Science of Design

Kei Ishii

WI + GE – Winter 2006/07

26. Januar 2007

Wikibook WI+GE
Zeitplan

Fr. 17.11.2006	Vorlesung
Mo 20.11.2006	Verbesserte Gliederung nach ersten Recherchen ins Forum stellen
Fr 1.12.2006	Vorlesung
Mo 11.12.2006	~~Erstes Einspielen in Wikibooks~~
Fr 15.12.2006	Vorlesung
Fr 12.1.2007	Vorlesung
Mo 22.1.2007	Bewertung der Gruppen (Inhalte um 18 Uhr)
Fr 26.1.2007	Vorlesung
Mo. 5.2.2007	Inhalte um 18 Uhr: Grundlage für Bewertung
zw. 5.2. und 9.2.	Einspielen der Texte in Wikibooks
Fr 9.2.2007	mündliche Rücksprache, Notenvergabe

Semesteraufgabe
Wikibook WI+GE

Termine
9. Februar 2007

- Wahlverfahren Rücksprachetermine:
  - 3 Terminvorschläge pro Gruppe (1./2./3. Wahl)
  - Per Mail (übliche Adresse) bis Mi 24. Jan., 23:59
  - Wenn keine Mail -> Zuweisung
  - Bei Überschneidungen -> Losverfahren am 26.1.

Zeitplan
9. Februar 2007

Zeit	Gruppe
0830-0845	G14
0845-0900	G15
0900-0915	G12
0915-0930	G17
0930-0945	ARA
0945-1000	G16
1000-1015	G13
1015-1030	G10
1030-1045	G18
1045-1100	

Zu Plagiaten
TU Fakultät IV, 17.1.2007

1. Die Studierenden werden zu Beginn jeder Lehrveranstaltung über die Problematik von Plagiaten und deren Konsequenzen hingewiesen.

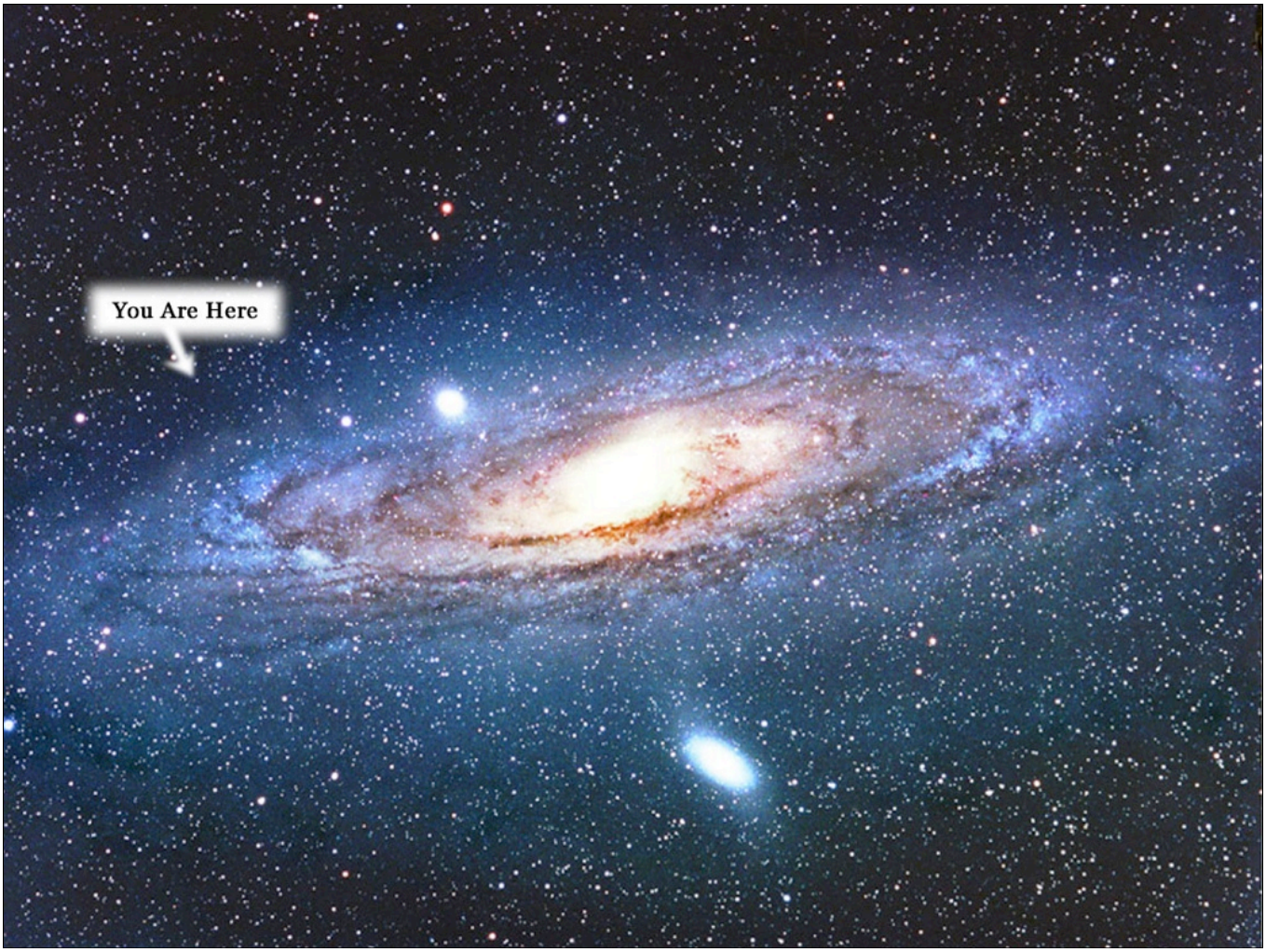
2. Jeder in der Fakultät abgegebenen wissenschaftlichen Arbeit ist eine elektronische Fassung sowie eine unterschriebene Erklärung beizufügen, dass die vorgelegte Arbeit selbständig verfasst wurde und alle Stellen der Arbeit, die anderen Werken entnommen sind, kenntlich gemacht wurden.

3. Die Verwendung von Plagiaten wird als Täuschungs-versuch gewertet. Wird ein Täuschungsversuch nachgewiesen, so gilt die Studien- oder Prüfungs-leistung als nicht bestanden.

TU Informatik-Fakultätsratsbeschluß vom 17.1.2007

"wissenschaftliche Arbeit" = "(neben Abschlussarbeiten auch Seminararbeiten, Studienarbeiten, Hausarbeiten, Projektberichte) "

IG und die Informatik

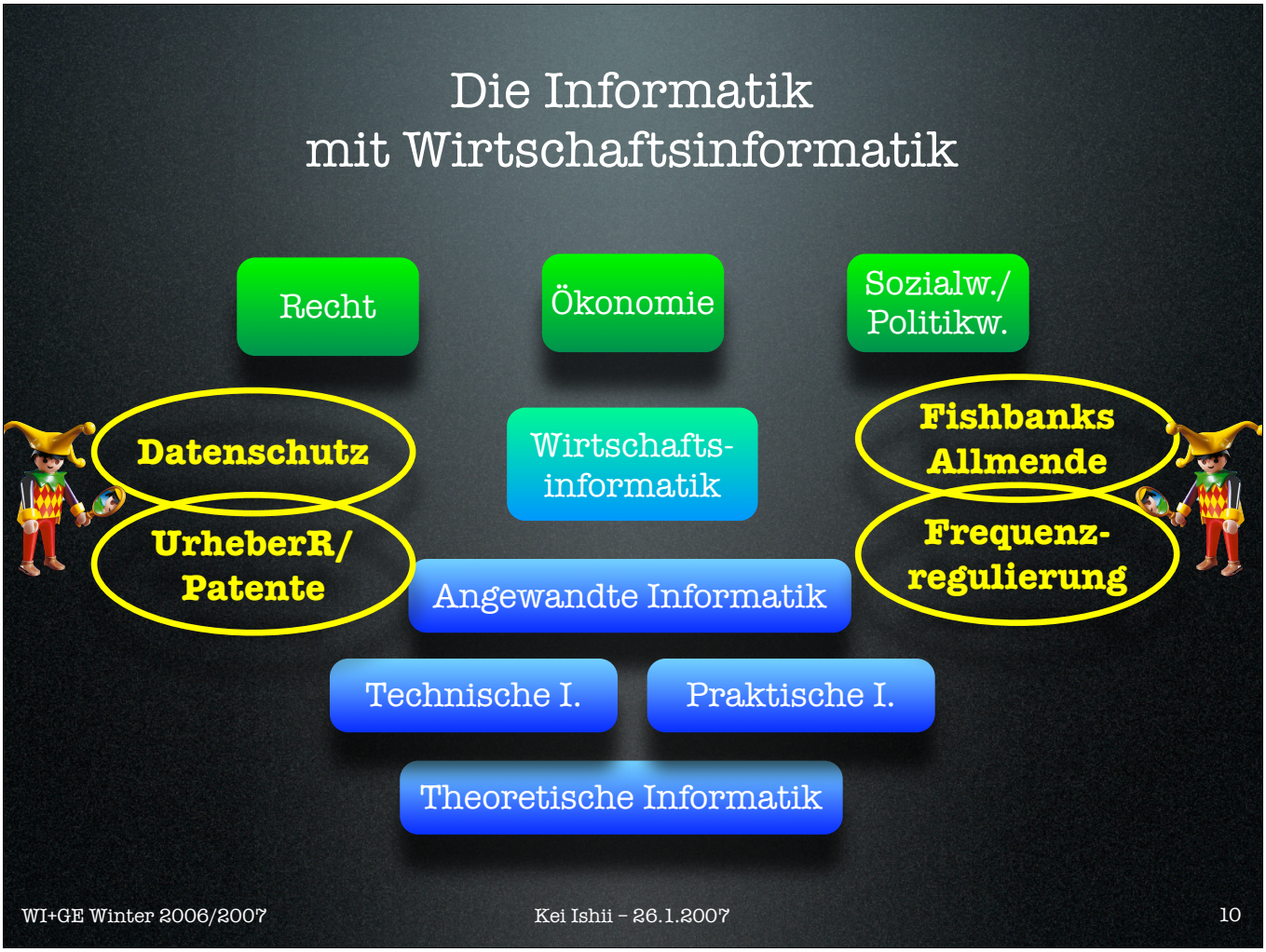


http://freepatriot.com/imagewarehouse/YouAreHereGalaxy.jpg

Hofnarr: <http://www.grossbahnen.de/informationen/playmobil/bildergalerie/sortiment_orig_2004/4610.jpg>

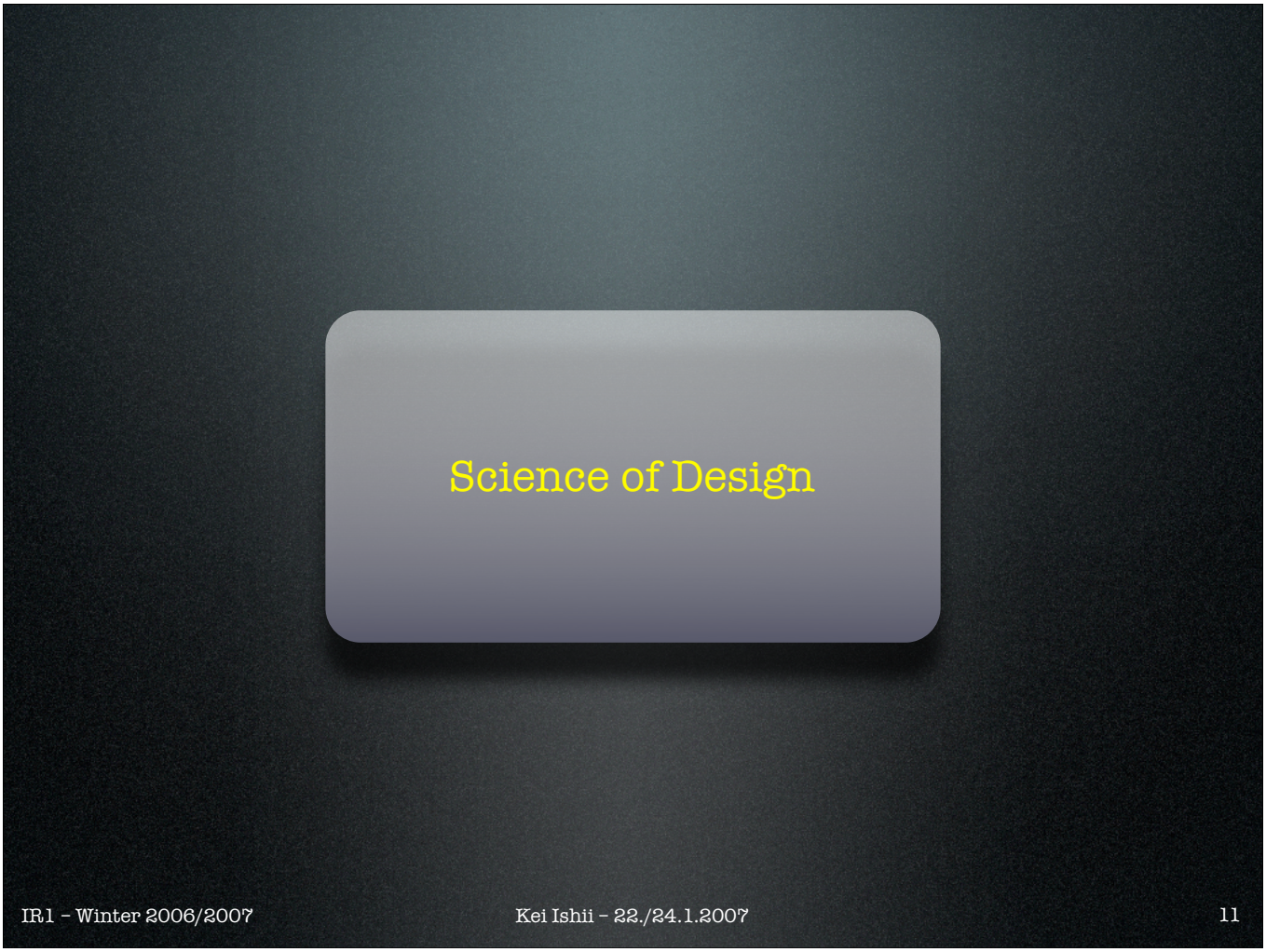
«Informatik und Gesellschaft» als «Hofnarr» der Informatik?

