V "Wissenschaftliches Arbeiten i. d. Informatik"

Was bedeutet "Wissenschaftliches Studium"?

Prof. Dr. Lutz Prechelt
Freie Universität Berlin, Inst. f. Informatik

- Wissenschaftliche Tätigkeit
  - Modi: Theorie, Konstruktion, Empirie
- Wissenschaft
  - Disziplinen, Ingenieurfach
  - "Erkenntnisstand"
- Ingenieurtätigkeiten
- Forschungsthemen-Beispiele
  - Peer my Proxy (kurz)
  - Defect-insertion circumstance analysis (lang)
- Wissenschaftliches Studium
  - Angewohnheiten
  - Literatursuche

Über dieses Modul

- Es heißt "Wissenschaftliches Arbeiten in der Informatik"
- Es besteht aus zwei Teilen:
  - dieser kleinen (1 SWS) Vorlesung
  - einem Proseminar
- Die Studienordnung beschreibt das Proseminar wie folgt:
  - "Zweck eines Proseminars ist
    - das Erlernen gründlicher **wissenschaftlicher Arbeit** unter Anleitung,
    - das Schreiben einer **wissenschaftlichen Arbeit** in Vorbereitung auf die Bachelorarbeit sowie
    - der Erwerb kommunikativer Kompetenzen und rhetorischer Fertigkeiten."

Umfrage

Welches ist eher Ihre Haltung?

- Haltung 1:
  "Wissenschaftliches Arbeiten?
  Klasse Sache!
  Das will ich auch lernen."

- Haltung 2:
  "Das interessiert mich eher wenig.
  Wissenschaft hat mit meinen Zielen kaum was zu tun."

Bildersuche "Wissenschaftler": Ergebnisse

https://www.google.de/search?q=wissenschaftler&source=lnms&tbm=isch&sa=X&ei=pvoPVYnVAcP5ygPcjoGICA&ved=0CAcQ_AUoAQ&biw=960&bih=1281 am 2015-03-23 von fu-berlin.de

Haftungsablehnung ("disclaimer")

- Dieser Foliensatz (und auch alle folgenden) verwendet nur die Perspektive der Informatik
  - Bitte nicht über die Informatik hinaus verallgemeinern
- Die Darstellung ist zudem stark verkürzt
  - Wer es genauer wissen will:
    Bitte ggf. die angegebenen Quellen verfolgen
  - und dann selber weiterbohren

Definition "wissenschaftliche Tätigkeit"

- "Wissenschaftliche Tätigkeit [ist] alles, was nach Inhalt und Form als ernsthafter planmäßiger Versuch zur Ermittlung der Wahrheit anzusehen ist"
  - Bundesverfassungsgericht, BVerfGE 35, 79
    - "Hochschul-Urteil", 1973
    - http://www.servat.unibe.ch/dfr/bv035079.html , Rn 128
- Ist also folgendes ein Beispiel?:
  - https://www.youtube.com/watch?v=-7usT_r1DGc

- Leider hat diese Form des Wissenserwerbs ihre Schwächen:
  - 40% der Versuchspersonen lösten z.B. folgende Aufgabe falsch
    - McCloskey and Kohl (1983), dort aber anders formuliert

The C-shaped tube you see below is lying flat on a table (this is an aerial view). A ball is placed in one end and launched so that it proceeds through the tube and out the other side.

Draw the path that the ball will take once it exits the tube, starting at the given point.

Riener, Proffitt, Salthouse in Psychonomic Bulletin & Review 12(4):740-745, 2005
http://link.springer.com/content/pdf/10.3758/BF03196766.pdf

Definition "wissenschaftliche Tätigkeit", erweiterte Form

Freie Universität Berlin

- *"Wissenschaftliche Tätigkeit ist alles, was nach Inhalt und Form als ernsthafter planmäßiger Versuch zur Ermittlung der Wahrheit anzusehen ist […].*
  Sie ist nach Aufgabenstellung und anzuwendender Arbeitsmethode darauf angelegt, neue Erkenntnisse zu gewinnen und zu verarbeiten, um den Erkenntnisstand der jeweiligen wissenschaftlichen Disziplin zu sichern oder zu erweitern"
  - Bundesarbeitsgericht (BAG)
    Urteil vom 1. Juni 2011, Az. 7 AZR 827/09, Rn 62
    https://openjur.de/u/597059.html

Arten von Arbeitsmethoden: Modi

Wissenschaftliche Tätigkeit findet in drei "Betriebsarten" statt:

- Theorie (T):
  - Das Entwerfen und geistige Untersuchen rein geistiger Systeme (Begriffe, Kalküle, Theorien)
- Empirie (E):
  - Das Untersuchen der Eigenschaften von Systemen (natürlichen oder künstlichen, aber nicht nur rein geistigen)
- Konstruktion (K):
  - Das Entwerfen und Realisieren technischer Systeme

Die Betriebsarten wechseln, doch in jedem Moment ist eine davon vorrangig (nach Art der aktuellen Aufgabenstellung)

Definition "wissenschaftliche Tätigkeit", erweiterte Form

Freie Universität Berlin

- *"Wissenschaftliche Tätigkeit ist alles, was nach Inhalt und Form als ernsthafter planmäßiger Versuch zur Ermittlung der Wahrheit anzusehen ist […].*
  Sie ist nach Aufgabenstellung und anzuwendender Arbeitsmethode darauf angelegt, neue Erkenntnisse zu gewinnen und zu verarbeiten, um den Erkenntnisstand der jeweiligen wissenschaftlichen Disziplin zu sichern oder zu erweitern"

✓ T,K,E

  - Bundesarbeitsgericht (BAG)
    Urteil vom 1. Juni 2011, Az. 7 AZR 827/09, Rn 62
    https://openjur.de/u/597059.html

Gemeinsames Merkmal der Aufgabenstellungen bei "Wissenschaft":

Freie Universität Berlin

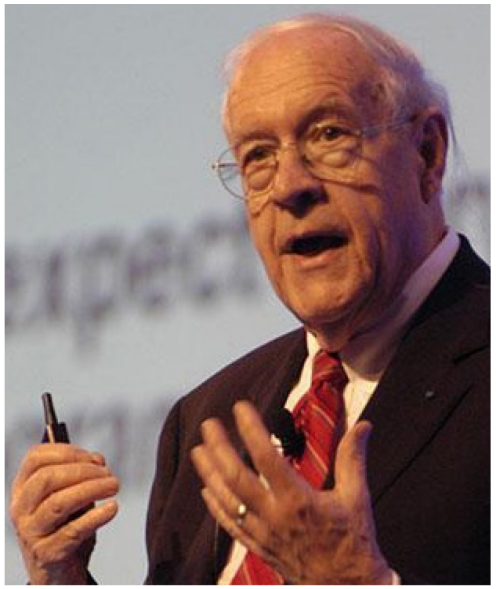
Wissenschaft ist eine Gemeinschafts-unternehmung

Gruppen von Aufgabenstellungen: Die Wissenschaftsgebiete und Disziplinen

- Naturwissenschaften:
  - Physik, Chemie, Biologie, Geologie u.a.
  - Untersuchen die Eigenschaften eines gegebenen Systems (Natur)
  - Hauptmodi: Emp. + Theorie
- Sozialwissenschaften
  - Soziologie, Volkswirtschaft, u.a.
  - Untersuchen die Eigenschaften menschlichen Verhaltens (Kultur)
    - viel variabler als der Rest der Natur
  - Hauptmodi: Emp. + Theorie
- Geisteswissenschaften
  - reine Mathematik, Philosophie, div. Philologien
  - Untersuchen geistige Systeme
  - Hauptmodus: Theorie
- Ingenieurwissenschaften
  - Masch.bau, Bauing.wesen, angewandte Mathematik, Medizin, Informatik u.a.
  - Konstruieren nützliche künstliche Systeme
    - erarbeiten wie das geht
  - Hauptmodi: Konstruktion und Empirie
- (Die Grenzen sind fließend!)

Informatik als Ingenieurwissenschaft

- Fred Brooks: *The Computer Scientist as Toolsmith II*, Communications of the ACM 39(3):61-68, 1996



- Hauptaussagen (S.61/62):
  - Die Informatik ist ein Ingenieurfach, keine Naturwissenschaft
    - Die Ziele sind ganz andere
    - Der Gütemaßstab ist die Nützlichkeit einer Erkenntnis, nicht ihre Neuheit
  - *"The scientist builds in order to study; the engineer studies in order to build."*
  - "Computer Science" ist deshalb ein schlechter Name
  - Dinge zu bauen ist großartig.
    - Wir sollten gegenüber den Naturwissenschaften selbstbewusst sein

Bildersuche "Ingenieur": Ergebnisse

- Das ist schreiender Blödsinn!
  - aber verbreitet: https://www.youtube.com/watch?v=g5ymi1ehypU

https://www.google.de/search?q=ingenieur&source=lnms&tbm=isch&sa=X&ei=NxAZVd-nBoOYPMuygZgB&ved=0CAcQ_AUoAQ&biw=1060&bih=1482 (2015-03-25)

Was hat "ingenieurmäßiges Arbeiten" mit "wissenschaftlichem Arbeiten" zu tun?

Freie Universität Berlin

Nütz-
lich-
keit

Ersetze "Ermittlung der Wahrheit" durch "Erzielen von Nützlichkeit".

Genauer bitte: Was tun (Software)Ingenieur/inn/e/n?

