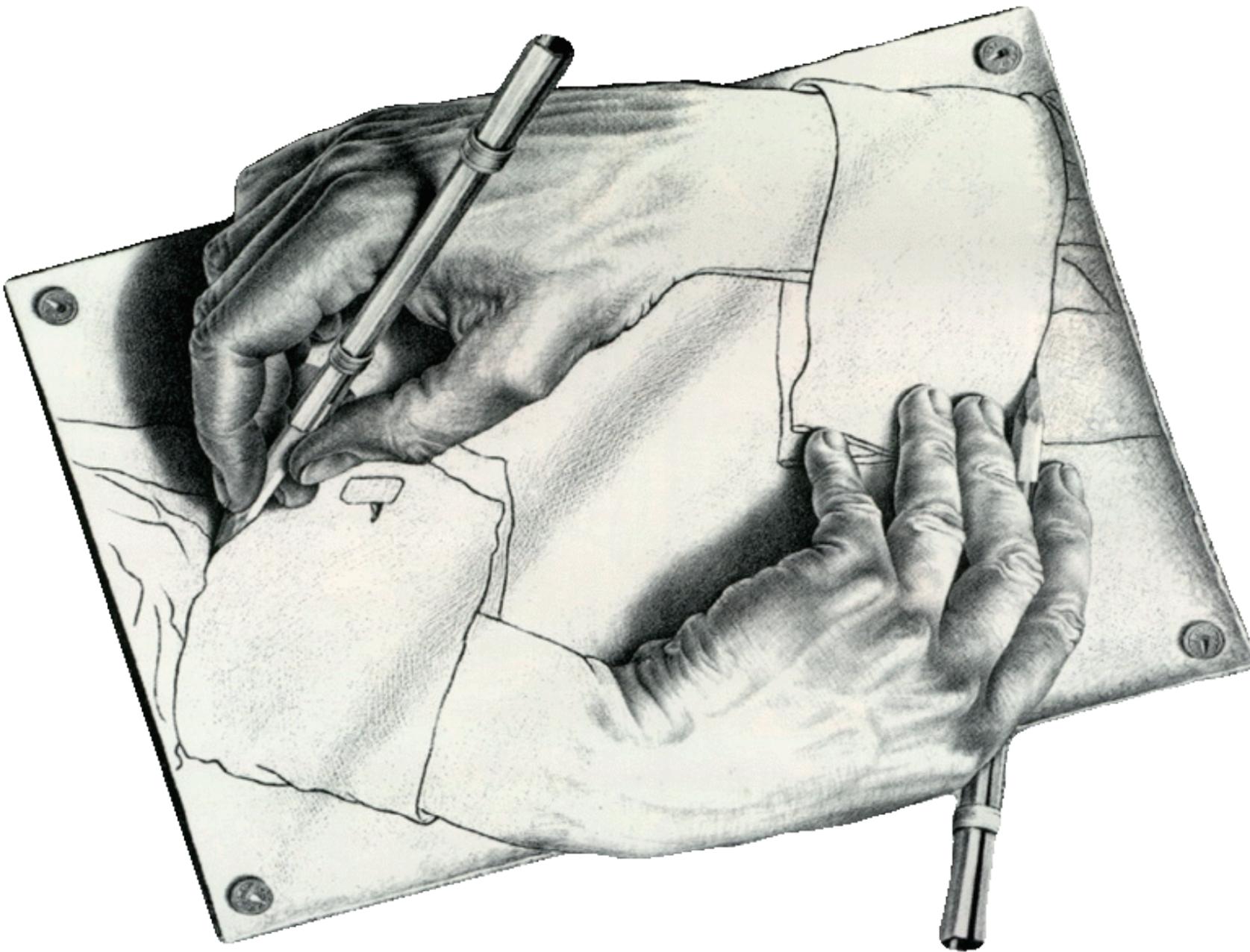
Humbert: *Didaktik der Informatik.*

Tomayko; Hazzan: *Human Aspects of Software Development.*

SEUH: *Software-Entwicklung im Unterricht der Hochschulen.*

So Kinder, Wir sind immer noch bei den Grundlagen des Formatierens.
Wer wiederholt nun bitte den Stoff, den ihr mir gestern
beigebracht habt? Mir ist das alles ein bisschen zu schnell gegangen.