Die I&G hat innerhalb der Informatik eine möglicherweise kleine, aber wichtige Funktion.



Hofnarr: <http://www.grossbahnen.de/informationen/playmobil/bildergalerie/sortiment_orig_2004/4610.jpg>

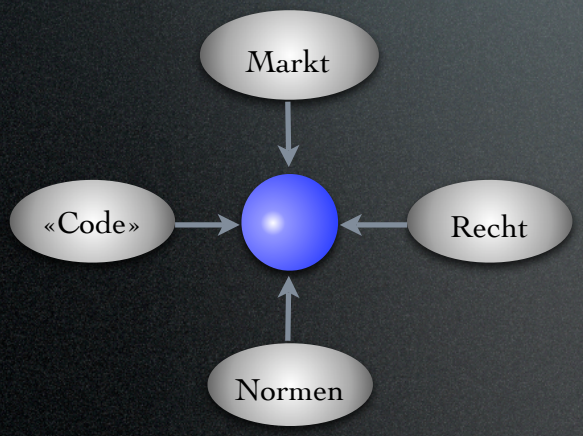
Einordnung einiger WI+GE-Vorlesungen



«Science of Design»:
Wir basteln uns eine mögliche zukünftie "Teildisziplin" der Informatik

Wichtige Modelle

Regulierungsarten
«Kleeblattmodell»



Vom Recht zum «Code»
Lex Informatica

	Rechtssystem	DVD	Lex Informatica
Geltungsbereich	"Politische" Grenzen	"Code Regionen"	Netzwerk
Quelle	Legislative	DVD-Konsortium	Technologen
Primäre Durchsetzung	Exekutive Judikative	DVD + Gerät	Automatisiert, selbstausführend
Inhalt	Rechtsvorschriften Gerichtsurteile	Abspielkontrolle DVD-Standard	Techn. Funktionen Gewöhnl. Gebrauch
Rahmen	Rechtssystem	DVD-"System"	Architektur, Standards
Anpassung	Vertrag	(DVD-Laufwerk von Computern)	Konfiguration

Eine kleine Rekapitulation wichtiger Modelle aus der Veranstaltung.

Das hehre Ziel: Science of Design

(Titelblatt von: Simon, Herbert 1996 (1969): The Sciences of the Artificial. Cambridge (Mass.), London: The MIT Press.)

Science of Design
Wozu?

Brueghel, Tower of Babel: <http://witewave.com.ne.kr/gallery/Bruegel_Tower_of_Babel1.jpg>
Tower of Inkompatibabel: <http://hello.eboy.com/eboy/wp-content/uploads/2007/01/PT_babeltower_01t.png>

Die Informatiker «bauen» mitunter große Gebilde.

Informatikerbedarf 2010

COMPUTERWORLD
Careers

Print Article Close Window

What's Hot, What's Not: IT Skills You'll Need in 2010

Stacy Collett

July 17, 2006 (Computerworld) The most sought-after corporate IT workers in 2010 may

http://www.computerworld.com/action/article.do?command=viewArticleTOC&specialReportId=9000100&articleId=112360

IT in 2010
"Technology Infrastructure and Services"

Hot

- System analysis
- Systems design
- Network design
- Systems auditing

Cold

- Programming
- Routine coding
- Systems testing
- Support and help desk
- Operations -- server hosting, telecom-munications, operating systems

Zitat aus dem Computerworld-Artikel:

«According to respondents to the Gartner survey, the skills that will have the steepest decline in 2010 will be in technology infrastructure and service jobs -- such as programming and operations work. Those roles will go overseas or more likely be automated.

"The more that [a task] can get codified or changed into explicit instructions or documentation, the more likely it can get transferred. The more likely it can be transferred, the more likely someone will come along and will develop tools to reduce even further the number of people required to do the job," Morello says.

Kaiser says that in the SIM study, keeping systems design and analysis skills in-house in the next five years were considered critical, yet those tasks are frequently outsourced. The reason respondents gave for outsourcing these skills included meeting project needs and enabling flexible staffing.»

Komplexe SIS: Beispiel NHS IT-Programme

UK Staatliche IT-Projekte und Firmenjahresumsätze

- ID-Cards and Biometric Passports: £3-19 Mrd.
- eBorders: £400 Mio.
- Police: Impact: £140 Mio.
- Criminal Justice IT £2 Mrd.
- National programme for NHS IT: £6-20 Mrd.
- Defence Information Infrastructure: £4 Mrd.

- RedHat: £150,6 Mio.
- Schering: £7,6 Mrd.
- Microsoft: £ 21,6 Mrd.

Quelle: Sarah Arnot, Computing 09 Nov 2005

RedHat-Umsatz: Finanzjahr 2006 278 Mio $ (£150,6 Mio.)
Schering: Pro-Forma Jahresumsatz: 11,2 Mrd. € (£7,6 Mrd.)
Microsoft: Umsatz 2005 39,79 Mrd. $ (£ 21,6 Mrd.)

Beispiel aus UK: NHS

- Staatlicher britischer Gesundheitsdienst
- Projekt: "National Programme for IT in the NHS"
  "Better information for health, where and when it's needed"
- Zentrale Datenbank mit Patientendaten
  - 50m Patienten/1.4m Angestellte/30k Ärzte
  - "utopian cradle-to-grave vision of a centralized, monolithic national health care system"
- 'wohl größtes Softwareprojekt der Welt' (NZZ)

NHS = National Health Service

Projektzitat (2. Punkt) von der NHS-Website, http://www.connectingforhealth.nhs.uk/index_html (2006-06-13)

"utopian ...": Baselinemag 2006: "U.K. Dept of Health: Prescription for Disaster". 13 Nov 2006 (<http://www.baselinemag.com/print_article2/0,1217,a=193664,00.asp>)

Voraussetzungen

- Budget £ 6,2 Mrd. (ca. 9 Mrd. €) eher £20 Mrd.

  ➡ bestes Wissen "that money can buy"

- Zeitraum: Auf 10 Jahre angelegt (2002-2012)

  ➡ kein Mangel an Zeit 2005: min 2½ Jahre hinter Plan

- Forschung in Medizinischer DV, Datenbank- und Datenschutz-Technologien

  ➡ kein Mangel an Fachexpertise Datenschutzfragen ignoriert

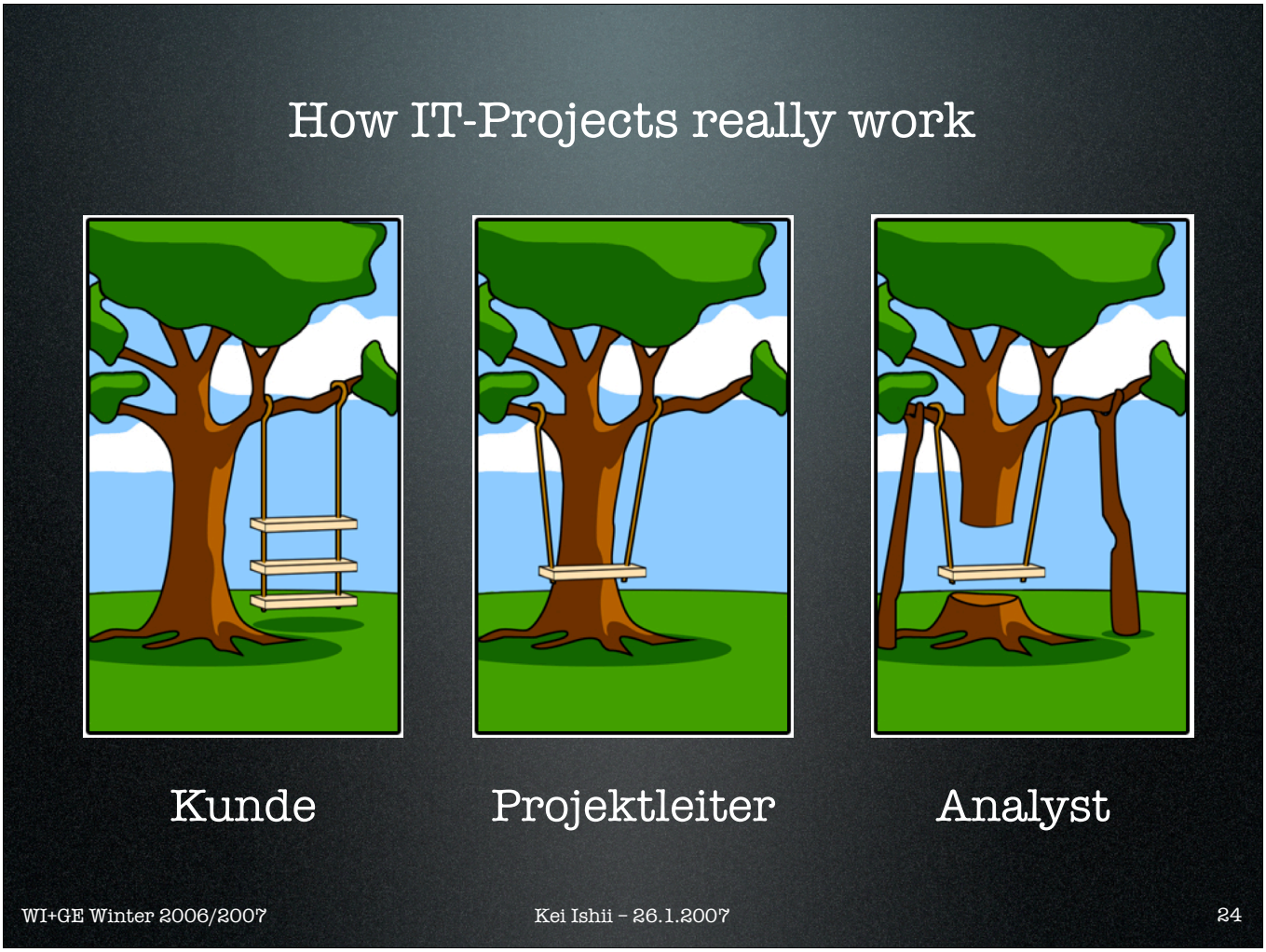
- "Der britische Gesundheitsdienst schlingert. In: Neue Zürcher Zeitung v. 8. Juni 2006 S. 21.
- Baselinemag 2006: "U.K. Dept of Health: Prescription for Disaster". 13 Nov 2006 (<http://www.baselinemag.com/print_article2/0,1217,a=193664,00.asp>)

Expertise:
«Die Verzögerungen sind zum Teil die Folge eines Streites darüber, welche Patientendaten überhaupt in das landesweite System aufgenommen werden sollen. Diese für die Privatsphäre der Patienten äußert zentrale Frage wurde im Vorfeld nicht ausdiskutiert.» (NZZ 2006)