Freie Universität Berlin

- https://en.wikipedia.org/wiki/Engineer#Design
  - Informationen suchen und verarbeiten
  - Informationen weitergeben
  - Probleme definieren
  - Kriterien entwickeln
  - Lösungsvorschläge entwickeln, bewerten, abwägen, auswählen
  - Lösungen umsetzen und verbessern
  - Lösungen bewerten
- Was für Lösungen?
  - sozio-technische Systeme
  - vor allem deren technische Anteile
  - Ziele: Nützlichkeit, Kosteneffizienz
- All dies
  - auf Basis von tiefem Wissen
    - Studium, kont. Fortbildung
  - und hohem Können
    - (Studium), Praxis
  - fast immer in Teamarbeit
    - sehr kommunikations-intensiv!

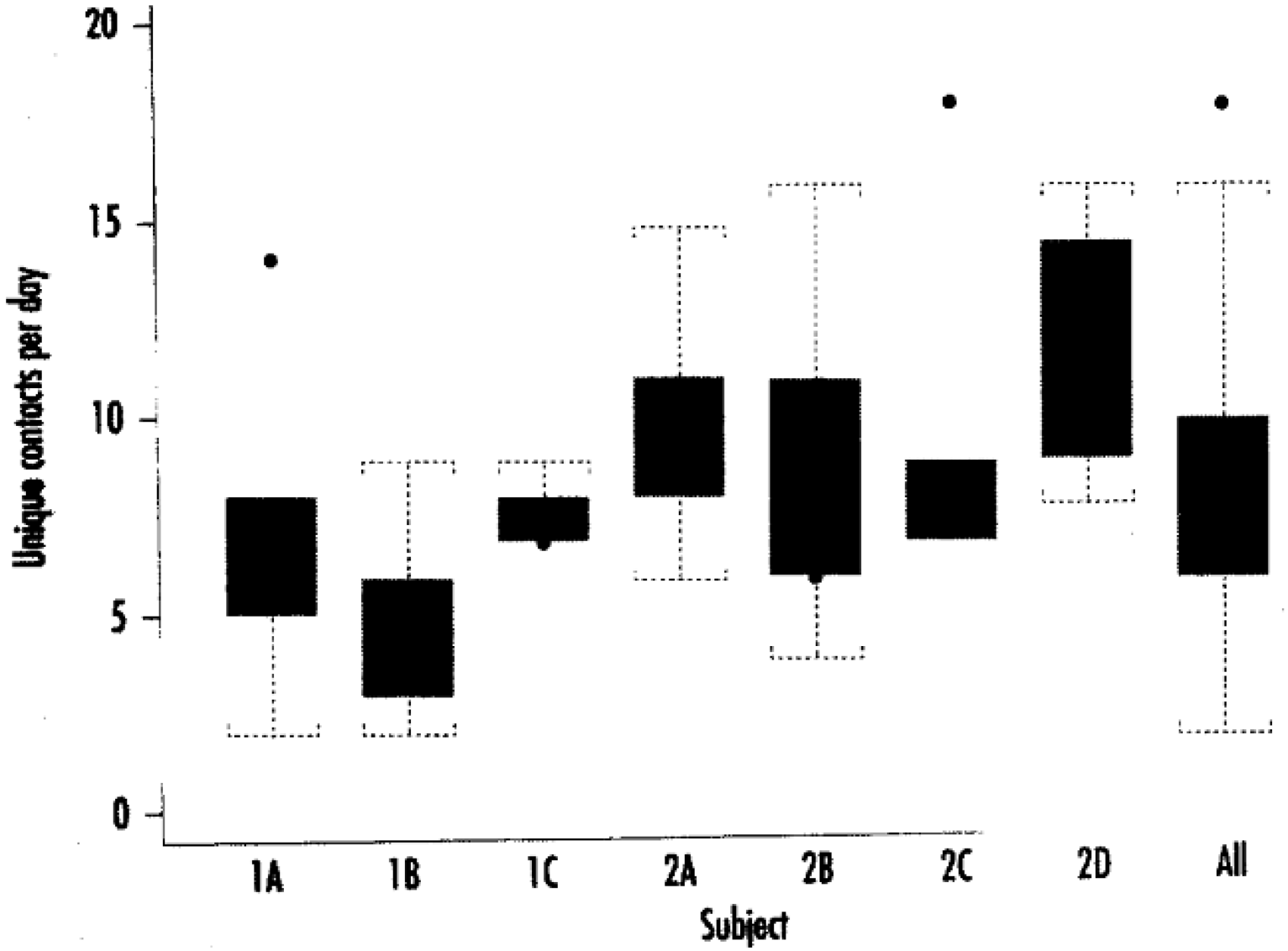
Was tun Ingenieur/inn/e/n?
Grobsicht und detailliertere Sicht (Beispiel)

Freie Universität Berlin

Diary		Observer's notes
0800-1800	Working on high-level design	0800
		0900
		1010
		1021
		1135
		1226
		1314
		1330
		1349
		1406
		1500
		1626
		1701

Dewayne E. Perry, Nancy A. Staudenmayer, Lawrence G. Votta: *People, Organizations, and Process Improvement*, IEEE Software July 1994

Was tun Ingenieur/inn/e/n?
Anzahl verschiedener Kontaktpartner pro Tag



Dito. Über 5 Tage hinweg; inklusive email, voice mail, Telefon und persönlich, aber abzüglich formaler Besprechungen und Testaktivitäten

Definition "wissenschaftliche Tätigkeit", erweiterte Form

Freie Universität Berlin

- *"Wissenschaftliche Tätigkeit ist alles, was nach Inhalt und Form als ernsthafter planmäßiger Versuch zur Ermittlung der Wahrheit anzusehen ist […].* ✓ "Study in order to build"
  Sie ist nach Aufgabenstellung und anzuwendender Arbeitsmethode darauf angelegt, neue Erkenntnisse zu gewinnen und zu verarbeiten, um den Erkenntnisstand der jeweiligen wissenschaftlichen Disziplin zu sichern oder zu erweitern"
  ✓ T,C,E
  ✓ Informatik
  - Bundesarbeitsgericht (BAG)
    Urteil vom 1. Juni 2011, Az. 7 AZR 827/09, Rn 62
    https://openjur.de/u/597059.html

Und was ist mit dem Erkenntnisstand?: Wissenschaft

Freie Universität Berlin

"Wissenschaft" (lt. Wikipedia):

1. *Stand* der Erkenntnisse über die wesentlichen Eigenschaften der Natur, Technik, Gesellschaft und des Denkens, das in Form von Begriffen und Theorien fixiert wird.
2. Der *Prozess* des Forschens und Erkennens in einem bestimmten Bereich, der ein begründetes, geordnetes und gesichertes Wissen hervorbringt

- Wissen ←→ Erkenntnisstand
  1. Fixiert/dokumentiert durch wissenschaftliche Artikel und Bücher (v.a. Dissertationen)
  2. Geordnet: z.B. durch Teildisziplinen und thematische Zeitschriften und Konferenzen
  3. Begründet, gesichert: z.B. durch Begutachtung wissenschaftlicher Artikel

Siehe Einheit 3: "Wissenschaftsbetrieb"

http://de.wikipedia.org/w/index.php?title=Wissenschaft&oldid=137189665

Ein Blick in die wissenschaftliche Informatik-Literatur

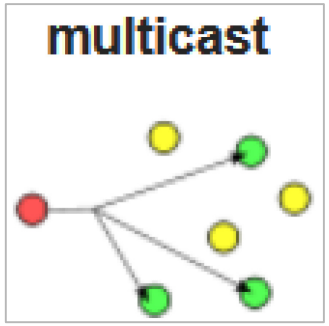
Freie Universität Berlin

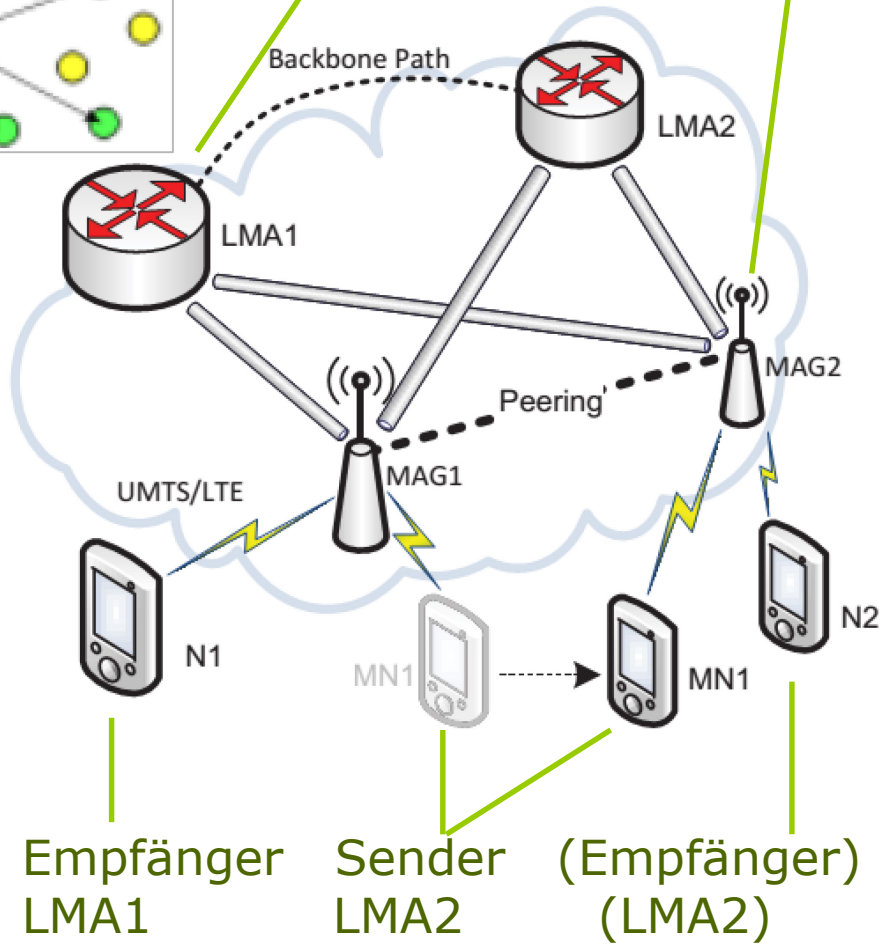
https://www.google.de/search?q=%22journal+of%22&source=lnms&tbm=isch&sa=X&ei=DDoQVcrGEKncywOB_YKACA&ved=0CAcQ_AUoAQ&biw=960&bih=1311#tbm=isch&q=%22journal+of%22+software am 2015-03-23

