

Media Engineering

Shit Happens

...aber nicht notwendigerweise



R. Weller

University of Bremen, Germany

cgvr.cs.uni-bremen.de

Murphy's law

- “Anything that can go wrong, *will* go wrong.”



Edward A. Murphy

Beispiel: CrowdStrike (2024)

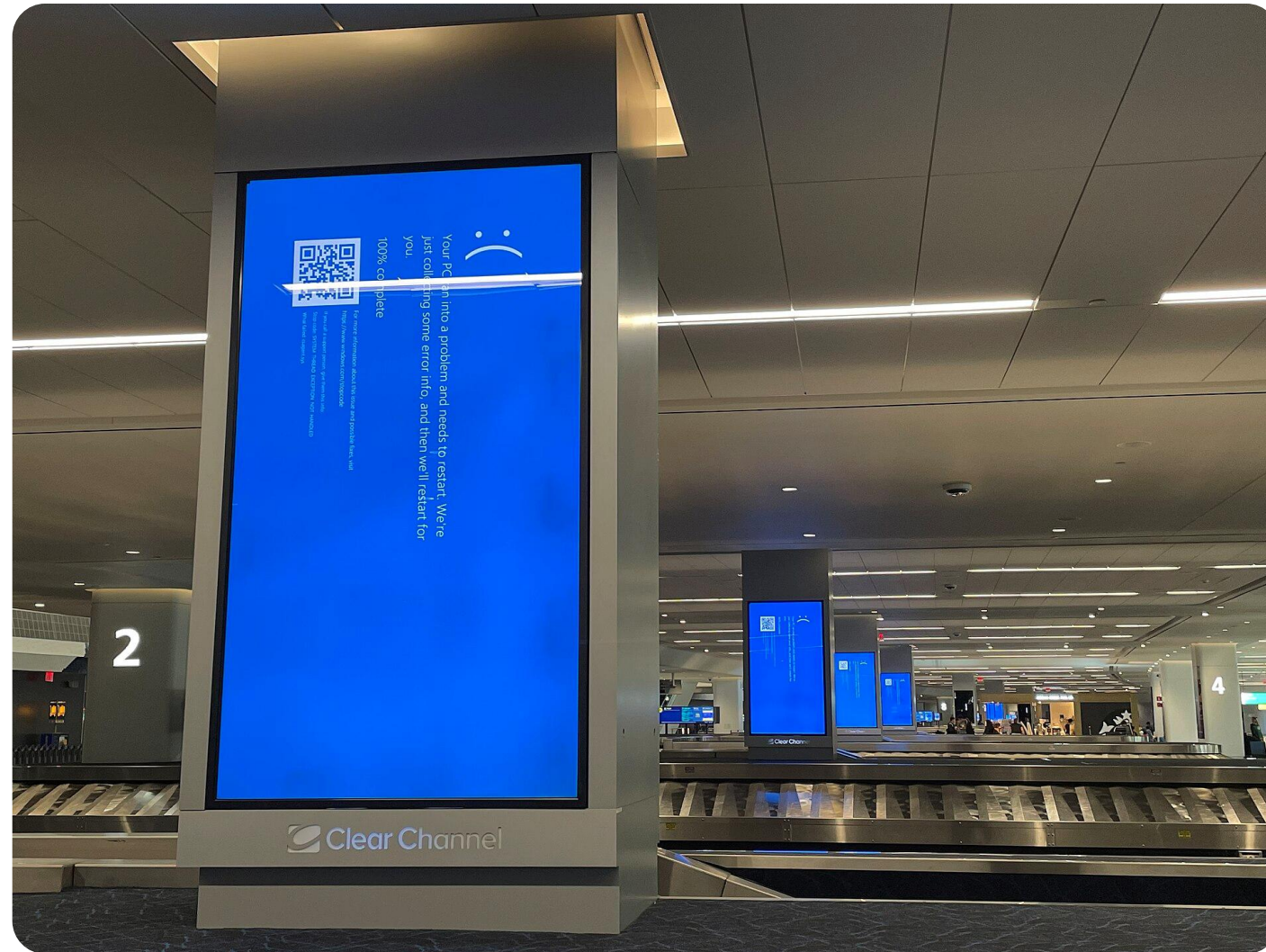


Foto: Wikimedia Commons („CrowdStrike BSOD at LGA“), Flughafen LaGuardia, 19. Juli 2024