Außerdem: Die "Clinicians" (Ärzte, Schwestern etc.), die mit dem System arbeiten sollen, wurden erst spät in den Entwicklungsprozeß einbezogen. (Verträge abgeschlossen: 2/2003-2/2004; Beteiligte wurden erst im August 2005 gesucht, damit sie mit den Entwicklern sprechen. (Baselinemag 2006)

"How IT Projects really work"

Source: <http://www.projectcartoon.com/cartoon/2>

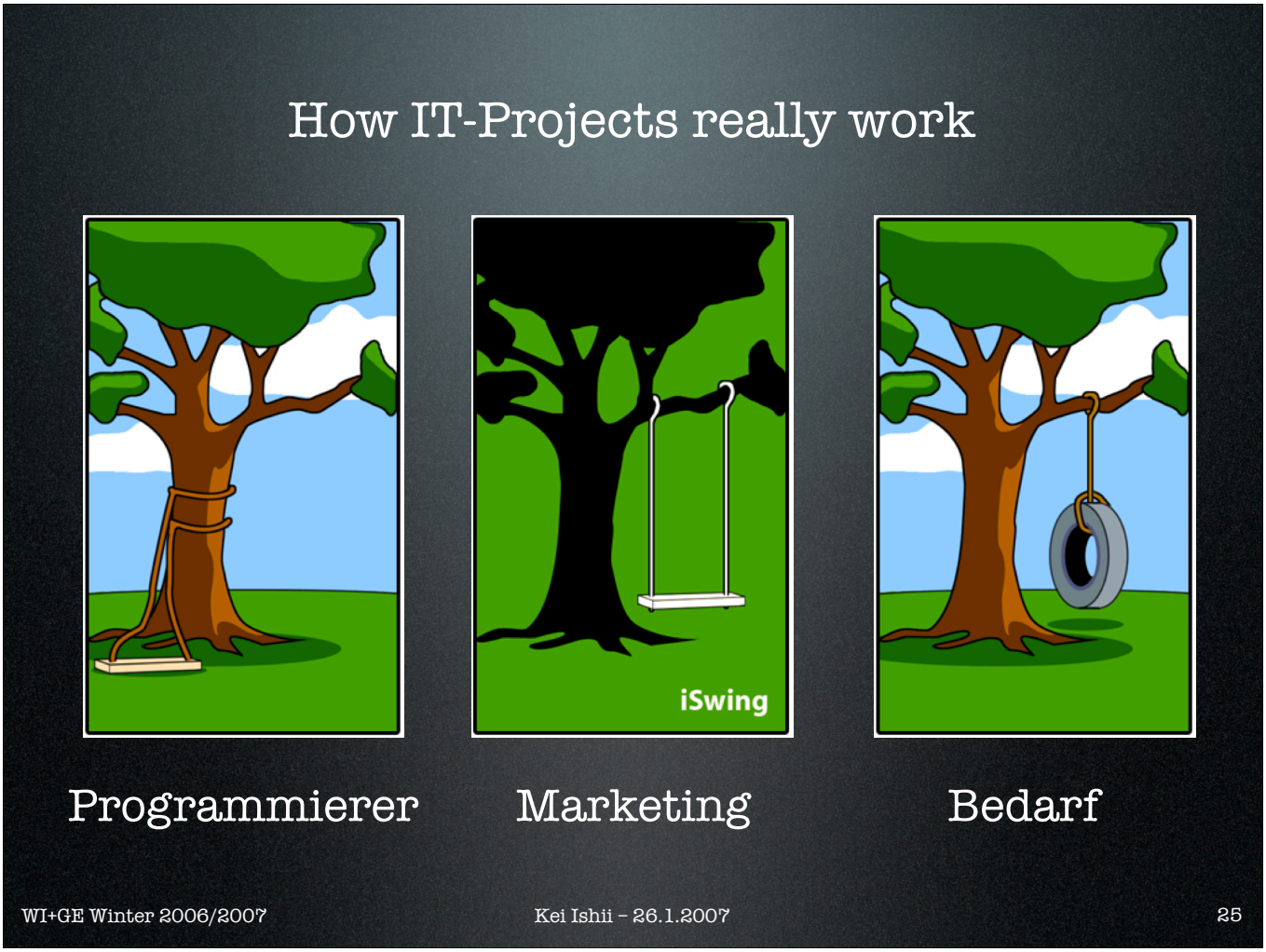


Source: <http://www.projectcartoon.com/cartoon/2>

– Wie der Kunde es erklärte («How the customer explained it»)

– Wie der Projektleiter es verstand («How the project leader understood it»)

– Wie der Analyst es designed hat («How the analyst designed it»)



Source: <http://www.projectcartoon.com/cartoon/2>

– Wie der Programmierer es schrieb («How the programmer wrote it»)

– Wie Marketing es anpries («What marketing advertised»)

– Was der Kunde wirklich brauchte («What the customer really needed»)

Science of Design – Software Intensive Systems (NSF)

Science of Design – SIS
Freeman, Hart

Viewpoint | Peter Freeman and David Hart

A Science of Design for Software-Intensive Systems

Freeman: "assistant director for Computer and Information Science and Engineering"

Hart: "information specialist in the NSF's Office of Legislative and Public Affairs"

Freeman/Hart 2004: A Science of Design for Software-Intensive Systems. In: Communications of the ACM August 2004, Vol. 47 No. 8, pp. 19-21.

–NSF = National Science Foundation, amerikanische staatliche Wissenschaftsstiftung

–Freeman ist "assistant director for Computer and Information Science and Engineering"
–Hart ist "information specialist" in der "Office of Legislative and Public Affairs"
(man beachte: Technologie-Division und Büro für Recht und Öffentlichkeit)

Science of Design – SIS Grundproblem

- Nach 30 Jahren Forschung:
  keine wissenschaftliche Basis für den Bau komplexer software-intensiver Systeme

  ➡ Problem, die Systeme zu verstehen und zu kontrollieren

  ➡ Sicherheits-, Datenschutzprobleme oder unerwartete Eigenschaften als häufige (negative) Begleiterscheinungen

"[I]n spite of more than three decades of effort, we still lack what could be called a solid scientific basis for building any kind of large-scale system that includes software elements. Our software-intensive systems are so complex we often find ourselves struggling to understand and control them. New systems are created and adopted with high hopes, crossed fingers, and even controversy, as in the case of e-voting. Security breaches, privacy violations, and emergent (unexpected) properties are some of the regular, often unwelcome, results." (Freeman/Hart 2004, S. 19)

Science of Design-SIS
konkrete Beispiele

- **Elektronische Wahlsysteme**
  Schutz von Bürger- und Individualrechten, Minimierung politscher Verzerrungen und Sicherheitsbedrohungen
- **Stadtplanung**
  Soziale/kulturelle Diversität, öffentliche Entscheidungs-findungen, Zugänglichkeit für Minoritäten
- **Pervasive Technologien**
  Verhinderung negativer Effekte auf Gesundheit, informationelle Selbstbestimmung und Bürgerrechte

"How might e-voting systems be designed, to protect civil rights and personal privacy while minimizing political bias and security threats?

"How might urban planning systems be designed to account for social and cultural diversity, public decision making, and accessibility?

"How might systems using pervasive “disposable” technologies (such as wireless sensor networks) be designed to avoid adverse effects on people’s physical health, personal privacy, and civil liberties?" (Freeman/Hart 2004, S. 20)

Science of Design-SIS
Conclusio

Herausforderung Science of Design

- anders über Software nachdenken
- grundlegende Änderungen in der Arbeitsweise von Informatikern

Ohne ein Science of Design

- sei man unfähig, mit den Chancen umzugehen, die vor uns liegen

«Creating a science of design requires computer scientists and engineers in academic institutions and in industry to think about software differently and make fundamental changes in the way they work. But without a science of design, it is likely that we will soon be unable to deal with the opportunities before us.» (Freeman/Hart 2004, S. 21)

Science of Design-SIS Forderung

Die Informatik braucht einen

intellektuell strengen, analytischen, lehrbaren Designprozeß,

um die Entwicklung von Systemen zu ermöglichen,

mit denen wir alle leben können.

"Computer science and engineering needs an intellectually rigorous, analytical, teachable design process to ensure development of systems we all can live with." (Freeman/Hart 2004, S. 19)

Science of Design-SIS
Ein wichtiger Punkt

Vielleicht die wichtigste Frage von allen ist:
Wie stellen wir sicher, daß wir von unseren Mißerfolgen lernen?

"Perhaps the most important question of all is: How do we ensure that we learn from our failures?"

"Engineering again offers a useful analogy. In the years since the Industrial Revolution, though engineering has matured and justifiably boasts many successes, steam engines have exploded, bridges collapsed, buildings burned, and flying machines crashed—not to mention other, less dramatic failures. Many engineered artifacts (such as bridges and canals) are public works, whose failures are subject to public examination. Others (such as steam engines) are proprietary, with the knowledge kept secret within the limited scope of companies. Because so much software is proprietary, even in publicly deployed systems, a special challenge for the science of design is to find ways to learn from our failures, spectacular or otherwise, as we build software-intensive systems." (Freeman/Hart 2004, S. 20-21)

Science of Design Ansätze

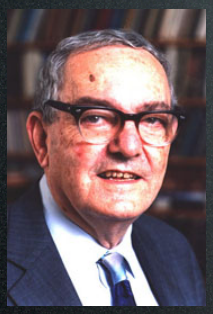
Einige Ansätze

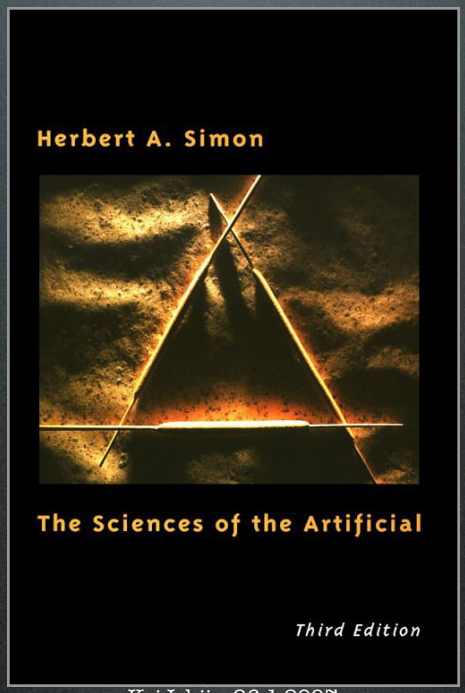
- Von den Alten lernen
- End-to-End als politisches Prinzip
- Regulierungsmuster

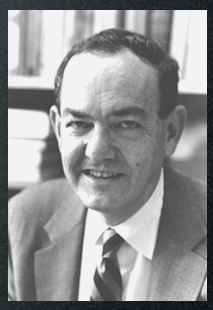
Von den Alten lernen

Herbert Simon

Science of Design
Herbert Simon



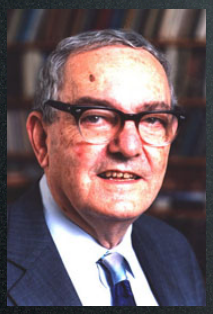




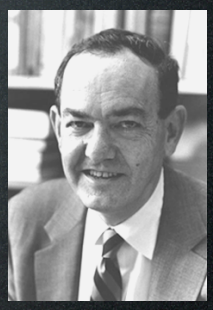
Simon, Herbert 1996 (1969): The Sciences of the Artificial. 3. Auflage. Cambridge (Mass.), London: The MIT Press.

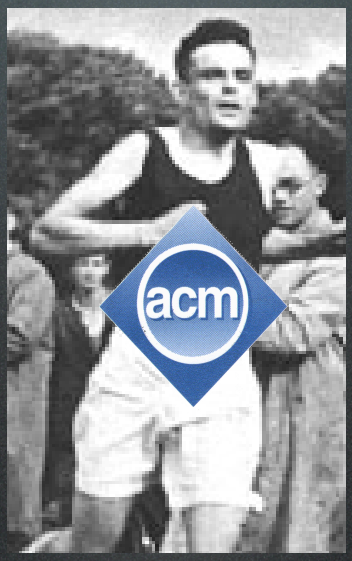
Die Idee des "Science of Design" geht auf einen berühmten Informatiker zurück: Herbert Simon.

Herbert A. Simon



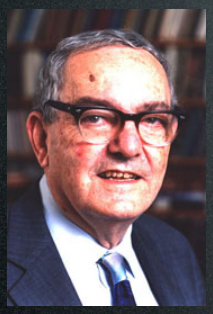
ACM Alan M. Turing
Preis 1975



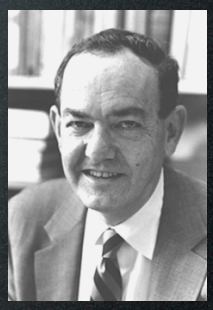


Als (Mit-)Begründer der Künstlichen Intelligenz erhielt er den "Nobelpreis der Informatiker", den Alan Turing Preis.