2 Beispiele aus unserem Institut:
1. Peer my proxy (Mobile IP Multicast)

Modi K+T, Netzwerke:

- Thomas C. Schmidt, Sebastian Wölke, Matthias Wählisch: *Peer my Proxy: A Performance Study of Peering Extensions for Multicast in Proxy Mobile IP Domains,* Proc. 7th IFIP Wireless and Mobile Networking Conference (WMNC 2014), IEEE Press, May 2014.
  - AG-Webseite
  - Verlagswebseite



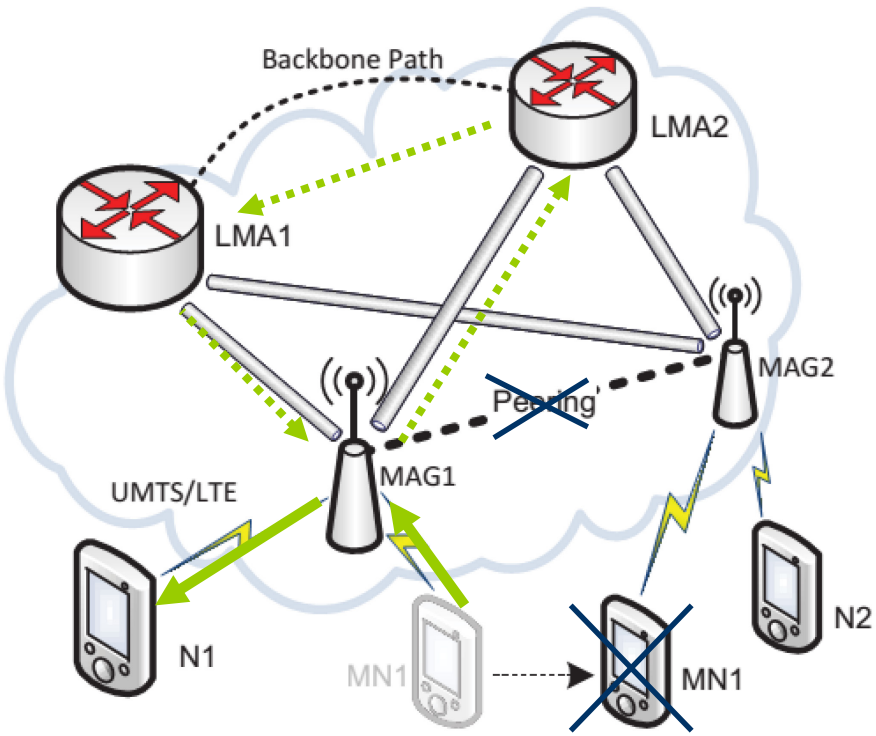


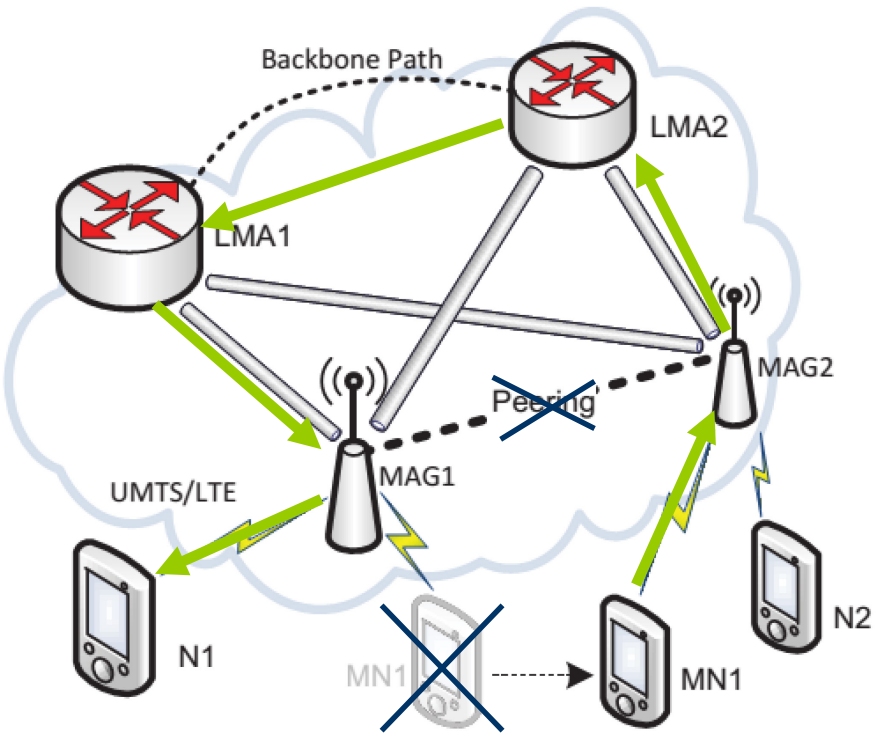
LMA: local mobility anchor MAG: mobile access gateway

1. Peer my proxy (Mobile IP Multicast): Routen **vor** Einführung des Peering

Freie Universität Berlin

- Vor Bewegung von MN1
  - MAG optimiert lokal
- Nach Bewegung von MN1
  - Rückfall auf logische Sicht



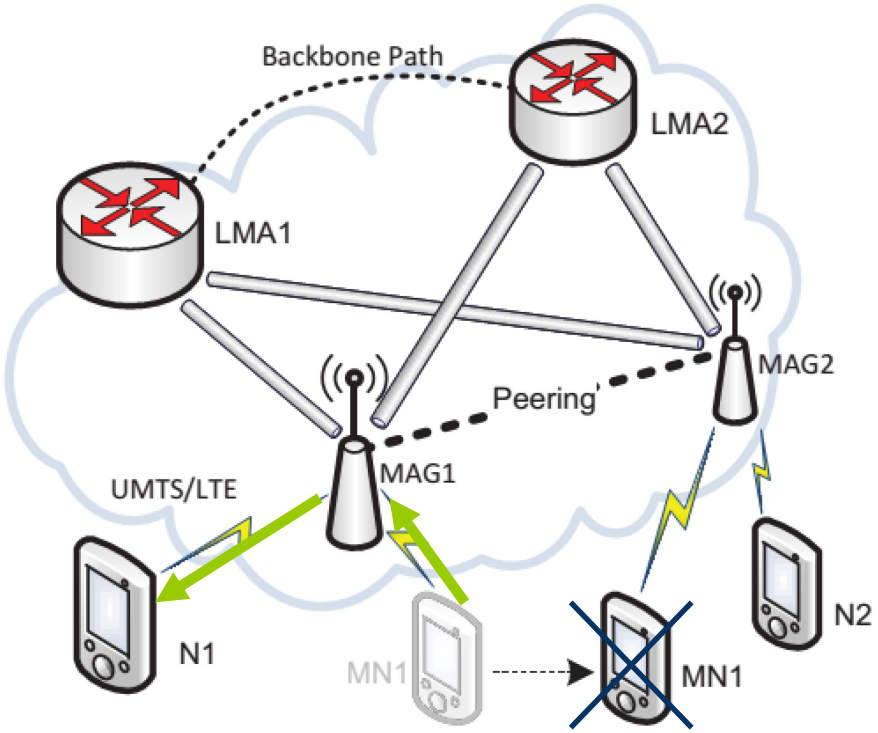


=entfällt!

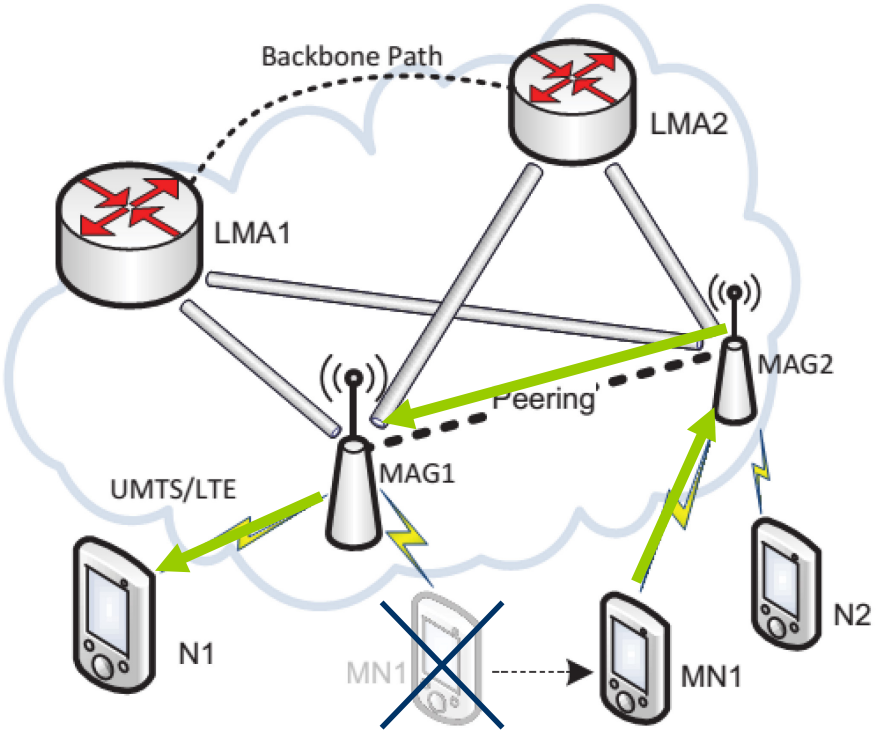
1. Peer my proxy (Mobile IP Multicast): Routen **nach** Einführung des Peering