Beispiel: Citibank (2020)

BDLL		Borrower LIBOR Drawdown Prod		Drawdown	
001BDLL201480094				001BDLL201480094	
024462		REVLON CONSUMER PRODUCTS CORP			
Facility Name		REVLON TERM LOAN 2016			

GL Detail		Internal GL	Overwrite default settlement instruction
Component			
COLLAT			☐
COMPINTSF			☐
DEFAULT			☐
DFLFTC			☐
FRONT			☐
FUND			☐
INTEREST			☐
PRINCIPAL	3003000023		☒
			☐

Originalmaske aus dem Urteil In re Citibank August 11, 2020 Wire Transfers, U.S. District Court S.D.N.Y., 2021

Beispiel: Concord (2024)



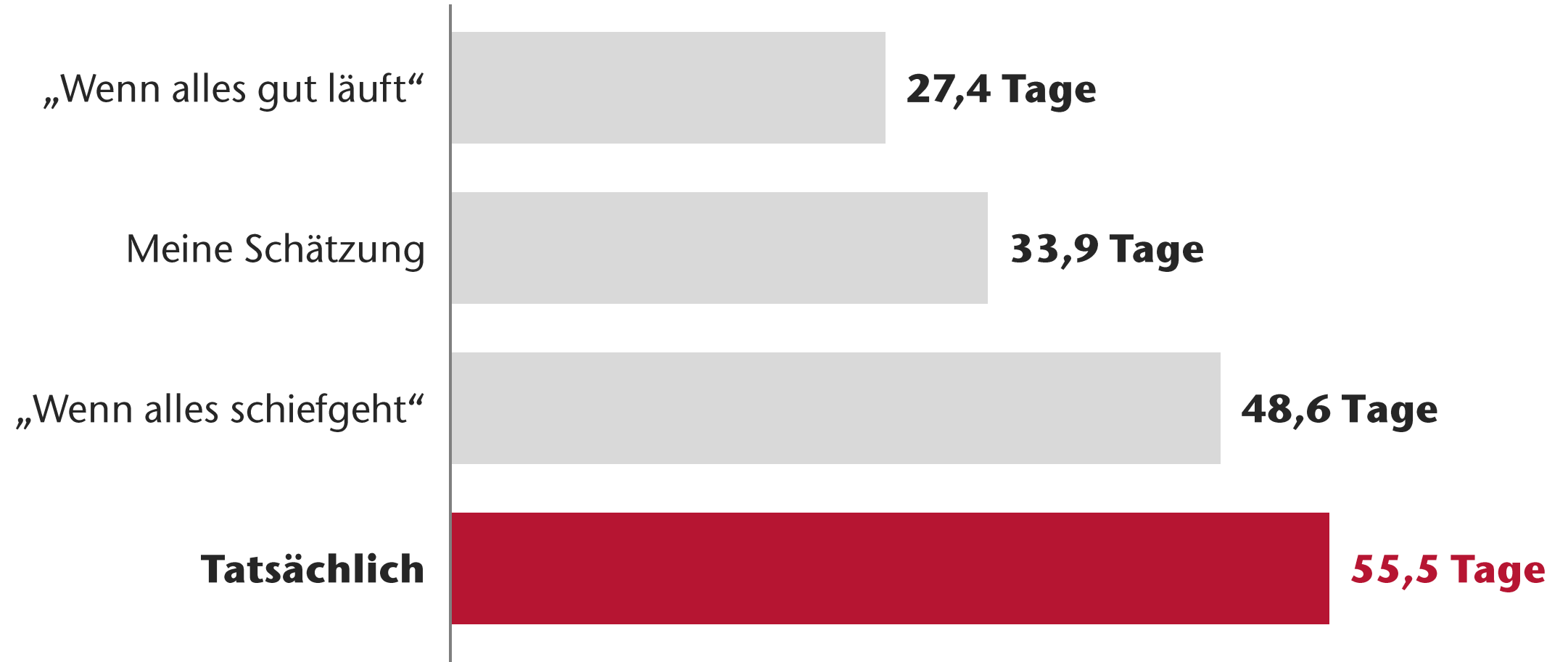
Bild: Sony Interactive Entertainment / Firewalk Studios