Erkenntnistheorie



Pasch: *Software-Entwicklung im Team.*

Floyd: *Softwareentwicklung als Realitätskonstruktion.*

Floyd et al.: *Software Development and Reality Construction.*

Milewski; Clayton: *Cognitive & Cultural Diversity and Software Design.*

Winograd; Flores: *Understanding Computers and Cognition.*

Naur: *Programming as Theory Building*

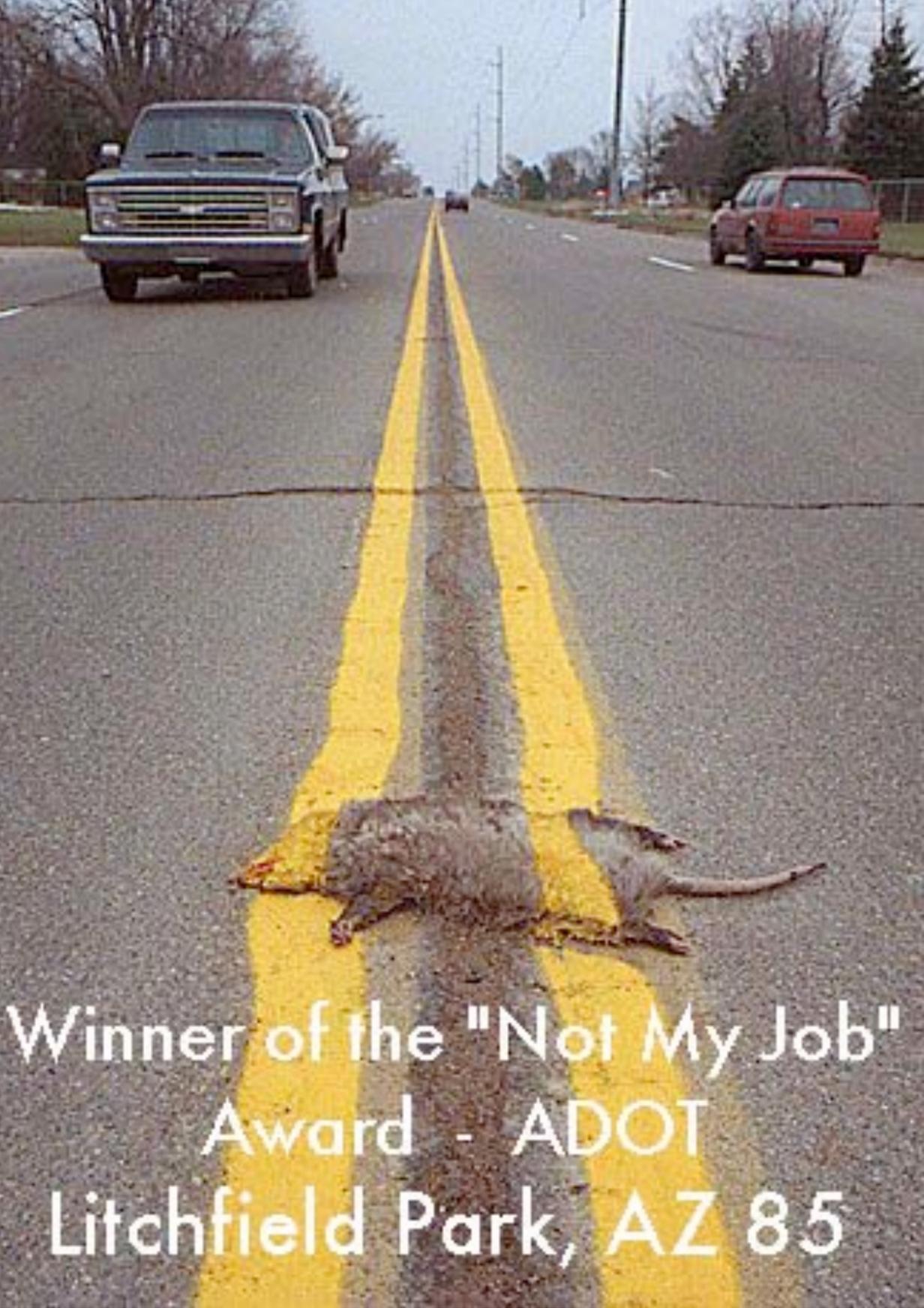
Bråten: *Model Monopoly*

Ethik

Koubek: *Software-Modellierung und Ethik.*

Fehling; Jahnke: *Wirtschaftsinformatik und Ethik.*

Scheffé: *Ohnmacht der Ethik?*



Winner of the "Not My Job"
Award - ADOT
Litchfield Park, AZ 85

Gender

Schinzel: *Das unsichtbare Geschlecht der Neuen Medien.*

Schinzel: *Informatik -Genderforschung: Integraler Bestandteil oder verzichtbare Außensicht?*

Schinzel; Ruiz Ben: *Software-Entwicklung als Profession?*





Geschichte

Brennecke; Keil-Slawik: *History of Software Engineering.*

Naur; Randell (Ed.): *Software Engineering.*

Michael S. Mahoney: *Finding A History For Software Engineering*

Brooks: *The Mythical Man Month;*

Ewusi-Mensah: *Software Development Failures: Anatomy of Abandoned Projects.*

Chapmann: *In Search of Stupidity: Over Twenty Years of High Tech Marketing Disasters*

Sozial



Vigenschow: *Soft Skills für Softwareentwickler.*

Evans: *Achieving Software Quality through Teamwork.*

Bruns: *Kommunikative Kompetenz in der Informatik und die curricularen Konsequenzen*