Freie Universität Berlin

- Vor Bewegung von MN1 (wie gehabt)
  - MAG optimiert lokal



- Nach Bewegung von MN1
  - MAGs optimieren gemeinsam



1. Peer my proxy (Mobile IP Multicast): Emulated sender handover **without** peering

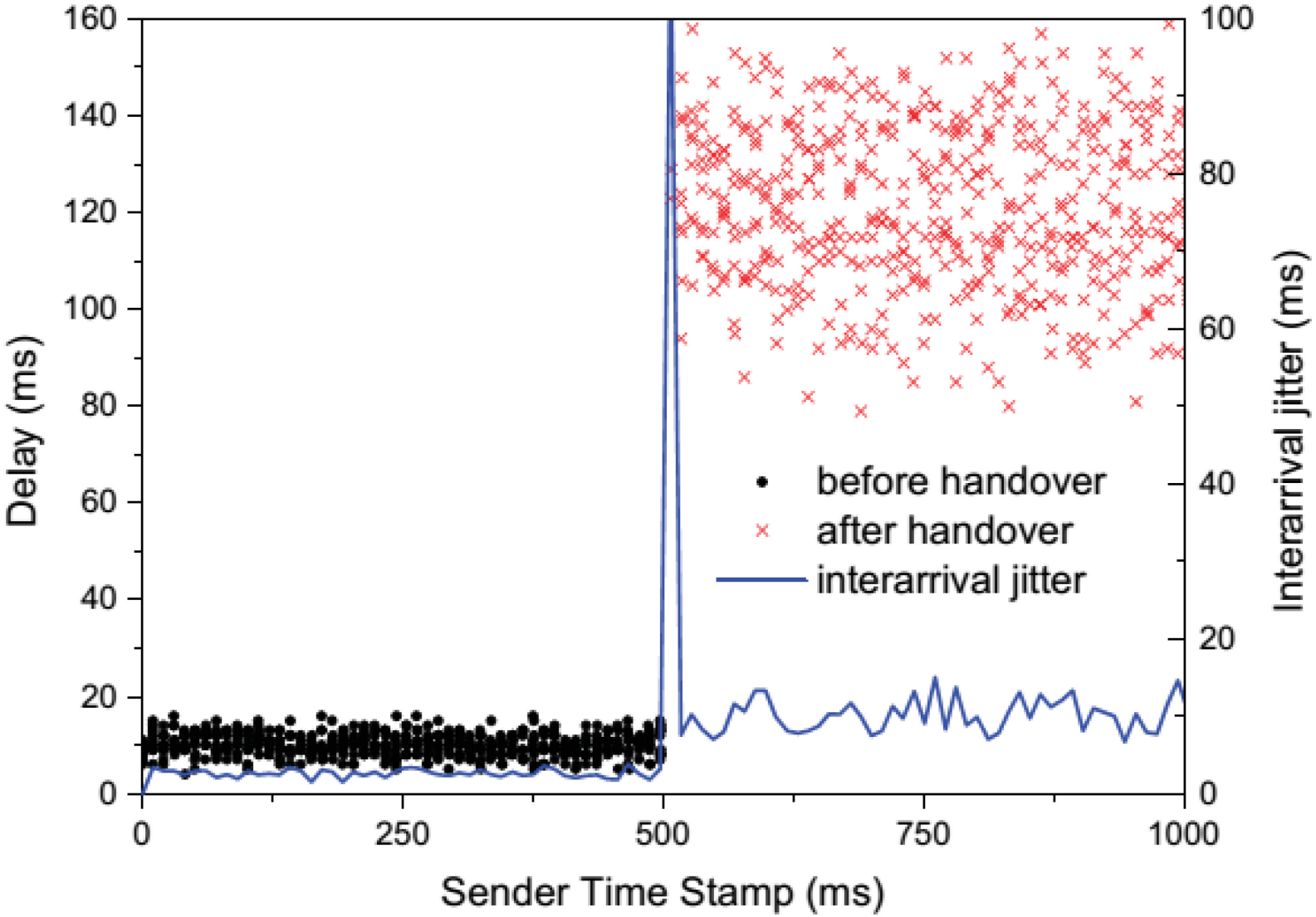


Figure 9a

1. Peer my proxy (Mobile IP Multicast): Emulated sender handover **with** peering

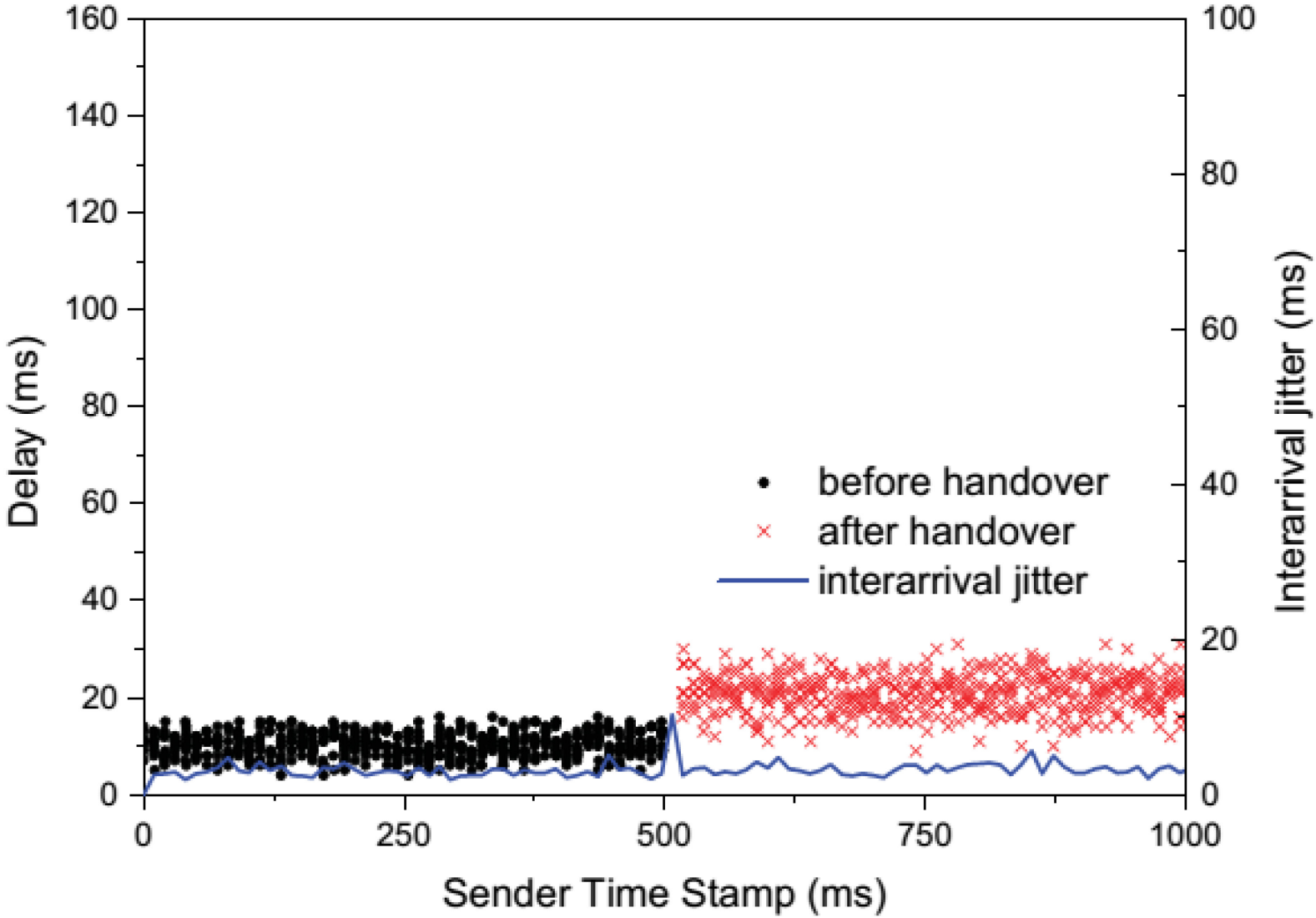


Figure 9b

1. "Peer my proxy"-Artikel: Die Zusammenfassung

Freie Universität Berlin

Abstract—Proxy Mobile IPv6 (PMIPv6) and its multicast extensions have been designed by the IETF as a deployment friendly mobility scheme. Although easy to implement, the basic multicast proxy solution suffers from unwanted delay and jitter due to suboptimal traffic flows. In this paper, we recap recent IETF work on peering extensions for multicast proxies and make the following two contributions. First we introduce the design and implementation of a highly flexible, open proxy that allows for dynamic reconfiguration at runtime. In particular, the system can support a variety of functional extensions including peering. Second we report on extensive performance measurements of proxy peering in LTE and UMTS type networks. Our findings indicate that a transparent deployment of the peering option significantly smoothes handovers and chokes delay variation throughout the access network.

Peer my Proxy – A Performance Study of Peering Extensions for Multicast in Proxy Mobile IP Domains

Thomas C. Schmidt, Sebastian Wölke
Hamburg University of Applied Sciences
INET Research Group, Department Informatik
Berliner Tor 7, 20099 Hamburg, Germany
t.schmidt@ieee.org, sebastian.woelke@haw-hamburg.de

Matthias Wählisch
Freie Universität Berlin
CST Research Group, Institut für Informatik
Takustr. 9, 14195 Berlin, Germany
waehlisch@ieee.org

Abstract—Proxy Mobile IPv6 (PMIPv6) and its multicast extensions have been designed by the IETF as a deployment friendly mobility scheme. Although easy to implement, the basic multicast proxy solution suffers from unwanted delay and jitter due to suboptimal traffic flows. In this paper, we recap recent IETF work on peering extensions for multicast proxies and make the following two contributions. First we introduce the design and implementation of a highly flexible, open proxy that allows for dynamic reconfiguration at runtime. In particular, the system can support a variety of functional extensions including peering. Second we report on extensive performance measurements of proxy peering in LTE and UMTS type networks. Our findings indicate that a transparent deployment of the peering option significantly smoothes handovers and chokes delay variation throughout the access network.