Beispiel: Star Citizen (seit 2012)



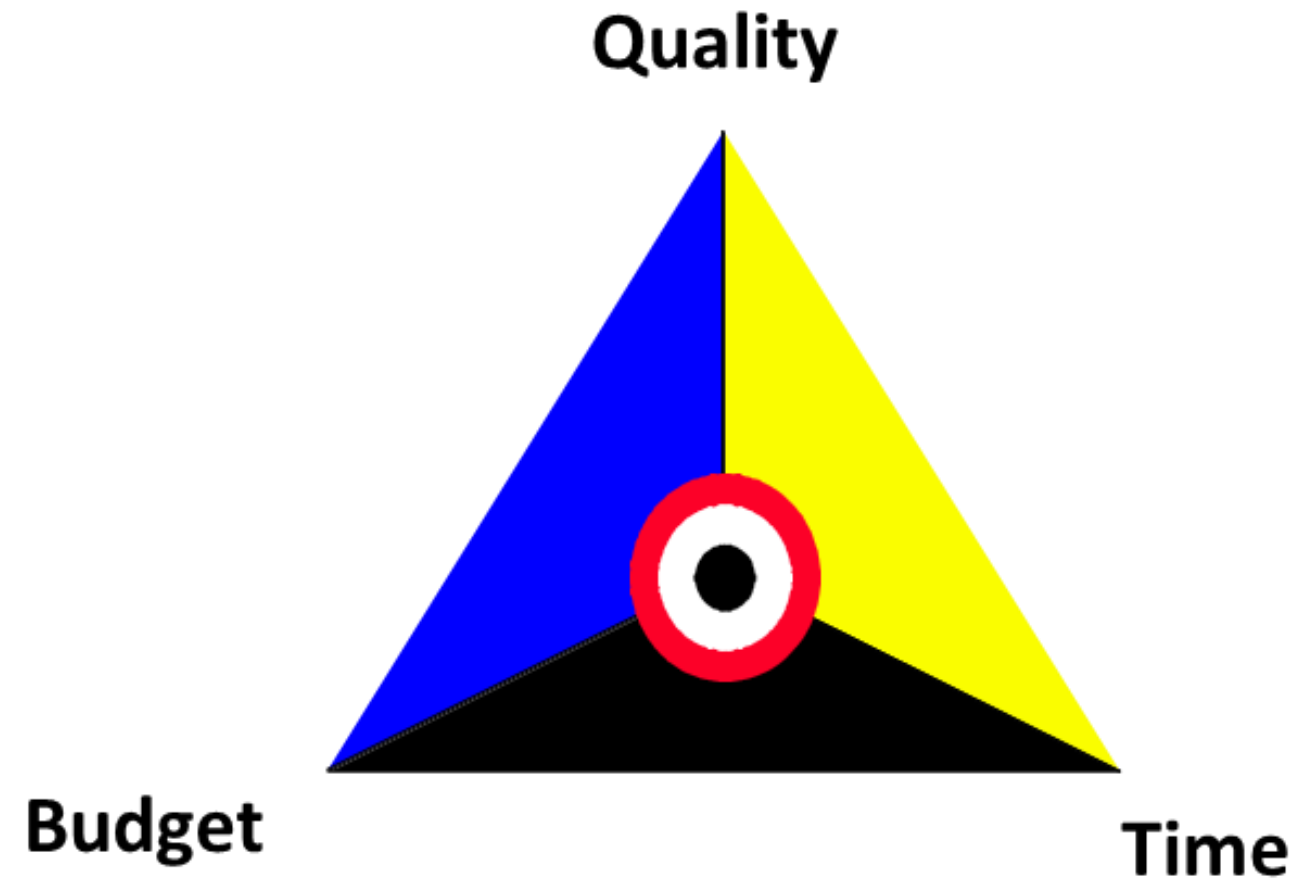
Bild: Cloud Imperium Games (Pressematerial)

Wann ist meine Abschlussarbeit fertig?