Herbert A. Simon



Nobelpreis für Ökonomie 1978



Als Wirtschaftswissenschaftler erhielt er auch den Nobelpreis für Ökonomie.

Herbert A. Simon

- Problemlösungsprozesse beim Menschen
- Entscheidungsprozesse in Wirtschaftsorganisationen
  - ➡ begrenzte Rationalität (→Nobelpreis)
- Computersimulation und Künstliche Intelligenz ("general problem solver") (→Turing Award)
- "Science of Design"

Seine Interessen galten diesen Themen:

–Human Problem Solving
–Organizational Decision Making (hierfür Nobelpreis)
–>Prägte das Konzept der "bounded rationality"

– Mitbegründer der Artificial Intelligence (hierfür Turingpreis)

–Und auf ihn geht der Vorschlag für ein "Science of Design" zurück, bereits in den 1960er Jahren formuliert

Herbert A. Simon: The Sciences of the Artificial

1. Die natürlichen und die künstlichen Welten verstehen
2. Rationales Verhalten in der Wirtschaft. — **Ökonomie**
3. Die Psychologie des Denkens. — **Psychologie**
4. Erinnern und Lernen.
5. Die Wissenschaft vom Entwerfen. — **Science of Design**
6. Gesellschaftsplanung. — **Sozialwissenschaften**
7. Die Architektur des Komplexen. — **Komplexitätstheorie**

Das Buch erschien 1969 in erster Auflage, und wurde 1981 (2.Aufl.) und 1996 (3. Aufl.) jeweils erweitert.

Sein Bestreben: Eine neue Wissenschaft neben Natur- und Geisteswissenschaften zu begründen, die sich mit der Gestaltung von "Artefakten" beschäftigt.

Die englischen Kapitelüberschriften der 3. Auflage:

1. Understanding the Natural and Artificial Worlds
2. Economic Rationality: Adaptive Artifice
3. The Psychology of Thinking: Embedding Artifice in Nature
4. Remembering and Learning: Memory as Environment for Thought
5. The Science of Design: Creating the Artificial
6. Social Planning: Designing the Evolving Artifact
7. Alternative Views of Complexity
8. The Architecture of Complexity: Hierarchic Systems

Herbert A. Simon: Sciences of the Artificial

Informatik

Ökonomie

Psychologie

Sozialwissenschaften

Komplexitätstheorie

Frederick P. Brooks

Frederick P. Brooks

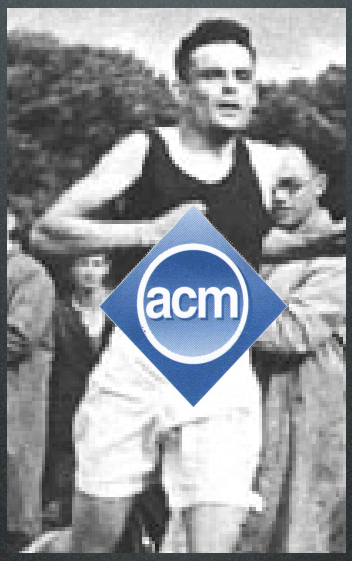
- IBM OS/360
- "Mythical Man Month" (1975)
  - ➡ "Adding manpower to a late software project makes it later" (Brookssches Gesetz)
- "No silver bullet" (1987)
  - ➡ 'unbeabsichtigte' vs. 'prinzipielle' Komplexität von Software

Brooks, Frederick P. (1975/1995): "Mythical Man Month. Essays on Software Engineering. Addison-Wesley Professional. 20th Anniversary Edition.
Brooks, Frederick P. (1987): "No Silver Bullet: Essence and Accidents of Software Engineering," Computer, Vol. 20, No. 4 (April 1987) pp. 10-19.

–Verantwortlicher Leiter für die Entwicklung des IBM OS/360
–Verarbeitet Erfahrungen in "Mythical Man Month" (1975)
  –Brooks' Law: "Adding manpower to a late software project make it later" (Komplexität Programmierer wächst n(n-1)/2, Projekt aber höchstens n)
–Weiteres bekanntes Paper: "No silver bullet" (1986)
  –unterscheidet zwischen "accidental" (unbeabsichtigter) und "essential" (prinzipieller) Komplexität, wobei die Informatik erstere grundlegend gelöst habe, und die Probleme jetzt prinzipieller Natur seien. (WE "No silver bullet"

Frederick P. Brooks

ACM Alan M. Turing
Preis 1999



Frederick P. Brooks

- "The Design of Design"
  - "Design Model"
  - "Collaboration"
  - "Great Designer"

Die folgenden Folien aus: Brooks, Frederick P. 2000: "The Design of Design". Präsentationsfolien anläßlich der Übergabe des Turing Awards. <http://online.cs.nps.navy.mil/DistanceEducation/online.siggraph.org/2001/SpecialSessions/2000TuringLecture-DesignOfDesign/brooks.pdf>

Vortrag "The Design of Design":
–Turing Award Address 2000
–Kein Fließtext vorhanden, nur Folien (sowie Video)

Brooks, The Design of Design
1. Design Model

Brooks – Design Model

I. How Engineers Think of Design

- Goal
- Desiderata
- (Non-linear) utility function
- Constraints, especially budget (not necessarily $ cost)
- Design tree of decisions

```
UNTIL (design is "good enough") or (time has run out)
  DO another design (to improve utility function)
    UNTIL design is complete
      WHILE design remains feasible,
        make another design decision
      END WHILE
      Backtrack up design tree
      Explore a path not searched before
    END UNTIL
  END DO
  Take best design
END UNTIL
```

7 © FPB – 7/4//00

Brooks 2000, Folie 7.

Das "rationale Modell"

Brooks – Design Model

What's Wrong with This Model? — 1

- We don't really know the goal at first –

- **Here is where experts go wrong:**
  - Miss fresh vision – e.g., minicomputer, microcomputer
  - Vision not high enough – e.g., OS/360 JCL

9 © FPB — 7/4//00

Brooks 2000, Folie 9.

Probleme 1:

Ziel ist am Anfang unbekannt, selbst für "Experten"

Brooks – Design Model

What's Wrong with This Model? — 2

- The desiderata and their weightings keep changing.
- We usually don't know the design tree.
- The constraints keep changing.
  - Often by the ever-changing world.
  - Sometimes by total systems thinking, outside the box!

13 © FPB —7/4/00

Brooks 2000, Folie 13.

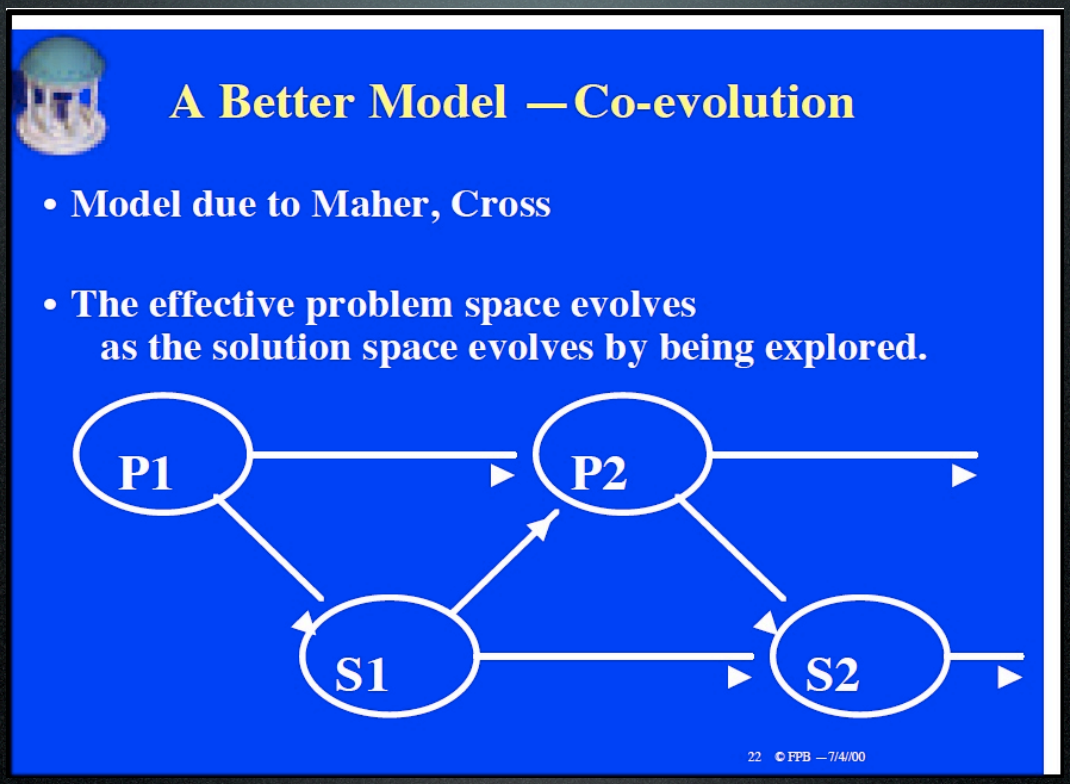
Probleme 2

–Desiderata (Wünsche bzw. was gebraucht wird) und ihre Gewichtung ändert sich

–Design tree = ?

Die Rahmenbedingungen ändern sich

Brooks – Design Model



Brooks 2000, Folie 22.

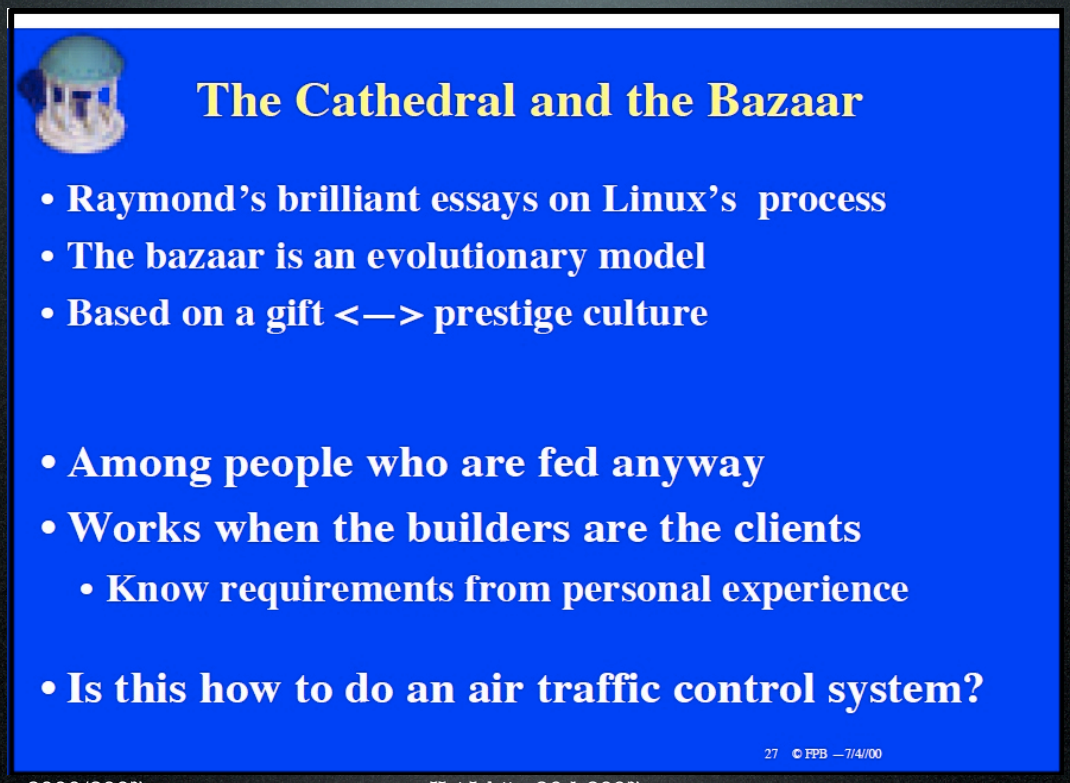
Verweise auf Maher, Cross:

Nigel Cross: http://design.open.ac.uk/cross/publications.htm
–Buch 2006: "Designerly Ways of Knowing" (Springer)

Mary Lou Maher: http://www.arch.usyd.edu.au/~mary/
–Buchprojekt (in progress?): "Design as Coevolution" http://www.arch.usyd.edu.au/~mary/coevolve/

Brooks, The Design of Design
2. Collaboration

Brooks – Collaboration



Brooks 2000, Folie 27

OSS-Modell kann nützlich sein, aber nicht immer (häufig nicht?)