Keywords—Seamless mobility management; handover measurement; mobile multicast; wireless multimedia networking

I. Introduction

Prevalent infotainment offers like IPTV and other multimedia applications require scalable and efficient network services throughout the mobile realm. IP layer multicast [1] has been the key technology for distributing such multimedia services in a highly scalable fashion. It is widely implemented for fixed networks. In the wireless world, neither typical flash crowds of mobile users, nor the dense dissemination of infotainment channels can be supported at reasonable cost without multicast. Still the IETF has only recently released work on standard protocols for multicast mobility.

Proxy Mobile IPv6 (PMIPv6) [2] refines Mobile IPv6 (MIPv6) [3] by network-based management functions that enable IP mobility for a host without its active participation in any mobility-related signaling. Additional network entities called the Local Mobility Anchor (LMA) and Mobile Access Gateways (MAGs) are responsible for managing IP mobility on behalf of the mobile node (MN). PMIPv6 can be extended to support multicast services by deploying MLD proxy instances [4] at MAGs [5].

Even though easy to deploy and simple, the multicast basic solution for listeners leaves challenges open, as it may exhibit suboptimal routes and can lead to unwanted delay and jitter throughout the wireless access network. Handovers in particular may change the service quality at the receiver and thereby alienate users. Conversely, mobile multicast sources can cause a major service degradation for thousands of listeners, when changing from an efficient topological attachment to a disadvantageous network position. It is the objective of the present paper to present and analyze a deployment-ready solution that prevents service instabilities and enables an optimized traffic scheme.

In this paper, we take up the peering proposal for MLD Proxies presented in [6] and introduce Mcproxy[1], a fully compliant open source implementation of such an extended MLD Proxy, which has deployment in industry production systems. We show that this software has near-unicast forwarding performance and use it to experimentally validate the service gain of our peering scheme. Our findings indicate that a direct peering among multicast proxies in the access network will eliminate service fluctuations and unwanted traffic flows throughout a PMIPv6 domain. Operators who deploy PMIP multicast with peering proxies are thus enabled to offer homogeneously fast and stable multimedia services to their customers.

The remainder of this paper is structured as follows. We summarize the problem of multicast in PMIP and discuss related work in Section II. Architectural and implementation concepts of the Mcproxy solution and its evaluation are introduced in Section III. Section IV presents the performance measurements of proxy peering and discusses the results. We conclude with an outlook on future work in Section V.

II. The Problem of Multicast in PMIPv6 and Related Work

Multicast mobility has been subject to intensive studies and research proposals since the early days of the mobile Internet design [7], [8]. Various solutions have been proposed which were either built on top of multicast routers, or conjointly managed by mobility anchors. Over the years, not any one of these dedicated multicast mobility management schemes was adopted in standardization work nor in practical deployment—except for the elementary approach of bidirectional tunneling of multicast traffic via the Home Agent [9]. Reasons for this disappointment must be considered twofold, first in the hesitant

[1] See http://mcproxy.realbert.org.

1. Peer my proxy (Mobile IP Multicast): 4-Aspekte-Zerlegung

Freie Universität Berlin

- Disziplin:
  - Informatik, Bereich Netzwerkprotokolle
- Technikstand zuvor:
  - Multicast-Verkehr wird sehr beeinträchtigt, falls sich der Sender wegbewegt
- Aufgabenstellung:
  - Robustere Lösung finden und realisieren
  - Deren Qualität aufzeigen
- Arbeitsmethoden (K, E):
  - Verfahren mit Peering ausdenken (Autoren 1+3)
  - Softwarerealisierung
    - Bachelorarbeit, als Open Source (Autor 2)
  - Validierung per Emulation (Autoren 1+3)
- Was seitdem geschah:
  - Das Verfahren ist inzwischen der RFC 7287
  - Die Software wird von Herstellern wie Qualcomm in Produkten eingesetzt

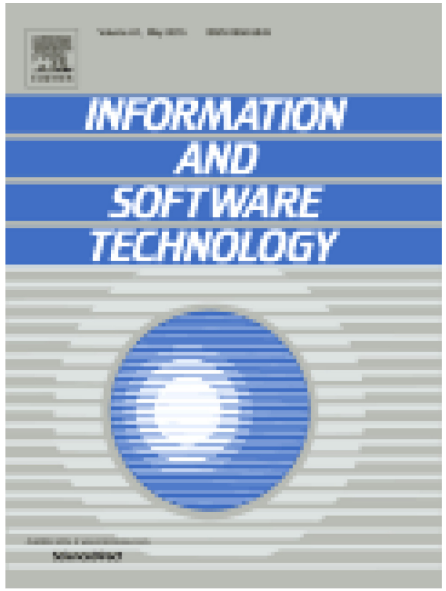
2. Defect-insertion circumstance analysis (DICA)

Freie Universität Berlin

Modus E, Software Engineering:

- Lutz Prechelt, Alexander Pepper:
  Why Software Repositories Are Not Used For Defect-Insertion Circumstance Analysis More Often: A Case Study.
  Information and Software Technology 56(10):1377-1389, October 2014.
  - AG-Webseite , Verlagswebseite

- Dies hatte mal begonnen im Frühjahr 2011 wie folgt:



2. Defect-insertion circumstance analysis (DICA): Ein Student kommt in eine Sprechstunde

Freie Universität Berlin

Alexander Pepper:

- "Ich möchte gern meine Diplomarbeit bei Infopark AG machen."
  - Daten des Versionsarchivs und der Defektdatenbank analysieren.
  - Ziel: Prozessverbesserungen erkennen.
- Nach Nachweis des Literaturstudiums stimmt Prof. Prechelt zu
  - Diplomarbeit wird mit gelegentlicher Beratung erfolgreich durchgeführt.
    - weitgehend selbständig

http://www.inf.fu-berlin.de/w/SE/ThesisDefectIdentification
https://avatars0.githubusercontent.com/u/86275?v=3&s=460 (2015-03-30)

2. Defect-insertion circumstance analysis (DICA): Hintergrund: Versionsablage

Freie Universität Berlin

- Eine Software besteht aus vielen Quelldateien
- Eine Versionsablage (z.B. Git, Subversion) registriert für jede Änderung (engl. *check-in, commit*):
  - welche Dateien geändert?
    - welche Stellen genau?
  - von wem? wann? Zweck (per *commit message*)?
  - ➔ Änderungen sind nachvollziehbar (*diff*)
- Gleichzeitige Änderungen werden möglichst kombiniert
  - ➔ Unverzichtbares Koordinationswerkzeug

```
@@ -1,3 +1,3 @@
 def say_hello(hour)
-   "Good "+ (hour <= 12 ? "day":"night")
+   "Good "+ (hour < 12 ? "day":"night")
 end
```

z.B.
"fixed #533 (noon bug)"

Eintragsnummer Defektdatenbank

Diff-Beispiel: Diplomarbeit Pepper Seite 33

2. Defect-insertion circumstance analysis (DICA): Hintergrund: Defektdatenbank

Freie Universität Berlin

- Eine Defektdatenbank (issue tracker, bug tracker) sammelt
  - aktuelle und frühere Aufgaben (Defekte, Featurewünsche u.a.)
    - **Nummer**, Autor, Beschreibung, Anhänge, betroffene Komponente, Version, OS, Erstelldatum u.a.
  - deren Status
    - neu/in Arbeit/fertig/geschlossen, Zuständigkeit, Diskussionsbeiträge, Anhänge, u.a.

- Beispiel:
  - https://jira.sakaiproject.org/browse/YAFT-66?jql=

- Mit solchen Daten kann man viele interessante Analysen machen
  - (darüber gibt es seit 2004(?) eine ganze Konferenz, genannt MSR: Mining Software Repositories)
- z.B. im Prinzip folgende:
  1. Ordne Änderungen ihren Defekt-DB-Einträgen zu
  2. Identifiziere damit defektkorrigierende Änderungen
  3. Untersuche, welche Zeilen f.d. Korrektur geändert wurden
  4. Finde zugehörige Änderungen, die diese Zeilen eingefügt haben
  5. Nimm an, das dies die defekt-einfügenden Änderungen waren
  6. Untersuche die Eigenschaften solcher defekt-einfügenden Änderungen

  (3, 4, 5 heißen zusammen auch *SZZ-Algorithmus*)

2. Defect-insertion circumstance analysis (DICA) Fiktives Beispiel

Freie Universität Berlin

- Defektdatenbank enthält:
  `#123, Bug, "Sroinc not frobnicated completely"`
- Versionsablage enthält (und zwar bei Änderung **`e7fb82a`**) als Commit-Nachricht:
  `"fixed #123: full frobnication"`
  - und geändert wurde dabei nur **`file_name.rb`**, Zeile 2
- Autorenzuordnung des Zustands von **`file_name.rb`** <u>vor</u> **`e7fb82a`** ergibt:

```
$ git blame HEAD~1 -- file_name.rb
^0ca0768 User V   2011-09-20 15:37:24 +0200 1) def say_hello(hour)
fd2d09ec User W   2011-09-20 15:37:47 +0200 2)   "Good "+ (hour <= 12 ? "day":"night")
^0ca0768 User V   2011-09-20 15:37:24 +0200 3) end
```

- Änderung **`fd2d09ec`** war also wohl die Defekteinfügung
  - vielleicht auch nicht, z.B. nur "day" statt "morning" eingetragen?
  - Eigenschaften dieser Änderung (Umfang, Zeitpunkt, Autor/in, Mitgeändertes u.a.) zählen also zu den Erkenntnissen

2. Defect-insertion circumstance analysis (DICA)
Vorhaben von Infopark AG, Ergebnis

- Infopark wollte dies einsetzen,
  um die folgende und ähnliche Fragen zu beantworten:
  - Zu welchen Tageszeiten sind unsere Änderungen besonders defektträchtig?