Pasch: *Software-Entwicklung im Team.*

McConnell: *Code Complete*

Dittrich; Floyd: *Social Thinking – Software Practice.;*

Kultur

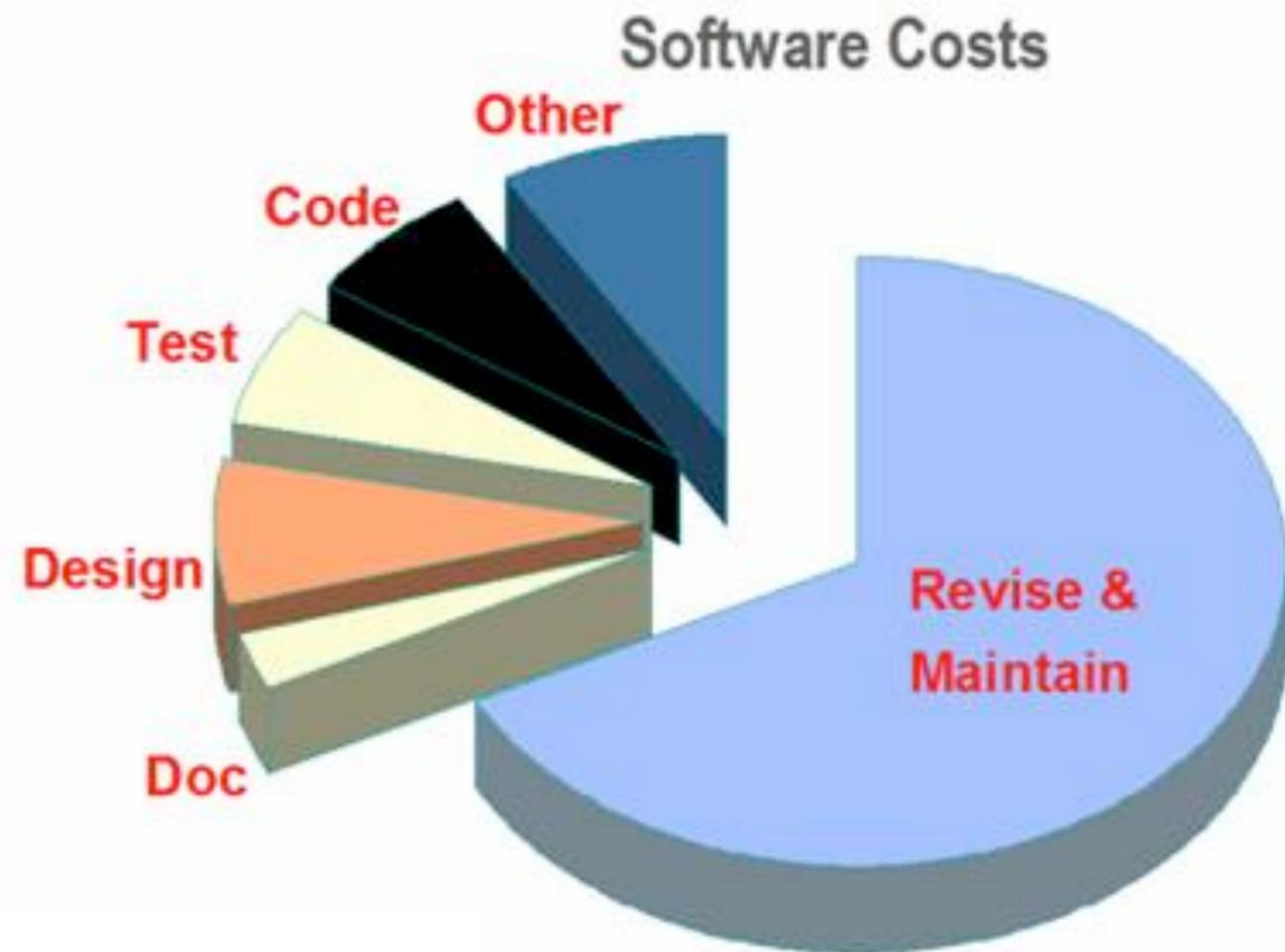


Sahay et al.: *Global IT Outsourcing;*

Erlbaum: *Usability and Internationalization of Information Technology*

Milewski; Clayton: *Cognitive & Cultural Diversity and Software Design.*

Ökonomie



Boehm: *Software Economics*

Boehm: *COCOMO II*

Boehm: *Software Development Cost Estimation Approaches – A Survey*

Organisation

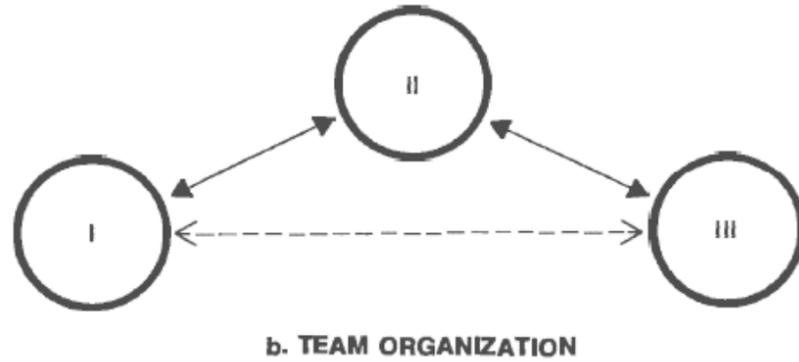


Figure 5-2 Program structure vs. team organization.