Brooks – Collaboration

Design Collaboration

- Real design is always more complex than we imagine.
- Design change control
- The cleaner the interfaces, the fewer errors.
- Collaboration is no substitute for "the dreariness of labor and the loneliness of thought."

35 © FPB —7/4/00

Brooks 2000, Folie 35

Brooks – Collaboration

When Does Collaboration Help?

- Determining needs from users
- Conceptual exploration — radical alternatives
- *Not* conceptual design nor detailed design
- Design reviews
  - Especially with different expertise
  - Need and exploit richer graphical representations

39 © FPB —7/4/00

Brooks 2000, Folie 39

Brooks, The Design of Design
3. Great Designer

Frederick P. Brooks

Great Designs Come From Great Designers

- How does one do great designs *within* a product process?
- How to make a product process than encourages, rather than inhibits, great designs?
- Where elitism is proper
- Entrust design to a chief designer

43 © FPB – 7/4//00

Brooks 2000, Folie 43

Frederick P. Brooks

Where Do Great Designers Come From?

- We have to grow them *deliberately.*
- We have to manage them *imaginatively.*
- We have to protect them *fiercely.*
  - From managers
  - From managing

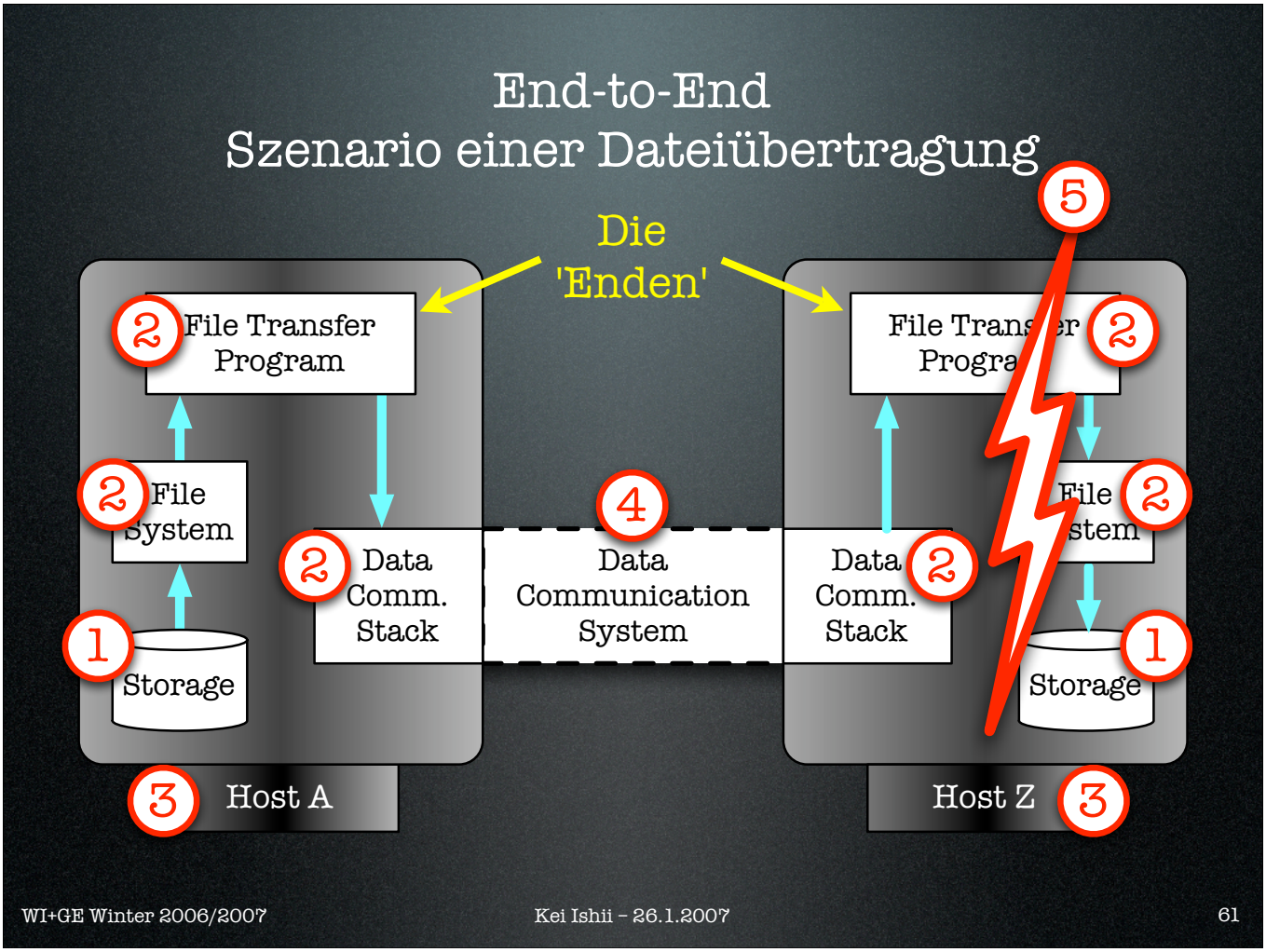
46 © FPB —7/4/00

Brooks 2000, Folie 46

Frederick P. Brooks
The Design of Design

- "Design Model"
  - Co-Evolution
- "Collaboration"
  - Nur beschränkt nützlich
- "Great Designer"
  - Schlüsselfigur für gutes Design

End-to-End als politisches Prinzip

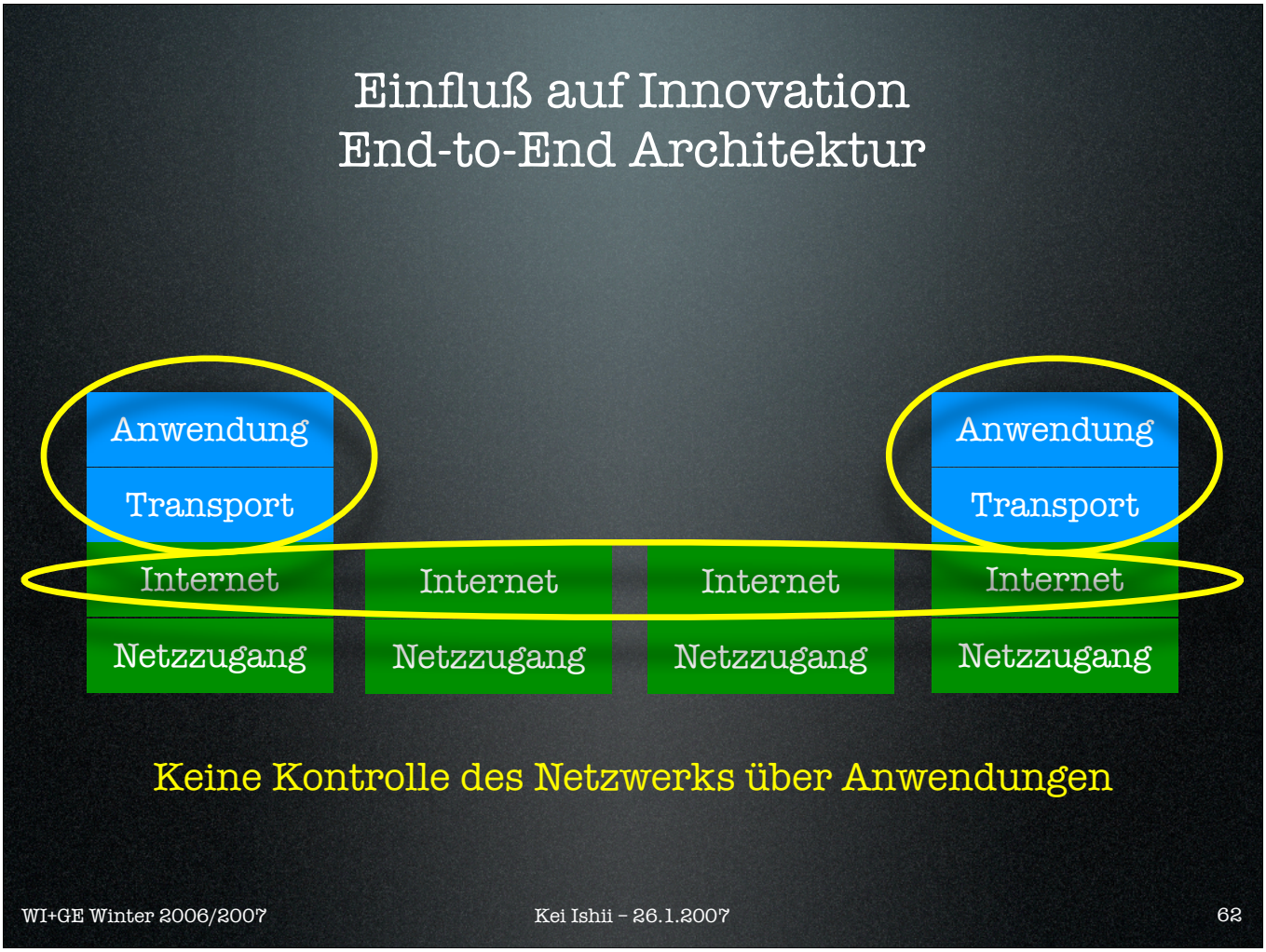


(Wiederholung aus der 4. Vorlesung; Referenzen siehe dort)

Nach Saltzer/Reed/Clark 1984, S. 2.

1 = Fehler in Massenspeicher
2 = Softwarefehler
3 = Fehler in Hardware des Hosts (z.B. CPU)
4 = Fehler in der Übertragung (z.B. verlorenes Paket)
5 = Ausfall eines Hosts (hier nur für Zielhost dargestellt)

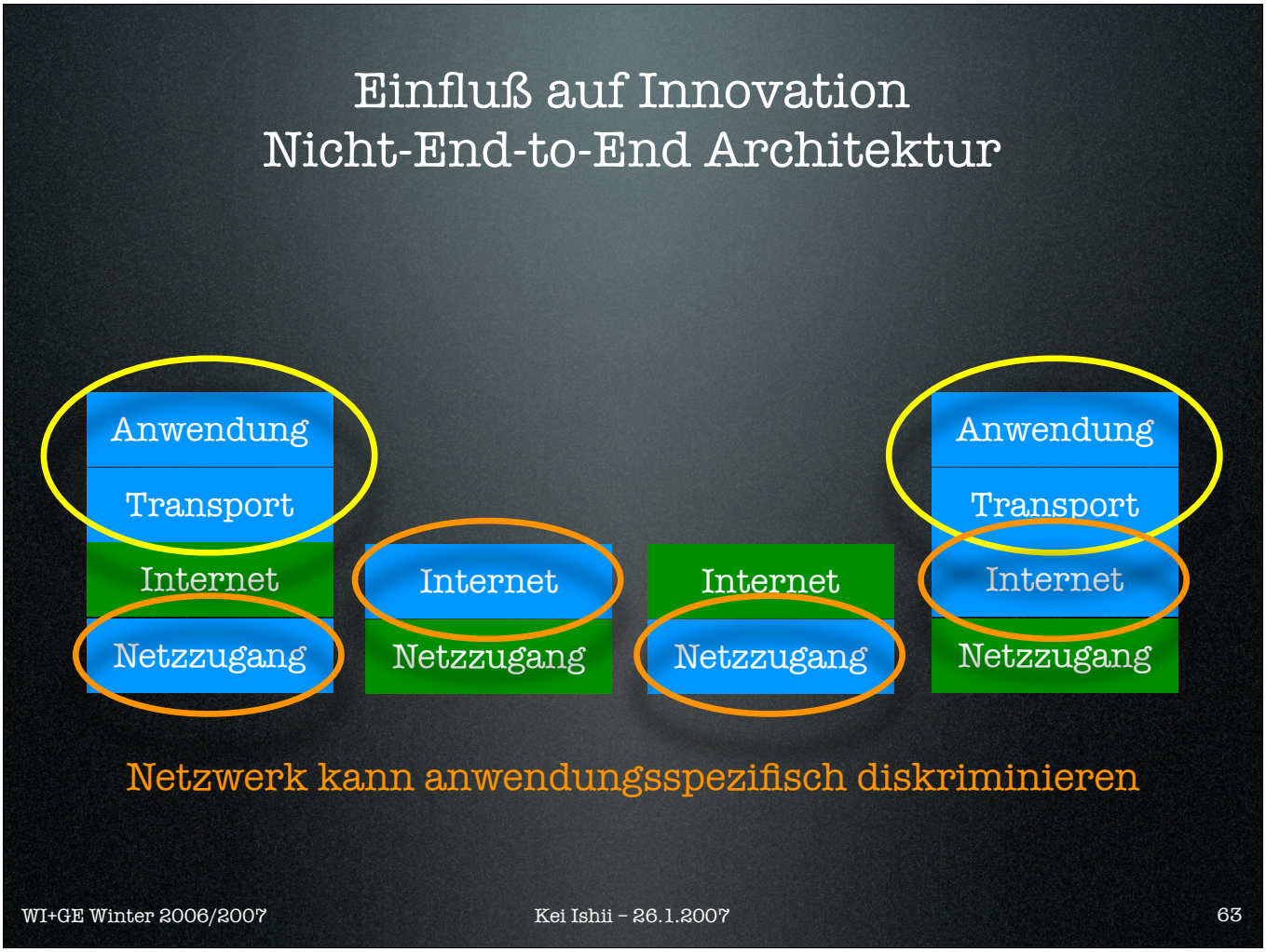
Frage: Wo sollten Fehlerentdeckungs- bzw. -behebungsroutinen implementiert werden?
- Frage der Nützlichkeit von Fehlerroutinen z.B. im Kommunikationssystem, Festplatte vs. im File Transfer Program
- Endpunkte hier sind die jeweiligen Instanzen des File Transfer Program. D.h. selbst wenn in den anderen Komponenten Fehlerroutinen sind, muß hier nochmals gecheckt werden. Bei relativ zuverlässigen Komponenten also reichte es, nur hier - an den Enden - Fehlerroutinen zu implementieren.