Buehler, Griffin & Ross: Exploring the “planning fallacy”. Journal of Personality and Social Psychology 67(3), 1994

Was bedeutet scheitern?



Hall of Shame (sehr unvollständige Auswahl)

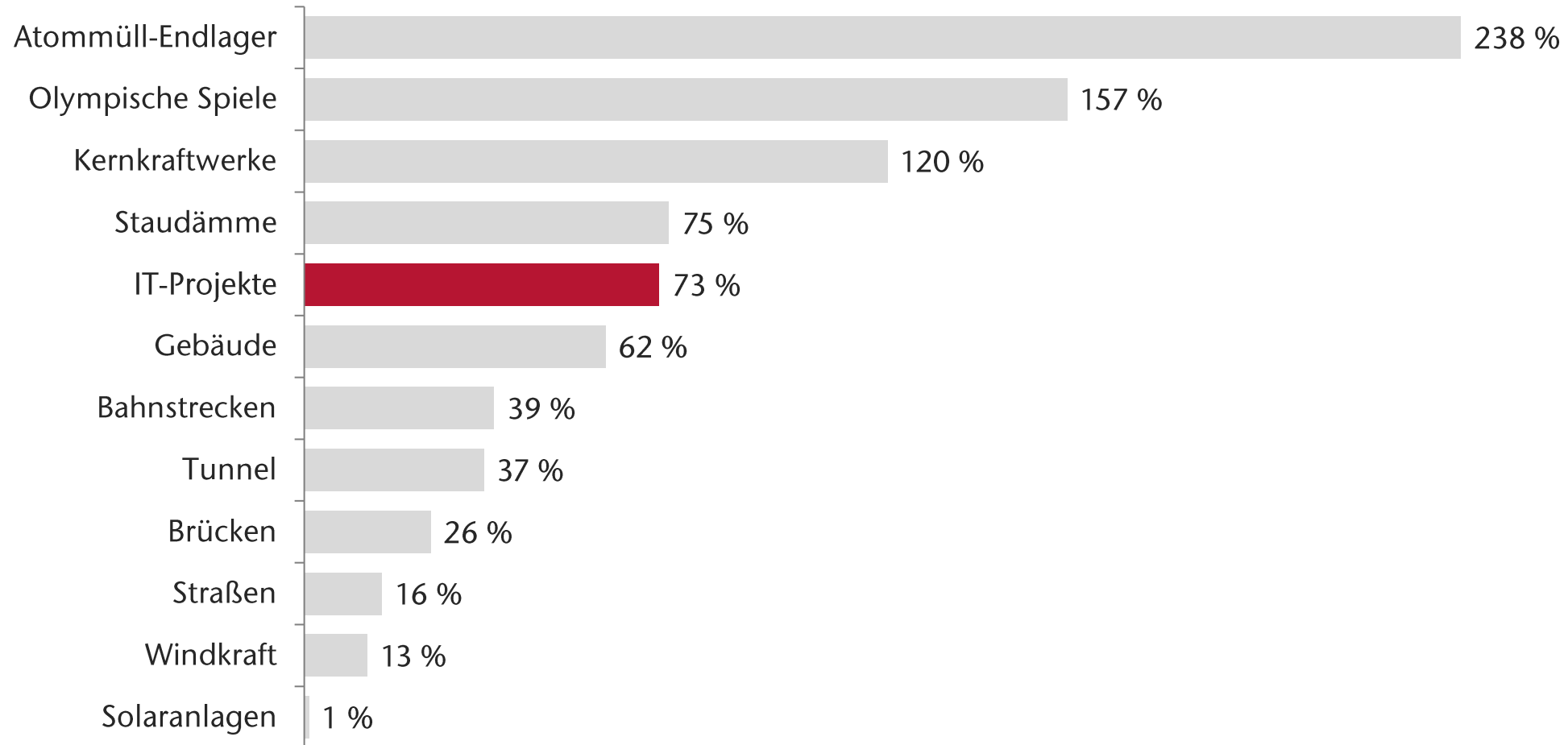
Projektname/Beteiligte	Dauer	Ende	Kosten
eLWIS Lidl und SAP	7 Jahre	2018	ca. 500 Mio. €
NPfIT (Patientenakte) NHS, Großbritannien	9 Jahre	2011	ca. 10 Mrd. £
Digital Media Initiative BBC	5 Jahre	2013	98 Mio. £
MyCalPays SAP und Kalifornien	9 Jahre	2013	254 Mio. \$
Oracle-Einführung Birmingham City Council	seit 2018	offen	19 → über 140 Mio. £
Anthem BioWare, Electronic Arts	7 Jahre	2019	nicht veröffentlicht
Insieme Eidg. Steuerverwaltung	7 Jahre	2012	ca. 116 Mio. Franken
Toll Collect Telekom, Daimler	+16 Mon.	2005	Vergleich: 3,2 Mrd. €