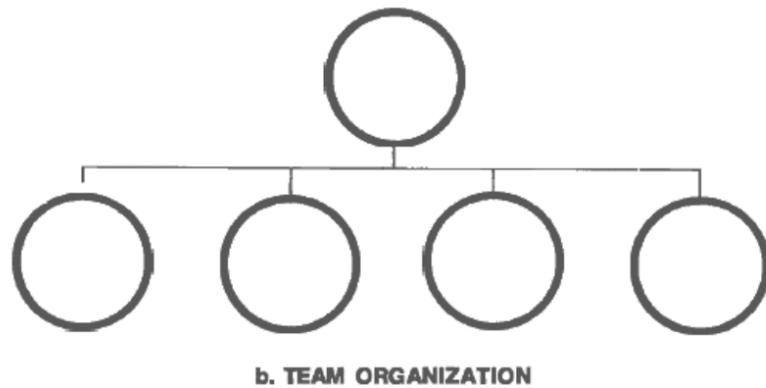
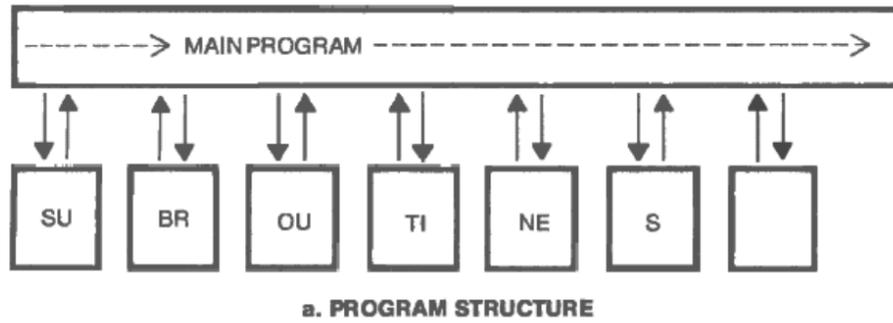


Figure 5-1 Program structure vs. team organization.

Conway: *How do committees invent?*