(Wiederholung aus der 4. Vorlesung; Referenzen siehe dort)

Van Schewick 2006.

- Innovation wird auf Anwendungs-/Transportebene realisiert.
- Das Netzwerk ist anwendungsblind, daher kann es keine spezifischen Anwendungen benachteiligen; das Netzwerk hat keine Kontrolle über die Anwendungen



(Wiederholung aus der 4. Vorlesung; Referenzen siehe dort)

Van Schewick 2006.

- Hier kann das Netzwerk zwischen Anwendungen unterscheiden, und damit anwendungsspezifisch diskriminieren. Das Netzwerk kontrolliert die Anwendungen

Die Suche nach Regulierungsmustern im «Code»

Lex Informatica

Regulieren durch Technologie

Informatiker regulieren

Welche Konsequenzen für...

...mich (als Informatiker)?

...die Informatik (als Disziplin)?

Forschungsidee
«Code» Governance

Spezifische Strukturen und Prozesse

in Software

als Regulierungssystem
interpretieren

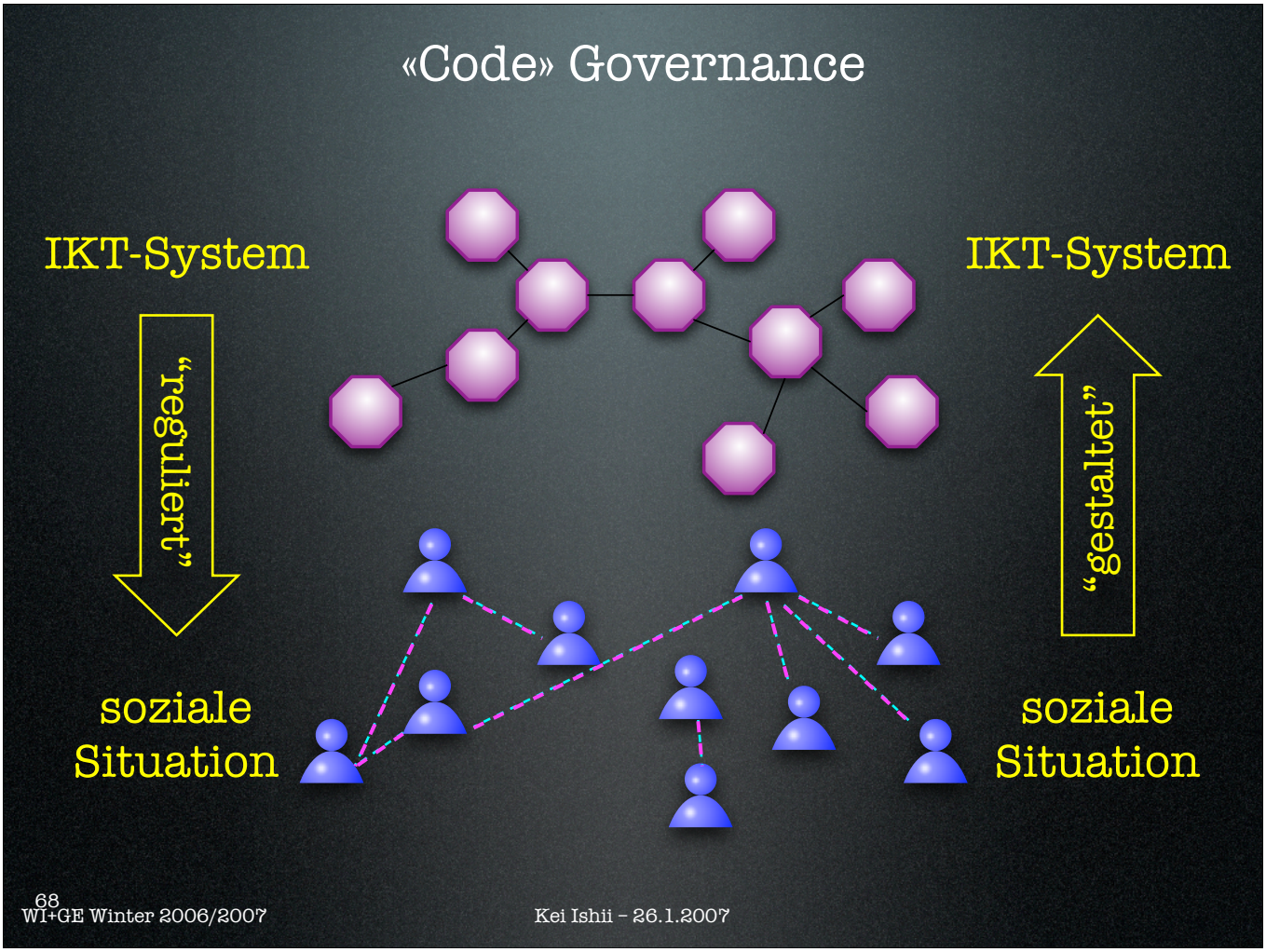
Ishii, Kei (2005): «Code Governance. «Code» as Regulation in a Self-governed Internet Application from a Computer Science Perspective (english). Dissertation thesis.» <http://ishii.de/kei/codegovernance/Ishii2005-CodeGovernance.pdf>

Forschungsidee:
–spezifische Strukturen und Prozesse

–in Software

–als Regulierungssystem interpretieren

Meine Bezeichnung dafür: «Code» Governance



Was ist Code Governance?

–Technisches System
–Individuen, die es benutzen, oder gestalten/verändern
–Diese Individuen stehen in sozialen Beziehungen zueinander; zusammen bilden sie eine "soziale Situation"
–Diese Beziehungen können natürlich auch IKT-mediiert sein

Die These besagt nun, daß
–1. System die soziale Situation (mit-)reguliert, UND
–2. die soziale Situation das System (mit-)gestaltet

Mein Interesse gilt insbesondere der technischen Seite.

Internet Relay Chat

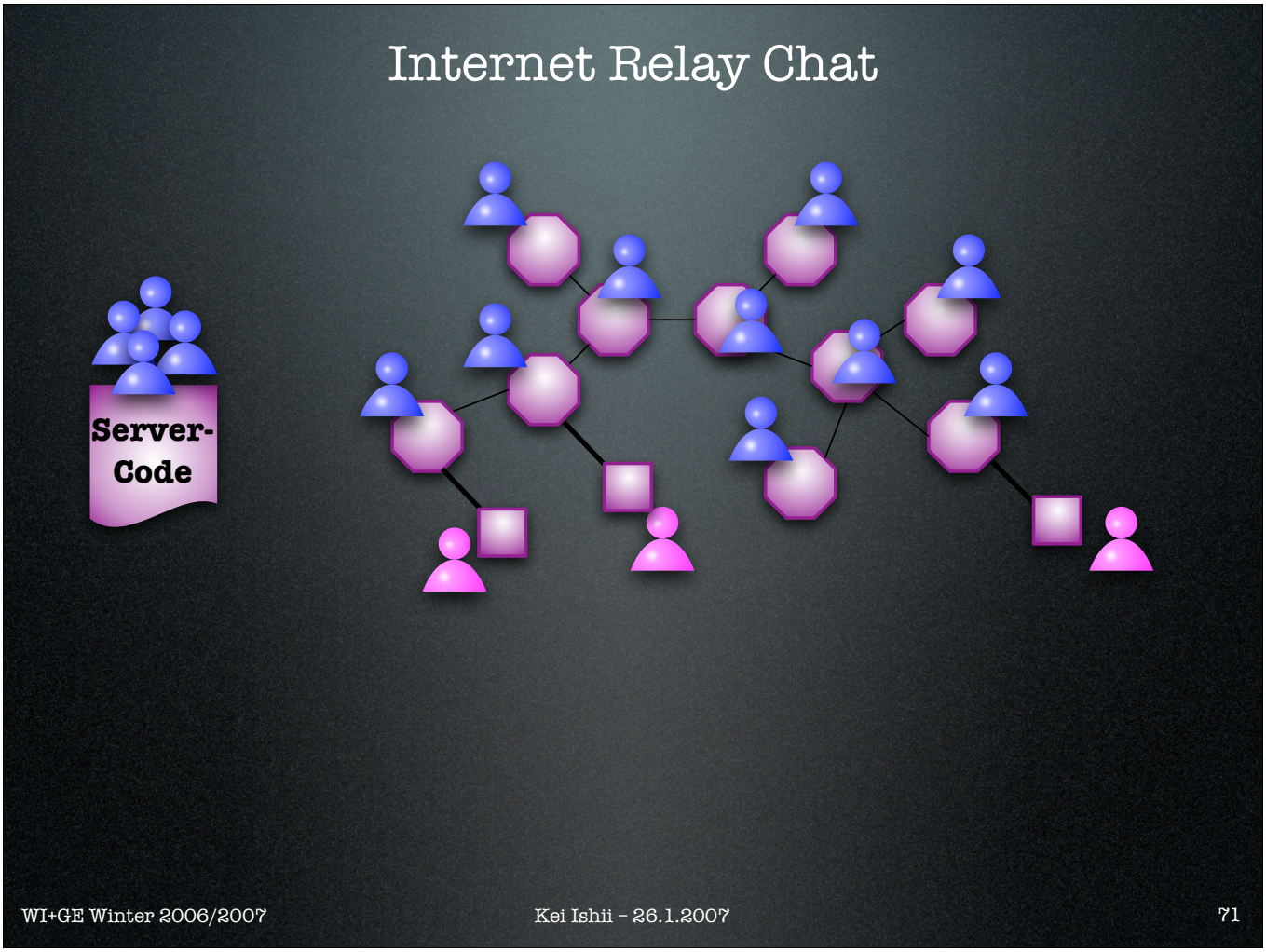
- Existiert seit 1989
- Echtzeit-Kommunikation (“chat”)
- Chat in Gruppen (Channels)

Als Untersuchungsobjekt habe ich das Internet Relay Chat (kurz IRC) gewählt, eine seit 1989 existierende Internet-Anwendung, in der Benutzer in Echtzeit miteinander kommunizieren oder "chatten" können.
Die Chats finden in Gruppen, sogenannten Channels, statt