Scheitern nur IT-Projekte?



Scheitern IT-Projekte häufiger?

Mittlere Kostenüberschreitung gegenüber dem Plan



Flyvbjerg & Gardner: How Big Things Get Done, 2023 (Datenbank mit über 16 000 Projekten)

Was ist Software überhaupt?

- Software besteht aus:
 - Programm
 - Daten
 - Dokumentation

Was macht Software so speziell?

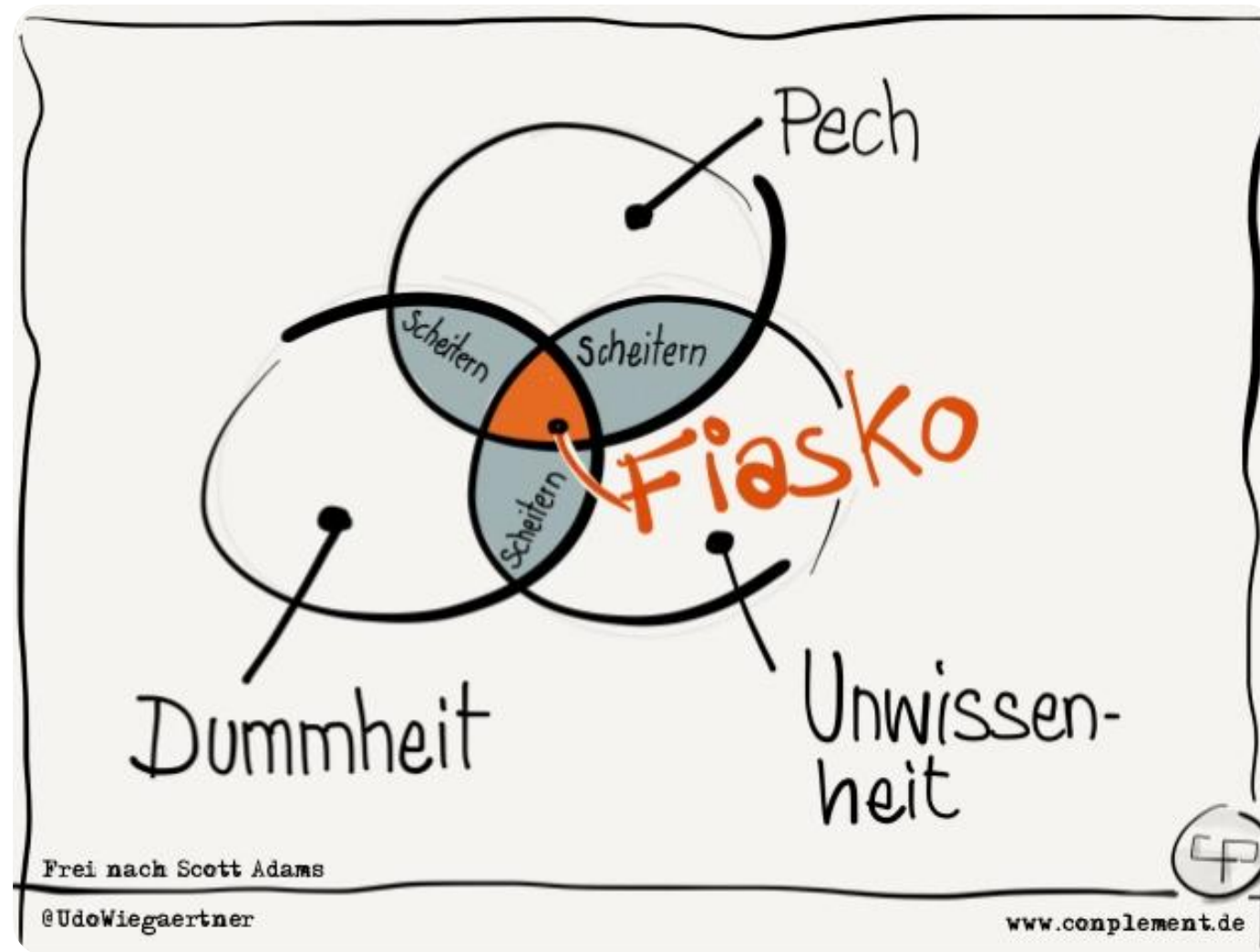


Foto: MIT Instrumentation Laboratory, 1969 (Wikimedia Commons)