Krüger; Seelmann-Eggbert: *IT-Architektur Engineering.*

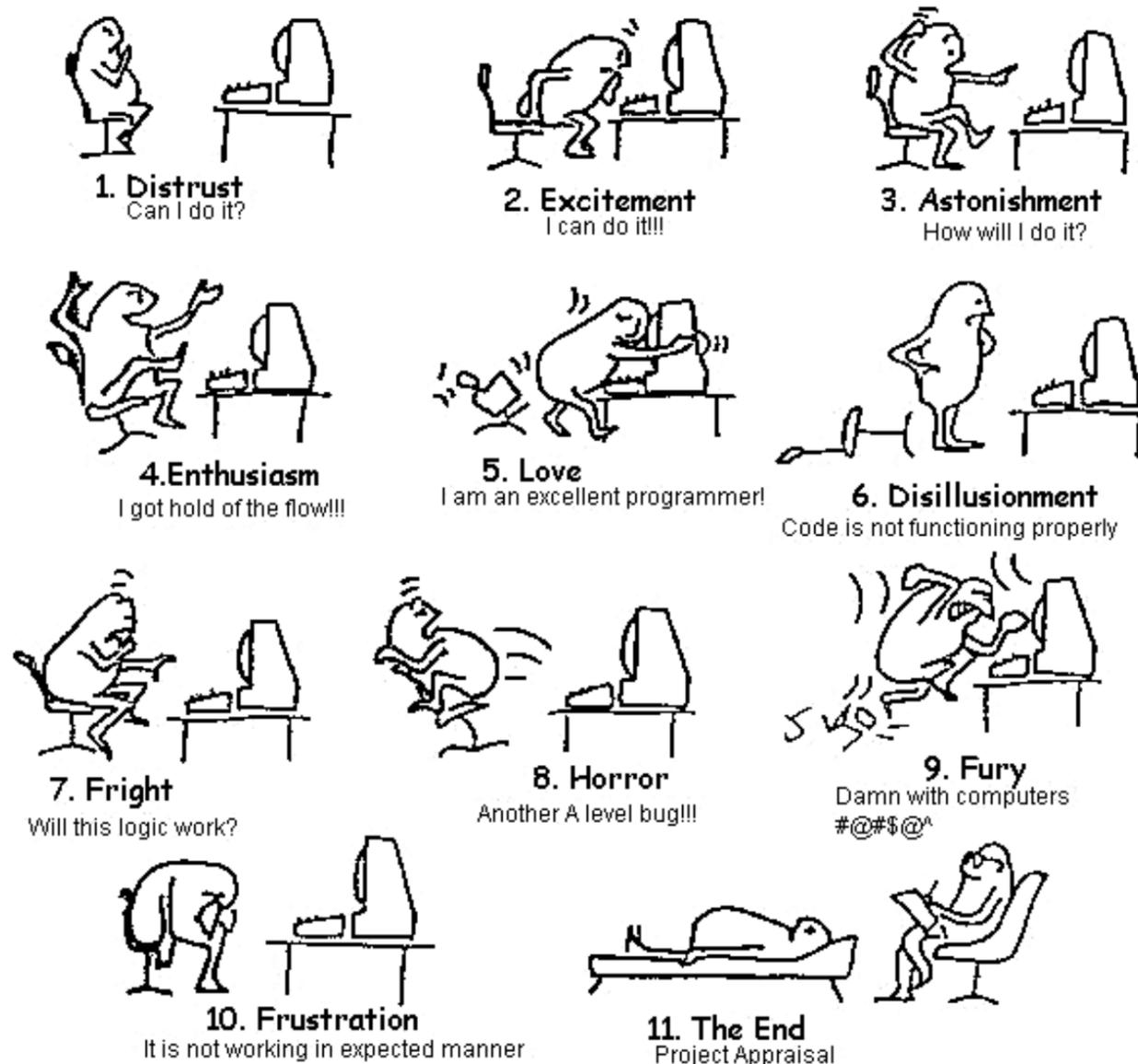
Weinberg: *Psychology of Computer Programming.*