Internet Relay Chat

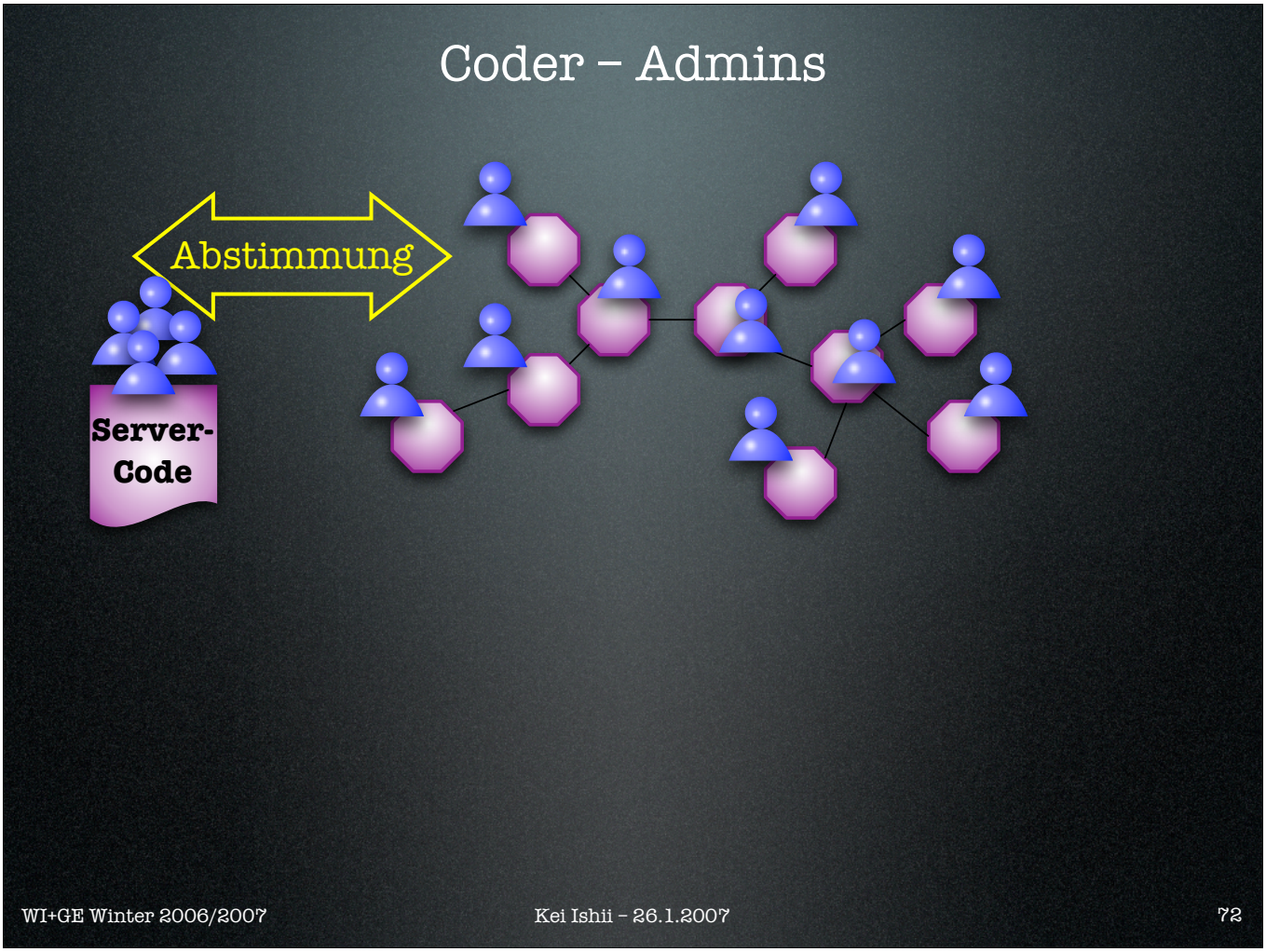
- Eine der größten Anwendungen im Internet (>100.000 Benutzer)
- IRC-Software frei verfügbar
- Selbst-organisiert, selbst-reguliert
- Hauptregulierungsmodalität: «Code»

Ich habe das IRC aus mehreren Gründen gewählt:
– Die Anwendung besteht schon sehr lange, und gehört zu den größten im Internet überhaupt. Zu jeder Tages- und Nachtzeit sind hunderttausende von Benutzern online.
– Für meine «Code»Untersuchung benötigte ich freien Zugang zum Quellcode der Anwendung: so konnte ich etwa 200 verschiedene IRC-Software-Versionen sammeln und analysieren.
– Und schließlich ist die Anwendung selbst-organisiert und selbst-reguliert. Sowohl die Aufrechterhaltung der IRC-Netze wie auch die Weiterentwicklung der Software liegt in den Händen von individuellen Freiwilligen. Das bedeutet, daß die Regulierungsmodalitäten "Recht" und "Markt" vernachlässigbar sind:
In der Hauptsache wird mithilfe von «Code»reguliert.



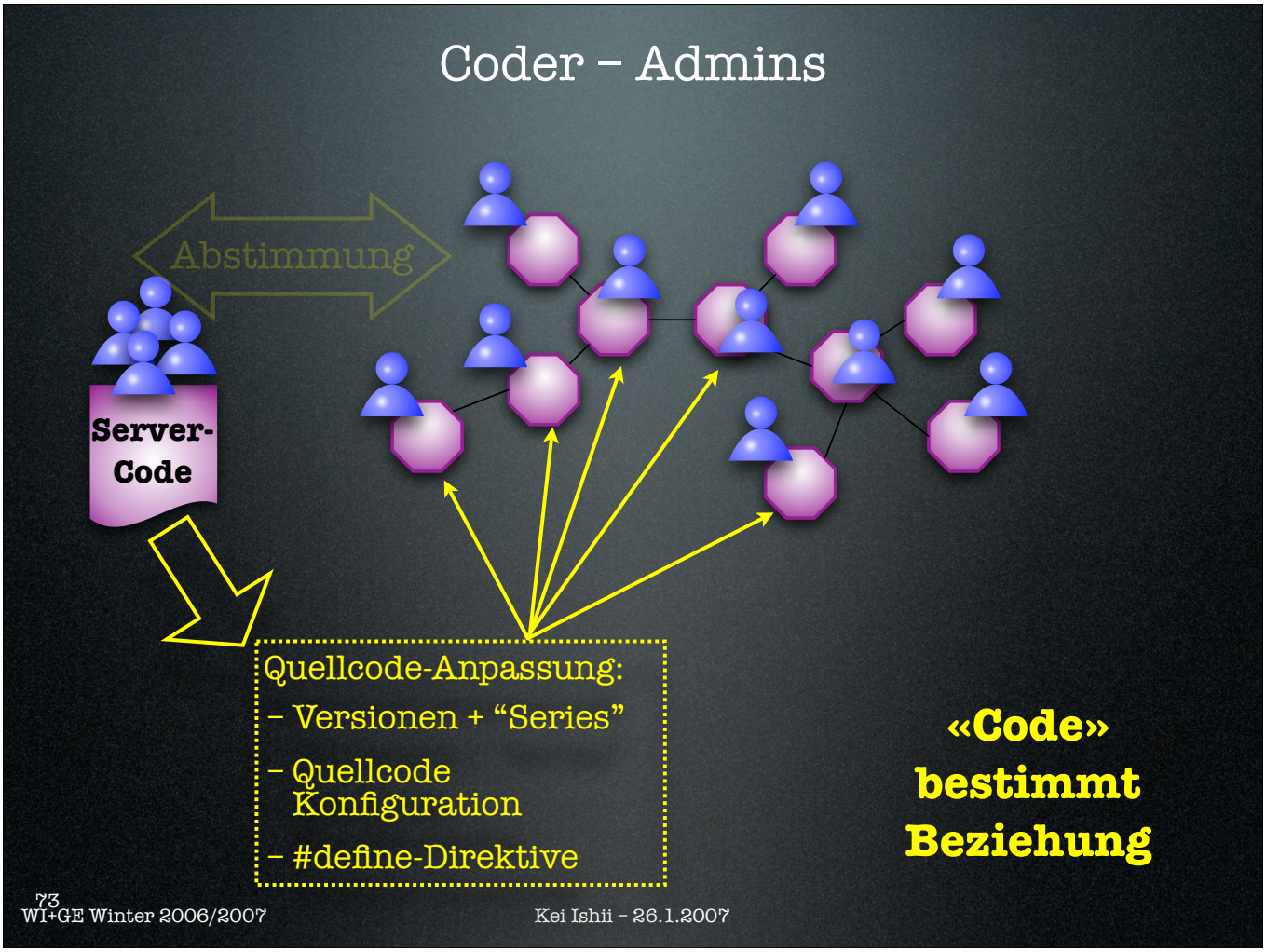
Bereits bei der Analyse der technischen Struktur des IRC kann man sehen, wie der «Code» Akteursrollen und die Verhältnisse zwischen ihnen definiert.
– Ein IRC-Netzwerk besteht aus einer Anzahl von miteinander verbundenen Servern, die jeweils von einem sogenannten IRC Administrator verwaltet werden.
–An dieses Server-Netzwerk können sich Benutzer mit IRC-Client-Programmen verbinden, um so mit anderen Teilnehmern zu chatten.
–Als weitere Gruppe gibt es die Programmierer, die die Server-Software weiterentwickelt

??Geltungsbereich, Quelle, Primäre Durchsetzung, Inhalt, Rahmen, Anpassung



Bemerkenswert hier ist das «Code»-geformte Verhältnis zwischen Programmieren und Administratoren:
Da der Servercode als offener Quellcode vorliegt, kann jeder Administrator für seinen Server beliebige Veränderungen vornehmen, und damit «Code»-Regulierungen der Programmierer kontrakarieren.

—Es muß also zwischen den Programmierern und Administratoren ein Abstimmungsprozeß erfolgen, damit die «Code»-Änderungen der Programmierer durch die Administratoren aktiviert werden.



Im Quellcode, dies zeigt meine Arbeit, kann man «Code»-Mechanismen identifizieren, die diesen Abstimmungsprozeß durch Anpassungsmechanismen unterstützen:
–Interoperable Versionen oder gar mehrere Server-software-Serien parallel entwickelt
–Konfigurationsmechanismen, in der vorgegebene Quellcode-Einstellungen vom Administrator verändert werden können
–Programmiersprachenkonstrukte, wie die #define-Direktive in C, mit der man funktionale Teile des Quellcode aktivieren oder deaktivieren kann
In dieser Weise erlauben die Programmierer den Administratoren, aus den implementierten «Code»-Regulierungen auszuwählen.
Die «Code»-Anpassungsmechanismen unterstützen den sozialen Abstimmungsprozeß zwischen Programmierern und Adminstratoren.

Fallbeispiel:
Max Users per Channel

Max users per channel Problem

”[a]fter a certain number of people jump in, the conversation often goes to hell.”

Jarkko Oikarinen
(IRC-”Erfinder”)

Konkretes Beispiel für Code-Regulierung, CodeGov.

– Zitat. Erfinder war Meinung, daß in Channels mit vielen Personen keine anständige Diskussion möglich.

Was tun?

Max users per channel
Erste «Lösung»

- IRC: Maximal 10 Users pro Channel
  - Ausnahme: Channels 1-9
- (AOL: 23 Users per Chat Room)

Beschränkung ist “hard-coded”

—Max 10 Users
—Ausnahme: 1-10
—Vergleich AOL (heute noch so?)

—Beschränkung/Regulierung hard-coded (fest verdrahtet) Code-Regulierung

Max users per channel Implementierung

```
#define MAXUSERSPERCHANNEL 10

#define is_full(ch, us)
            (!UnLimChannel((ch)) && ((us) >= maxusersperchannel))

#define UnLimChannel(x) (((x) > 0) && ((x) < 10))

 if (cptr == sptr && is_full(i, chptr->users)) {
     sendto_one(sptr, ":%s %d %s %d :Sorry, channel is full.",
         myhostname, ERR_CHANNELISFULL, sptr->nickname, i);
     return(0);
 }
 chptr->users++;
```


–In code als #define
–Weitere #define-Direktiven für Pseudo-Funktionen
(z.B. auch für die “public channels”)
–Bei Eintritt in Channel Abfrage über Anzahl aktiver members

Erstes define bedeutet:
==> Jeder Administrator könnte sofort ändern

Max users per channel
Bessere Lösungen?

- 1-9 have no limit
  10-19 have a limit of 20
  20-29 a limit of 15
  and the rest have a ten person limit.

- Negative channels: no limit
  0 to 99: no limit
  100 to 999: limit of 10 users
  1000 and above: no limit

Beispiele für Alternativvorschläge:

– Verteilung 1
– Verteilung 2

Max users per channel
Bessere Lösung.

- Beschränkung nicht mehr absolut festgelegt (hard-coded)
- Max Users für jeden Channel einzeln einstellbar (channel mode)
- Neue Rolle des Channel Operator entscheidet über Einstellung

Lösung (?)

- Beschränkung nicht mehr fest in Code

- Per Channel einstellbar (channel mode)

- Channel Operator als Entscheider (nicht normaler User)

Problemlösung
Wer entscheidet?

	IRC	AOL
	(Coder)	Firmenleitung
	IRC Admins	Angestellte
	Channel Operator	Channel Helper?
	Benutzer	

Vergleich, wer entscheidet.

In sozialer Struktur natürlich “Hierarchie”: Code/Admin, Chanop

Ähnlich wie in Firma (hier: hypothetisch AOL)

Wer entscheidet?

	IRC	AOL
	(Coder)	Firmenleitung
	IRC Admins	Angestellte
	Channel Operator	Channel Helper?
	User	

AOL-Lösung

AOL klar: Aus verschiedenen Gründen wird in Firmenleitung entschieden

Wer entscheidet?