Kostencharakteristik digitaler Güter

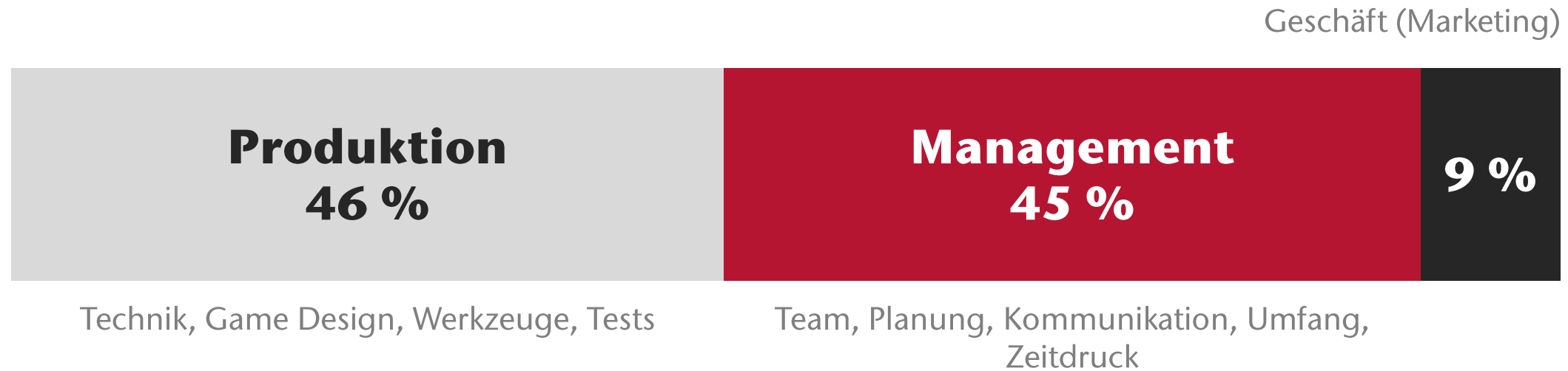


Gründe für das Scheitern von IT-Projekten