Pasch: *Software-Entwicklung im Team.*

DeMarco: *Peopleware.*

Kieser; Ebers: *Organisationstheorien.*

Psychologie



Weinberg: *The Psychology of Computer Programming.*

Kaluzniacky: *Managing Psychological Factors in Information Systems.*

Dollery: *Understanding the Psychology of Programming.*

Michael Herczeg: *Differenzierung mentaler und konzeptueller Modelle und ihrer Abbildungen als Grundlage für das Cognitive Systems Engineering.*

Politik



© Scott Adams, Inc./Dist. by UFS, Inc.

Barley; Kunda: *Gurus, Hired Guns and Warm Bodies*

Adams: *Das Dilbert-Prinzip*

Peter: *Das Peter-Prinzip*

Parkinson: *Parkinsons Gesetz*

DeMarco: *Peopleware*

Raum/Zeit



DeMarco: *Peopleware.*

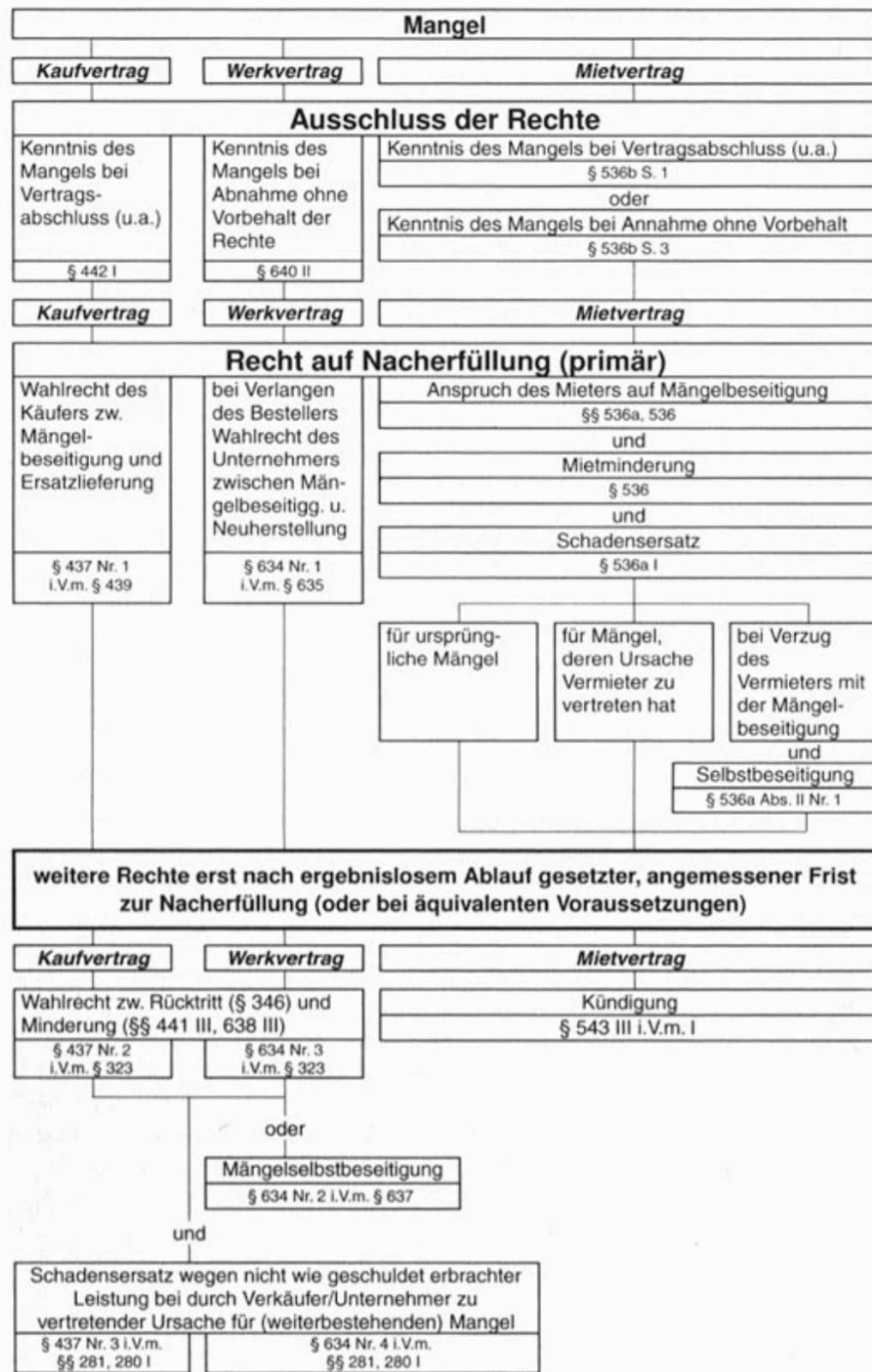
Williams; Kessler: *Pair Programming Illuminated.*

Hall: *The Hidden Dimension.*

Hall: *The Silent Language.*

McCue: *IBM's Santa Teresa Laboratory – Architectural design for program Development.*

Recht



Beck (Hg.): *Computerrecht*

Otto: *Recht für Software-Entwickler*

Seminararbeit

Iteratives Vorgehen mit folgenden Aktivitäten

1. Recherche
2. Vorbereitung der Untersuchungsmethode (z.B. Interview, teilnehmende Beobachtung)