	IRC	AOL
	(Coder)	Firmenleitung
	IRC Admins	Angestellte
	Channel Operator	Channel Helper?
	User	

AOL-Lösung

erstes Design

neue Lösung

erstes Design IRC: Coder haben eingefügt (i.e. Oikarinen),
IRC Admins könnten theoretisch entscheiden, da Zugang zum Code (#define etc.)

– Neue Lösung: Einführung einer neuen Rolle (Channel Operator). Jeder Channel (Op) kann jetzt selbst entscheiden. Flexibler, näher an Geschehen.

Wer entscheidet?

	IRC	AOL
	(Coder)	Firmenleitung
	IRC Admins	Angestellte
	Channel Operator	Channel Helper?
	User	

“Subsidia-risierung”

Pfeil: Tendenz näher an Benutzer, näher an konkretem Geschehen.
Vorteil: Nicht mehr so viel Entscheidungsaufwand
Nachteil: Weniger Kontrolle

—”Subsidiarisierung” mein Begriff.
Seht Euch “Subsidiarität” z.B. in Wikipedia an
•Verantwortung zu kleinerer Einheit
•Größere Einheit unterstützt

Entwicklung des «Code»

	Erstes Design	“Verbesserte” Lösung	Weitere Entwicklungen
Max Users per Channel	auf 10 festgelegt	per Channel einstellbar	--
Channel-Eigenschaften	fest verdrahtet für Channel	per Channel einstellbar	Speichern von Eigenschaften
Channel-Verantwortung	--	Channel Operator	Hierarchie von Channel Operatoren

funktionale Differenzierung

ENTWICKLUNG DER FUNKTIONALITÄTEN

Im zeitlichen Verlauf des IRC Anpassung des Channel-«Codes» an die soziale Situation:
–Beispiel Max Users
–erst auf 10 fest
–dann per Channel einstellbar
–Vergleichbares mit Channel-Eigenschaften [mündlich erklären]
–Channel-Verantwortung: Einrichten der neuen Rolle Chanop + weiteres

– Insgesamt läßt sich beobachten, was ich “funktionale Differenzierung” genannt habe

«Code» Governance

- Ansätze für «Code»-Designprinzipien
  - Subsidiarisierung
  - Funktionale Differenzierung
  - End-to-End Prinzip
- Beziehung zwischen technischer und sozialer Struktur
  - Coder – Administrator

Zusammenfassend sollten diese wenigen Beispiele einen Eindruck von der Fruchtbarkeit des Ansatzes vermittelt haben, den «Code», die technischen Strukturen, als Regulierungssystem zu betrachten:
–Es gibt im «Code», in der Technik Muster, Designprinzipien, die spezifische Wirkung für SozSituation haben: gezeigt “Subsidiarisierung”, “Funktionale Differenzierung”. Hierzu gehört auch “e2e”
Die Entwickler schaffen bestimmte Regelungsstrukturen

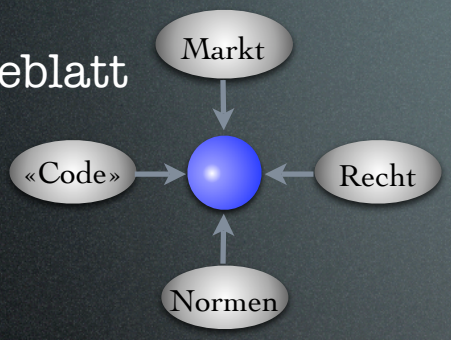
–Gleichzeitig sind die Beteiligten stets an die Eigenschaften des Codes gebunden, sowohl intendierte (reinprogrammierte) als auch unversehen entstehende.
In Beispiel das Verhältnis zw. Admins und Coder geprägt durch Zugang zu Source Code, daher explizite Mechanismen im Code, um den Admins Wahl zwischen Funktionen zu erleichtern.

Viewpoint | Peter Freeman and David Hart

A Science of Design for Software-Intensive Systems

Von den Alten lernen	Simon – Science of Design? Brooks – Design of Design?
End-to-End als politisches Prinzip	Förderung der Innovation durch anwendungsneutrales Netz?
Suche nach Regulierungsmustern im «Code»	Regulierungseigenschaften von Source code-Strukturen?

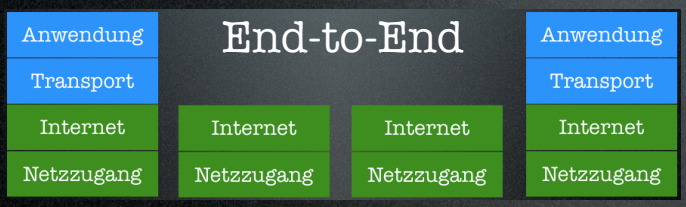
Kleeblatt

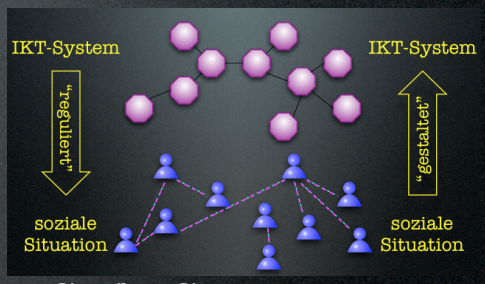


Lex Informatica

	Rechtssystem	DVD	Lex Informatica
Geltungsbereich	"Politische" Grenzen	"Code Regionen"	Netzwerk
Quelle	Legislative	DVD-Konsortium	Technologen
Primäre Durchsetzung	Exekutive Judikative	DVD + Gerät	Automatisiert, selbstausführend
Inhalt	Rechtsvorschriften Gerichtsurteile	Abspielkontrolle DVD-Standard	Techn. Funktionen Gewöhnl. Gebrauch
Rahmen	Rechtssystem	DVD-"System"	Architektur, Standards
Anpassung	Vertrag	(DVD-Laufwerk von Computern)	Konfiguration

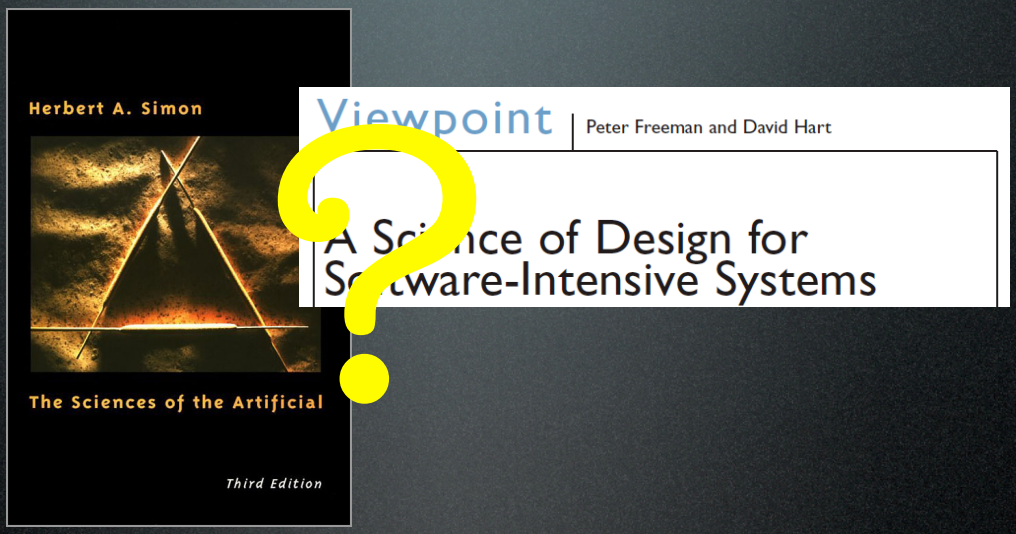
End-to-End





Code Governance

Science of Design
(Software-intensive Systems)



Informatik += ScienceOfDesign_SIS; //!!

Die Informatik muß ihre Forschung und Lehre um eine Theorie des «Designs» software-intensiver Systeme ergänzen

"Lehre"→SWE, InfoSys, System Eng.
"Gestaltung SIS"→ Science of Design – Software-Intensive Systems (SoD-SIS)

www.cyberinfrastructure.us

Wer zufällig nächsten Montag-Dienstag in Washington sein sollte...

Die Informatik mit Wirtschaftsinformatik

Recht

Ökonomie

Sozialw./ Politikw.

Science of Design

Wirtschafts-informatik

Angewandte Informatik

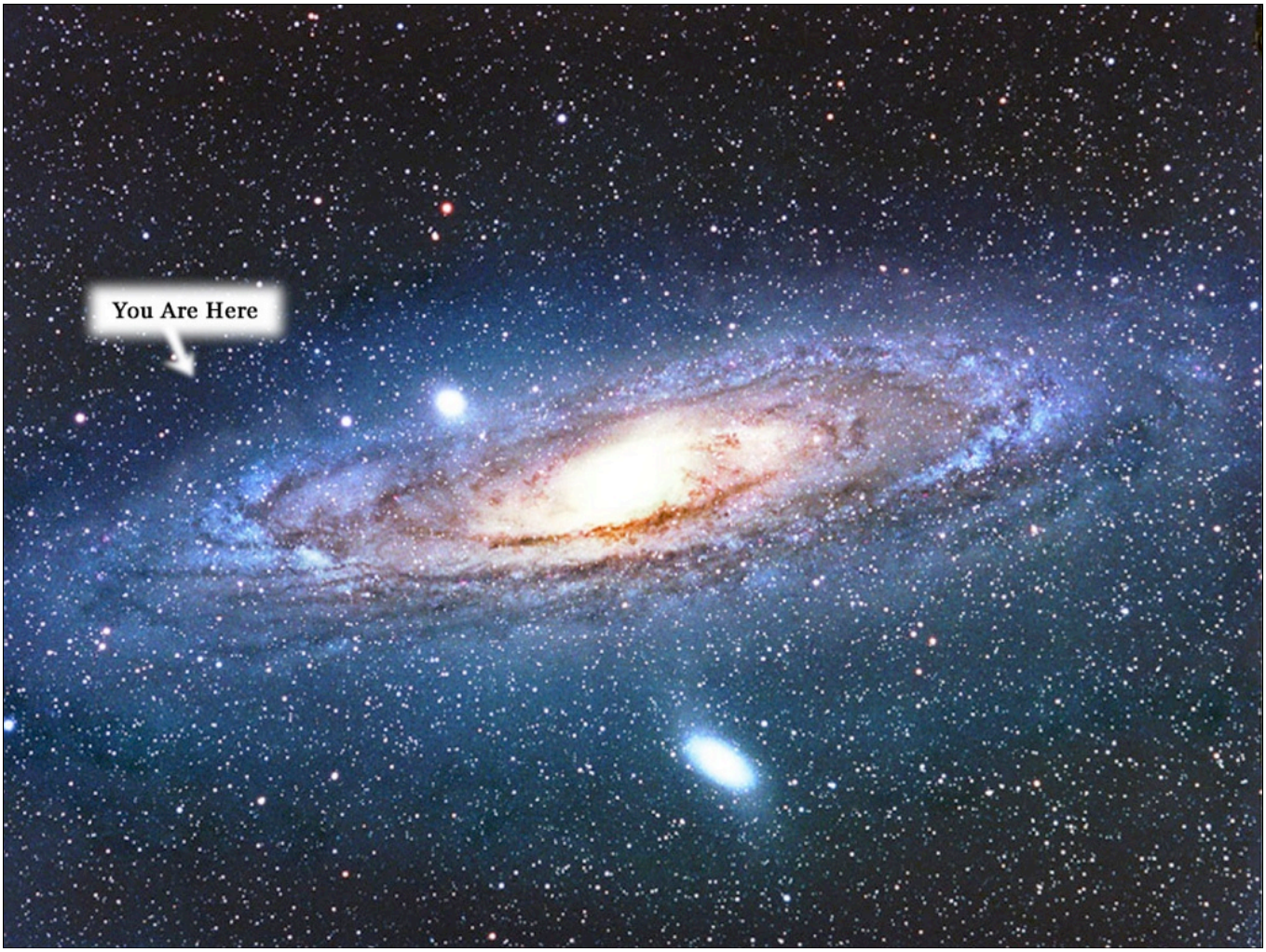
Technische I.

Praktische I.

Theoretische Informatik

Brueghel, Tower of Babel: <http://witewave.com.ne.kr/gallery/Bruegel_Tower_of_Babel1.jpg>

Tower of Inkompatibabel: <http://hello.eboy.com/eboy/wp-content/uploads/2007/01/PT_babeltower_01t.png>



http://freepatriot.com/imagewarehouse/YouAreHereGalaxy.jpg