- Doch am Ende von Peppers Arbeit stand die Einsicht:
  - Die Antwort bekommt man nicht raus!

  Warum nicht?
  - Weil man den Ergebnissen nicht trauen sollte

- Für Infopark war die Arbeit also nicht relevant

2. Defect-insertion circumstance analysis (DICA). Umwidmung von Anwendung zu Wissenschaft

- Für Infopark war die Arbeit also nicht relevant
- Nächste Frage für Pepper und Prechelt war:
  - Wie ist das denn bei anderen Anwendern von DICA?
    - Recherche in den Artikeln der letzten 2 MSR-Konferenzen
  - Antwort: Es scheint keine zu geben!

➔ Forschungsfrage:

- **Warum wird DICA so selten eingesetzt?**
  - Relevant für die MSR-Forscher, denn die glauben, dass ihre Verfahren praxisrelevant sind

- Forschungsmethode: Fallstudie (case study)
  - Benutze den Infopark-Versuch als beispielhafte Messlatte, um das Gewicht aller denkbaren Gründe zu vergleichen
    - "Theorie-Triangulation"
    - Jeder Grund liefert eine Mini-Theorie für die Antwort

2. Defect-insertion circumstance analysis (DICA)
Wissenschaft: Gründeliste, entdeckte Probleme

Freie Universität Berlin

- Niemand benutzt das.
  Frage: Warum nicht?
  - R1 Technik unbekannt
  - R2 zu schwierig anzuwenden
  - R3 zu viel Arbeit
  - R4 Ergebnisse unnütz
  - R5 Verlässlichkeit unklar
  - R6 Verlässlichkeit zu niedrig
- Studie vergleicht Gründe an Fallbeispiel von Infopark AG
  - Historie: 11 Jahre, 45000 Versionen
  - 3200 Defekte (→DefektDB), 4500 Korrekturversionen
- Schwierigkeit 1: Viele Defekteinträge falsch markiert
  - Defekt ←→ Feature
    - 9400 Einträge!
  - Defekt → Korrekturversion
    - 400 Mio. Möglichkeiten!
    - nicht manuell prüfbar
- Schwierigkeit 2: SZZ-Algor. ordnet oft falsche Quellversion zu
  - wg. sonstiger Änderungen
  - Aber wie oft?
    - Zehntausende Versionen betroffen!
    - nicht manuell prüfbar

2. Defect-insertion circumstance analysis (DICA)
DICA-Fehler-Bsp: Defekträchtige Tageszeiten

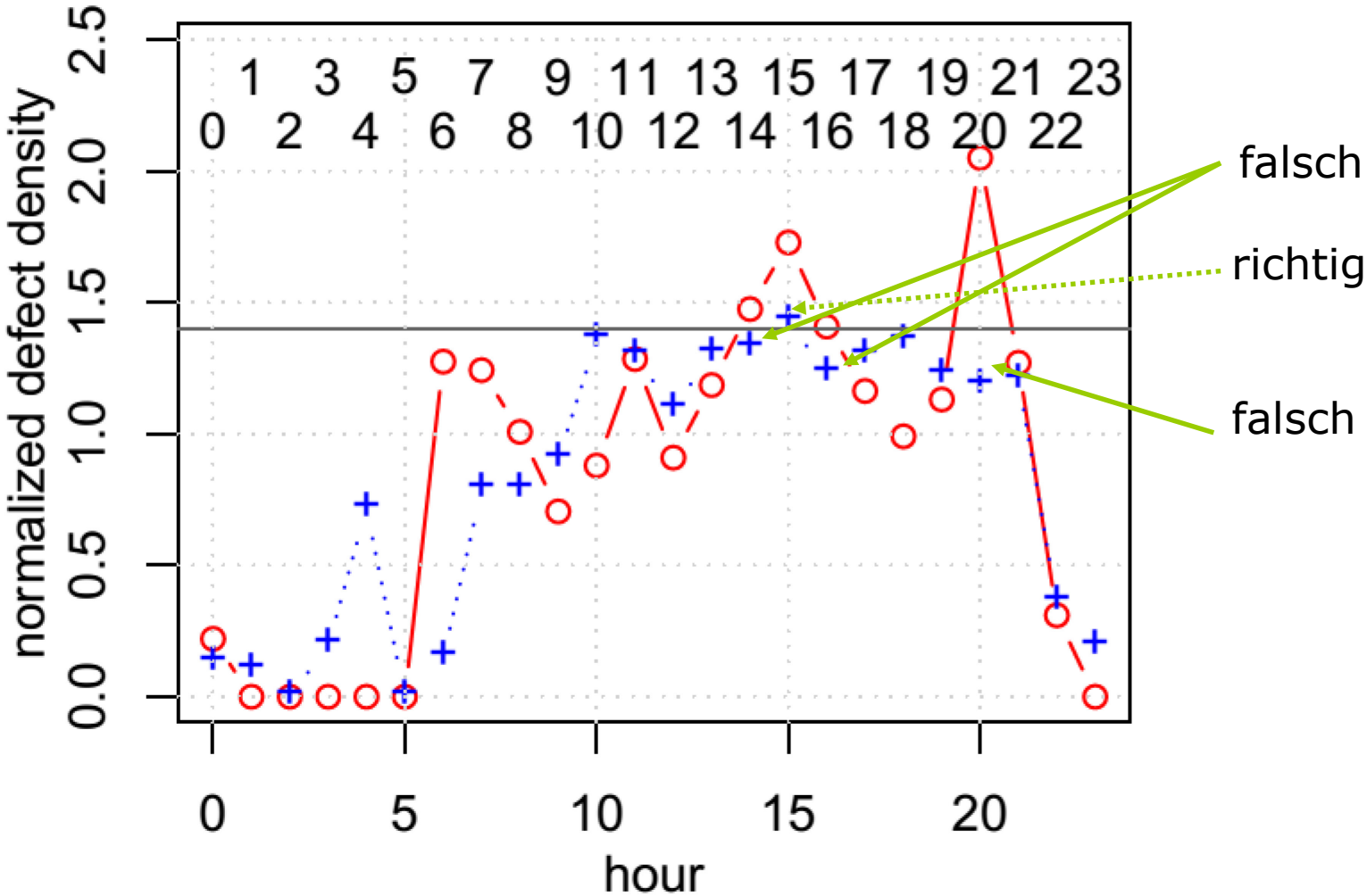


Figure 6: Normalized defect insertion densities for the improved set (circles) and the naive set (plus signs) for the 24 hours of the day; in natural order. 1 represents the median of the respective set. The cutoff is at 1.4 medians.

2. Defect-insertion circumstance analysis (DICA) Analysemethodik, Resultat

Freie Universität Berlin

- Fehlerschätzung durch manuell geprüfte Stichprobe
  - sagt: da sind viiiele Fehler
- Verbesserungsheuristik: Gucke nur "kleine" Änderungen an
  - → "improved set"
  - ohne Heuristik: "naive set"
- Glaubenssprung nötig:
  - Benutze improved set als Ersatz für die (unbestimmbare) Wahrheit
  - Bewerte damit naive set
  - Nimm an, dass Unterschied so groß ist wie von improved set zu Wahrheit

- Berechne damit, wie oft Prozessänderungsentscheidungen falsch wären

- **Antwort:** R5 (Verlässlichk. unklar) u. R6 (Verlässlichk. zu gering) dominieren
  - Nichtverwendung von DICA ist vernünftig

Kompliziert!

2. Defect-insertion circumstance analysis: 4-Aspekte-Zerlegung

Freie Universität Berlin

- Disziplin:
  - Informatik, Bereich Analyse des Entwicklungsprozesses
- Erkenntnisstand zuvor:
  - SZZ-Algorithmus erlaubt (mit Fehlern) die Ermittlung häufiger Umstände von Defekteinfügung
- Aufgabenstellung:
  - Niemand benutzt das. Warum nicht?
- Arbeitsmethoden (E):
  - Denkbare Gründe auflisten
  - Industrielle Fallstudie mit echtem Erkenntnisinteresse begleiten
    - und auf Probleme achten
  - Per Reflektion lösbare von schwerwiegenden Problemen trennen

2. Defect-insertion circumstance analysis (DICA) Anmerkung

- Ist das ein typisches Beispiel für Informatik-Forschung? Nein
  - Negatives Resultat ("Warum etwas nicht geht") ist selten
  - Wegen Nützlichkeitsorientierung ungeliebt
  - Aber wissenschaftlich sehr wohl wertvoll!
    - Sonst würden Praktiker/innen das immer wieder neu selbst herausfinden müssen

Qualitätsmaßstäbe für Forschungsergebnisse: Glaubwürdigkeit und Relevanz

Freie Universität Berlin

- Glaubwürdigkeit:
  - Wie verlässlich sind die gemachten Schlussfolgerungen?

  ➔ Oft sehr wackelig, weil allgemeiner als die Untersuchung hergäbe
  - Aber zumindest die Fakten sind in wiss. Artikeln i.d.R. korrekt
  - Und viele relevante Fakten werden auch angegeben

- Relevanz:
  - Wie wertvoll (nützlich) sind die Erkenntnisse für praktizierende Ingenieur/inn/e/n?