3. Kontaktaufnahme (Feldzugang)
4. Forschung
5. Auswerten
6. Dokumentieren



1. Recherche

Die oben angegebenen Quellen dienen als Einstieg. Weitere Quellen stehen jeweils in der Bibliographie.

Internetquellen sind ok, ihre Zuverlässigkeit muss aber begründet werden.

Bücher kann man in einer Bibliothek bestellen. Da stehen auch Zeitschriften.

Artikel der Wikipedia sind keine wissenschaftliche Quelle. Die guten verweisen aber auf welche.

2. Vorbereitung der Untersuchung

Teilnehmende Beobachtung ist eine Feldstrategie, die gleichzeitig **Dokumentanalyse**, **Interviews** mit Interviewpartnern und Informanten, **direkte Teilnahme**, **Beobachtung** und **Introspektion** kombiniert. Hier sind das Eintauchen des Forschers in das untersuchte Feld, seine Beobachtung aus der Perspektive des Teilnehmers, aber auch sein Einfluss auf das Beobachtete durch seine Teilnahme wesentliches Kennzeichen. Man unterscheidet drei Phasen, nach denen sich auch der Projektverlauf orientiert:

(1) **Deskriptive Beobachtung** dient zu Beginn der Orientierung im Untersuchungsfeld. Sie wird dazu genutzt, die Komplexität des Feldes möglichst umfangreich zu erfassen.