  ➔ In wiss. Artikeln oft niedrig

(Eigentlich wissen wir dafür zu wenig, aber dennoch:)

- Für die 2 obigen Beispiele: Argumente für/gegen
  - ihre Glaubwürdigkeit?
  - ihre Relevanz?

Hausaufgabe

- Lesen Sie den Artikel PrePep14:

  Lutz Prechelt, Alexander Pepper:
  Why Software Repositories Are Not Used For Defect-Insertion Circumstance Analysis More Often: A Case Study.
  Information and Software Technology 56(10):1377-1389, October 2014.

  - Abschnitte 1, 2, 3.2, 3.3, 3.4, 3.6, 6, 7.1, 7.2, 7.7, 7.8, 8, 9, 10 gründlich
  - Den Rest mindestens ungefähr

- Das ist wichtig, denn wir reden ausführlich weiter über diesen Artikel in den Vorlesungen Nummer
  - 2: Literatursuche
  - 3: Publikationssprozess, Begutachtungsprozess
  - 4: Qualitätsmerkmale wissenschaftlicher Texte

Ende von Teil 1

Was wissen wir jetzt?

1. Wissenschaftliche Tätigkeit = planmäßiger Versuch zur Ermittlung der Wahrheit
   - Aber: Das ist individuell oft unzuverlässig
   - Deshalb *gemeinschaftlich* organisiert: Wissenschaft
2. Drei Arbeitsmodi: Theorie, Konstruktion, Empirie
   - Unterteilt in Disziplinen
3. Informatik ist ein Ingenieurfach
4. Unsere "Wahrheit" sind Methoden zu Erzielung von Nützlichkeit
5. **Der Übergang von Ingenieurpraxis zu Wissenschaft ist nicht weit**

Also was heißt nun "wissenschaftliches Studium"?

Freie Universität Berlin

Studienziel:

- Systematisches Suchen nach Nützlichkeit (für Informatiksysteme) erlernen

Dazu gehört (beim Bachelor):

- Grundwissen ●●●
  - Kernkonzepte des Faches, theoretische Grundlagen (AlP, TI, GTI, MafI, SWT, DB)
- Orientierungswissen ●●
  - "Wissenschaftl. Arbeiten", "Auswirkungen d. Informatik"
- Methodenwissen ●●
  - "Wissenschaftl. Arbeiten", "Softwaretechnik", Projekt, …
- Fertigkeiten ("skills") ●●
  - Seminare, Softwareprojekte, V**+Ü**
- Angewohnheiten ● (schlecht!)

Angebotsmenge in unserem Studiengang: ●●● gut, ●● OK, ● eher zu wenig

Inhalte wissenschaftliches Studium: 1. Fertigkeiten

Freie Universität Berlin

- Viele Leute finden, die Universitäten vermitteln in Informatik viel zu wenig praktische Fertigkeiten
  - Technologie-Elemente (Programmiersprachen, Bibliotheken, DBMS, Frameworks u.ä.)
  - Software-Werkzeuge
  - Projektmanagement, Teamarbeit
- Das hat aber seinen Grund:
  - es kostet sehr viel Zeit und
  - Grundwissen hat erst mal Vorrang

  denn:

Freie Universität Berlin

Lang-
lebig-
keit

2. Warum so viel Grundwissen? Beispiel: Lutz Prechelt

- Seine ersten drei Programmiersprachen:
  - AppleSoft
  - Pascal
  - Elan
- Alle überholt!

- Seine ersten drei Betriebssysteme:
  - UCSD-P
  - EUMEL
  - MS-DOS
- Alle überholt!

- Das Theoriewissen aus den Semestern 1 bis 3 seines Studiums:
  - Alles noch gültig!
  - Große Teile davon auch nützlich (Denkstrukturen!)

3. Warum sind Angewohnheiten wichtig?
3. Warum gibt kommt das zu wenig dran?

Freie Universität Berlin

- Menschen fallen unter Stress leicht auf das zuerst erlernte Verhalten für so eine Situation zurück [1]
- Sehr heikel!
- Deshalb werden sicherheitskritische Aufgaben evtl. sorgfältig gedrillt:
  - z.B. die Handgriffe von Rettungskräften, Chirurg/inn/en, Schütz/inn/en, Maschinenbediener/inne/n, Pilot/inn/en u.ä.
- Warum nicht mehr davon im Studium?
  - Sehr zeitintensiv
  - Sehr betreuungsintensiv
  - Oft ist zuvor Wissen verlangt

[1] Barthol, Ku: "Regression under stress to first learned behavior", Journal of Abnormal and Social Psychology 59(1):134-136, 1959

3. Wichtige Angewohnheiten für Informatiker/innen

Freie Universität Berlin

	Wo wird das erwähnt?
• Zuerst das Problem und das Ziel verstehen	V P A
• Dabei die Begriffe klären	V S A W
• Dann in Teile zerlegen	V P A
• Gütekriterien identifizieren/entwickeln	V A
• Prioritäten setzen (Kosten/Nutzen-Relation)	P A
• Mehrere Lösungsansätze in Erwägung ziehen	V P A
• Dabei Rad nicht neu erfinden	V P S A W ←
• Aufwände abschätzen	V P A
• Vorgehen begründen	V P A
• Lösungselemente begründen	P S A
• Lösung bewerten	P S A
• Reflektieren	V P S A W

V: div. Vorlesungen und deren Übungen, **P**: Softwareprojekt, **S**: Seminar, **A**: Abschlussarbeit, **W**: hier!

4. Methodenwissen Technologie-/Literatursuche: Beispiele (von Pepper)

Freie Universität Berlin

- Pepper hat 13 Werkzeuge für die Analyse von Repositories evaluiert
- 7 von Wissenschaftlern
  6 von Praktikern
  - ausgewählt: CVSAnalY

Name	Paper	Url (Abruf: 30.06.2011)
APFEL	[21]	http://www.st.cs.uni-saarland.de/softevo/apfel
CVSANALY	[17], [18]	http://tools.libresoft.es/cvsanaly
EVOLIZER	[10]	http://evolizer.org
FIXCACHE	[12], [18]	http://github.com/SoftwareIntrospectionLab/FixCache
GIT STATISTICS		http://repo.or.cz/w/git-stats.git
GITANALY		http://bitbucket.org/spaetz/gitanaly
GITSTATS		http://gitstats.sourceforge.net
GIT_MINING_TOOLS	[4]	http://github.com/cabird/git_mining_tools
GIT_STATS		http://gist.github.com/393324
HATARI	[19]	
KENYON	[3]	http://grase.soe.ucsc.edu/kenyon
LOOKATGIT		http://github.com/mpdehaan/lookatgit
METIOR		http://koraktor.de/metior

Tab. 2.1: Evaluierte Analysetools

4. Methodenwissen Literatursuche: Beispiele (von Pepper)

Freie Universität Berlin

- Die wissenschaftlichen Quellen (Artikel) zu CVSAnalY sind:
  - [17] Gregorio Robles, Stefan Koch, Jesús M. González-Barahona: *Remote Analysis and Measurement of Libre Software Systems by Means of the CVSAnalY Tool*. Proceedings of the 2nd ICSE Workshop on Remote Analysis and Measurement of Software Systems (RAMSS), IEEE CS, 2004, S. 51–55
  - [18] Caitlin Sadowski, Chris Lewis, Zhongpeng Lin, Xiaoyan Zhu, James Whitehead Jr.: *An Empirical Analysis of the FixCache Algorithm*. Proceedings of the 8th Working Conference on Mining Software Repositories (MSR). ACM, 2011, S. 219–222

Gregorio Robles

Caitlin Sadowski

Wie findet man sowas?

4. Methodenwissen: Suche nach Wissen

Freie Universität Berlin

Wichtigste Arten von Quellen:

- allgemeines Web
  - sehr breit, unterschiedlich tief
    - f. Technologie gut
    - f. Wissenschaft???
  - sehr schwankende Qualität
  - schwach strukturiert: Suche entweder sehr leicht oder sehr mühsam
    - im Wiederholungsfall oft OK
- Wissenschaftl. Lehrbücher
  - Sehr organisiert, gewichtet
  - Für Grundlagenwissen super
- Wissenschaftliche Artikel
  - Ohne Terminologiekenntnis schwierig zu finden
  - Ohne Grundlagenwissen schwierig zu verstehen
  - hoch spezialisiert

4. Suche nach wissenschaftlicher Literatur: Methode 1: Suchmaschinen

Freie Universität Berlin

- Wie?
  - Google Scholar
    - Volltextsuche
    - alle Fächer
    - umfassend, schnell
  - ACM Guide to the Computing Literature
    - Suche in Metadaten+Abstract
    - Informatik
  - DBLP
    - nur Metadatensuche
    - Informatik
    - übersichtlich, strukturiert
- Vorteil:
  - sehr schnell, wenn man gute Suchbegriffe hat
  - erwischt auch Artikel an exotischen Orten
- Nachteil:
  - fast hoffnungslos, wenn man keine guten Suchbegriffe hat
- Suche Thema v. PrePep14:
  - DICA
  - DICA defect insertion
  - defect insertion

4. Suche nach wissenschaftlicher Literatur: Methode 2: Einschlägige Orte

Freie Universität Berlin

- Oft gibt es eine zum Interessenbereich gut passende Konferenz oder Zeitschrift
- Dann kann man die Suche in deren jüngeren Inhalten beginnen

- Suche Thema v. PrePep14:
  - Konferenz "Mining Software Repositories (MSR)"
    - http://msrconf.org
  - ACM → DL → Proceedings → letzter Jahrgang von MSR → Table of contents