(2) In **fokussierender Beobachtung** verengt sich die Perspektive zunehmend auf die für die Fragestellung besonders relevanten Prozesse und Probleme, während die

(3) **selektive Beobachtung** gegen Ende der Erhebung mehr darauf gerichtet ist, weitere Belege und Beispiele für die im zweiten Schritt gefundenen Typen von Verhaltensweisen und Abläufen zu finden.

3. Feldzugang

»Guten Tag, mein Name ist ..., ich studiere Informatik an der Humboldt-Universität zu Berlin und arbeite gerade an einer Seminararbeit über die Einflüsse von ... auf Software-Entwicklung. Für die Arbeit benötigen wir Beispiele aus der Praxis. Und da wollte ich fragen, ob es möglich ist, ein Interview mit einem Ihrer Entwickler zu führen?«



Verband der Software-, Informations- und Kommunikations-Industrie
in Berlin und Brandenburg



<http://www.sibb.de/Verband/Mitglieder/index.html>

4. Forschung

Feldtagebuch führen.

Auch Kleinigkeiten so bald wie möglich notieren, ehe sie in Vergessenheit geraten.

Gerade Fachsprache, unternehmensinterne Wortwendungen und Bezeichnungen oder Produktnamen machen die Forschung lebendig.

Möglichst viele Dokumente sammeln, um offizielle und inoffizielle Selbstdarstellung vergleichen zu können.

5. Auswertung

Beschreiben und Zusammenfassen der Ergebnisse mit den theoretischen Konzepten, die durch die Notizen gleichzeitig auf den Prüfstand gestellt werden: Lässt sich die Erfahrung mit den gewählten Kategorien strukturieren oder sind evtl. neue erforderlich?

6. Dokumentation: Gliederung der Seminararbeiten

Vorschlag

1. Einleitung.
2. Beschreibung des gewählten Projekts.
3. Analyse des Einflusses der gewählten Dimension auf den Projektablauf: Vorstellen der Begriffe und theoretischen Positionen, die zu dieser Analyse notwendig sind.
4. Ggfs. weitere Beispiele aus der Literatur anfügen.
5. Verallgemeinerungen auf spezifische Kompetenzen, die zur Wahrnehmung und Gestaltung von Phänomenen der gewählten Dimension benötigt werden.
6. Quellenangaben

Informatisches Modellieren

Modelle erstellen und bewerten

Die Schülerinnen und Schüler

- beschreiben Modelle als vereinfachtes Abbild der realen Welt,
- untersuchen, variieren und erweitern einfache Modelle,
- hinterfragen und bewerten Ergebnisse einer Modellbildung kritisch,
- wenden das Basiskonzept der objektorientierten Sichtweise auf Objekte von Standardsoftware an.

Mit Information umgehen

Information in Form von Daten darstellen und verarbeiten

Die Schülerinnen und Schüler

- unterscheiden zwischen Information und Daten,
- nutzen zielgerichtet bereitgestellte Informatiksysteme, digitale Datenbestände und Datenbanken,
- verwenden eigenständig die integrierten Hilfesysteme,
- wenden elementare Datentypen (integer, real, char, boolean, string) an.

6.5. Kompetenzen

»In Übereinstimmung mit Weinert verstehen wir unter **Kompetenzen** die bei Individuen verfügbaren oder von ihnen erlernbaren kognitiven Fähigkeiten und Fertigkeiten, bestimmte Probleme zu lösen, sowie die damit verbundenen motivationalen, volitionalen und sozialen Bereitschaften und Fähigkeiten, die Problemlösungen in variablen Situationen erfolgreich und verantwortungsvoll nutzen zu können.«

Klieme, Eckhard: Zur Entwicklung nationaler Bildungsstandards, S. 59.