4. Suche nach wissenschaftlicher Literatur: Methode 3: Experten fragen

Freie Universität Berlin

- Und wenn man keine passenden Suchbegriffe, Konferenzen oder Zeitschriften kennt?
- Jemand fragen, der sich in der ungefähren Gegend auskennt:
  - Welche Suchbegriffe könnten mich weiter bringen?
  - Wo würdest Du suchen?
  - Welche Autor/inn/en würdest Du sichten?
- Heute:
  - Profs, WiMis
- als Doktorand/in:
  - Autor/inn/en der besten bekannten verwandten Arbeiten
- als SW-Ingenieur/in:
  - beschlagene Kolleg/inn/en
  - Besucher/innen passender Treffen, z.B.
    - meetup.com
    - conference-service.com
    - gi.de
  - oder Foren, z.B.
    - stackexchange.com

4. Suche nach wissenschaftlicher Literatur: Methode 4: Zitationen in die Vergangenheit

Freie Universität Berlin

- Wenn man erste ungefähr passende Artikel gefunden hat, deren Literaturliste und Abschnitt "verwandte Arbeiten" durchsehen.
  - Das ist zwar alles älter
  - aber es gibt dort trotzdem regelmäßig viel zu lernen

4. Suche nach wissenschaftlicher Literatur: Methode 5: Zitationen in die Zukunft

Freie Universität Berlin

- Google Scholar erlaubt auch, Zitationen aus der Zukunft (des jeweiligen Artikels) zu finden:

Why software repositories are not used for defect-insertion circumstance analysis more often: A case study fu-berlin.de [PDF] E-Ressource

L Prechelt, A Pepper - Information and Software Technology, 2014 - Elsevier

Method We use a single-case, typical-case, revelatory-type case study to evaluate in parallel the importance of six plausible reasons (R1–R6). The case is based on 11 years of repository data from a small but mature software company building a product in the high- ...

Zitiert von: 2 Ähnliche Artikel Alle 4 Versionen Zitieren Speichern

Sehr nützlich!

4. Suche nach wissenschaftlicher Literatur: Methode 6: Google sieht alles

Freie Universität Berlin

- Google Scholar bietet außerdem Artikel an, die textuell ähnlich sind
  - gleiche Kernbegriffe
  - gleiche Zitationen
- Das ist einen Versuch wert, wenn man alles andere schon ausgelotet hat
  - Scheint zwar bequem, kann einem aber das Lesen nicht abnehmen…

Why software repositories are not used for defect-insertion circumstance analysis more often: A case study fu-berlin.de **[PDF]** E-Ressource

L Prechelt, A Pepper - Information and Software Technology, 2014 - Elsevier

Method We use a single-case, typical-case, revelatory-type case study to evaluate in parallel the importance of six plausible reasons (R1–R6). The case is based on 11 years of repository data from a small but mature software company building a product in the high- **...**

Zitiert von: 2 Ähnliche Artikel Alle 4 Versionen Zitieren Speichern

Zurück zu letzter Woche: Was tun (Software)Ingenieur/inn/e/n?

Freie Universität Berlin

- https://en.wikipedia.org/wiki/Engineer#Design
  - Informationen suchen und verarbeiten
  - Informationen weitergeben
  - Probleme definieren
  - Kriterien entwickeln
  - Lösungsvorschläge entwickeln, bewerten, abwägen, auswählen
  - Lösungen umsetzen und verbessern
  - Lösungen bewerten
- All dies
  - systematisch,
  - auf Basis von tiefem Wissen
  - und ohne das Rad neu zu erfinden
- Dies benötigt wissenschaftliches Arbeiten

  denn das heißt ja bekanntlich:

Definition "wissenschaftliche Tätigkeit"

- "Wissenschaftliche Tätigkeit [ist] alles, was nach Inhalt und Form als ernsthafter planmäßiger Versuch zur Ermittlung der Wahrheit anzusehen ist"
  - Bundesverfassungsgericht

- Und die gesuchte "Wahrheit" im Ingenieurbereich ist:

Wie löst man dieses Problem verlässlich und effizient?

Also nochmal: Was heißt nun "wissenschaftliches Studium"? (in Informatik)

- Ziel: Systematisches Suchen nach Nützlichkeit (für Informatiksysteme) erlernen
- Dazu gehört: Grundwissen, Orientierungswissen, Fertigkeiten Methodenwissen, Angewohnheiten
  - insbesondere: Wissenschaftlich arbeiten können und es auch tun

Wissenschaftlich arbeiten zu können heißt ungefähr so viel wie:

Ingenieurprobleme lösen zu können

und wissenschaftliche Forschung ist lediglich ein weiteres mögliches Einsatzgebiet.

Wissenschaftliches Studium befähigt zu beidem.

Und jetzt nochmal: Umfrage

Welches ist eher Ihre Haltung?

- Haltung 1:
  "Wissenschaftliches Arbeiten?
  Klasse Sache!
  Das will ich auch lernen."

- Haltung 2:
  "Das interessiert mich eher wenig.
  Wissenschaft hat mit meinen Zielen kaum was zu tun."

Hausaufgabe

- Wenn noch nicht geschehen: Lesen Sie den Artikel PrePep14

  Lutz Prechelt, Alexander Pepper:
  Why Software Repositories Are Not Used For Defect-Insertion Circumstance Analysis More Often: A Case Study.
  Information and Software Technology 56(10):1377-1389, October 2014.

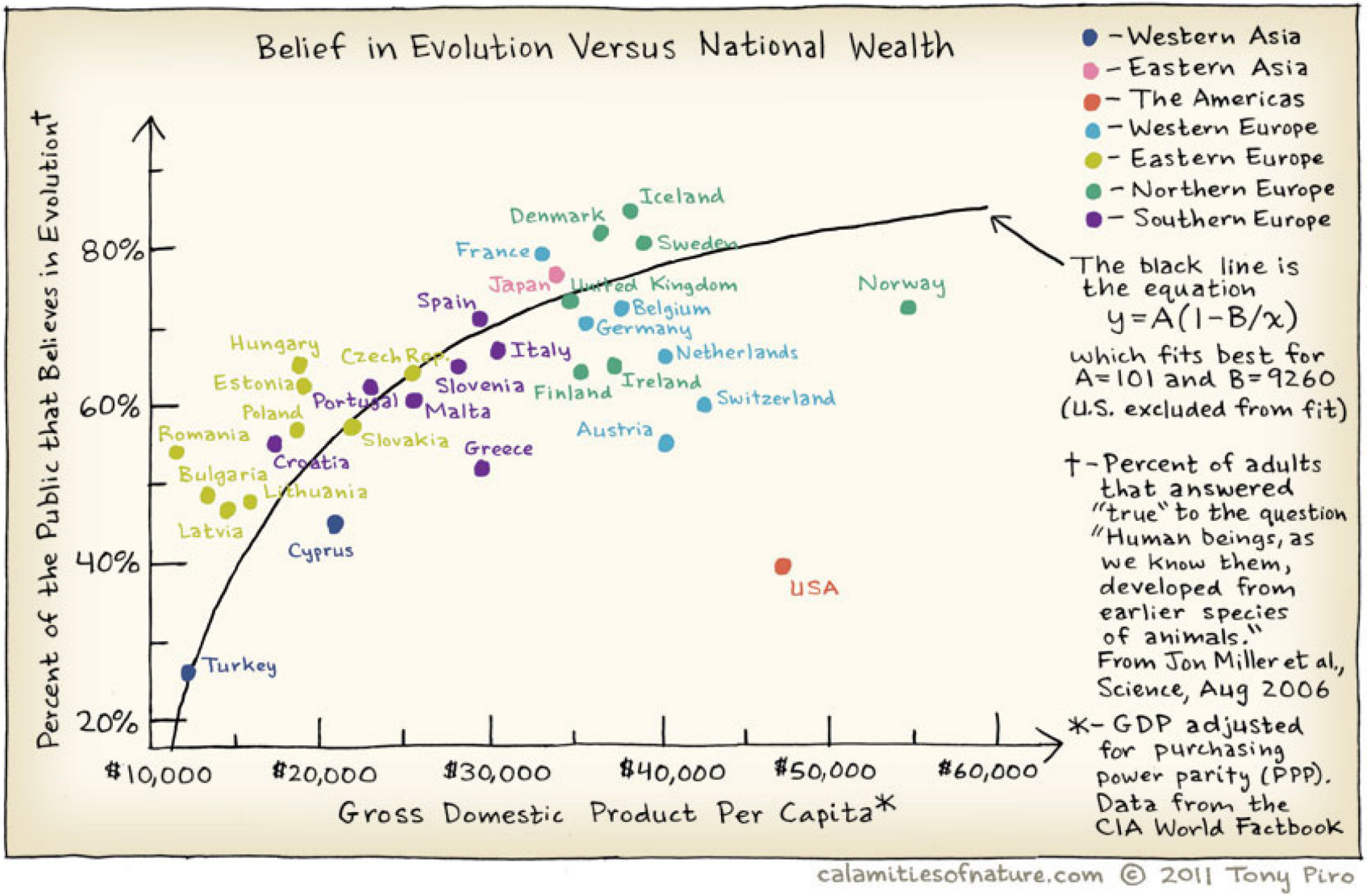
  - Abschnitte 1, 2, 3.2, 3.3, 3.4, 3.6, 6, 7.1, 7.2, 7.7, 7.8, 8, 9, 10 gründlich
  - Den Rest mindestens ungefähr

- Schauen Sie folgendes Video über die Natur wissenschaftlicher Theorien (9 Minuten):
  - https://www.youtube.com/watch?v=-X8Xfl0JdTQ
  - (Man kann in den Einstellungen das Sprechtempo reduzieren)

Danke!

Ups, Momentchen noch…

Nachbemerkung: Gesellschaftliche Aufgabe! (z.B. USA sind ein extrem rückständiges Land)



http://www.calamitiesofnature.com/archive/?c=559