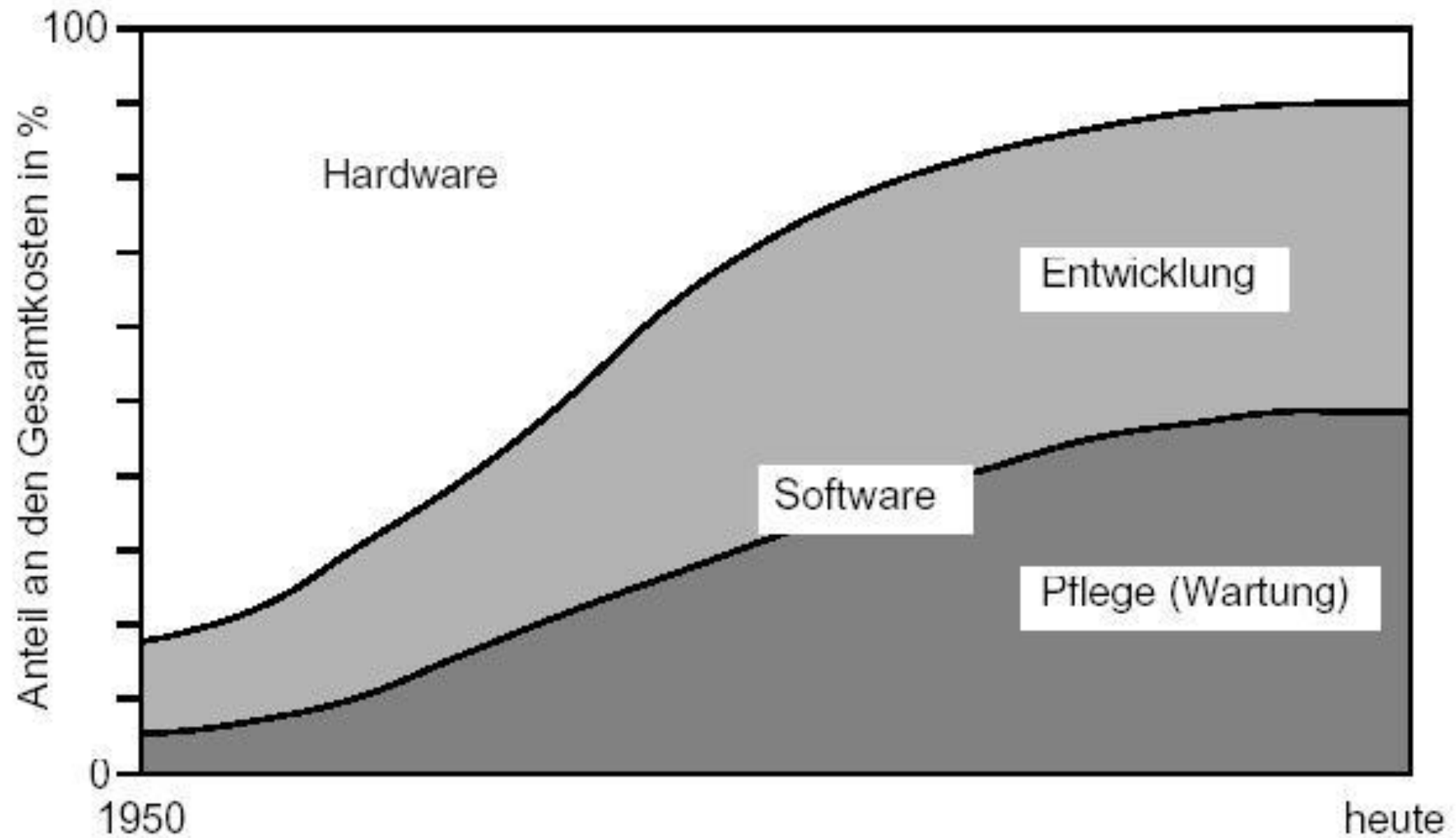
Was geht bei Spielen schief?

927 Probleme aus 200 Projektrückblicken (1997–2019)



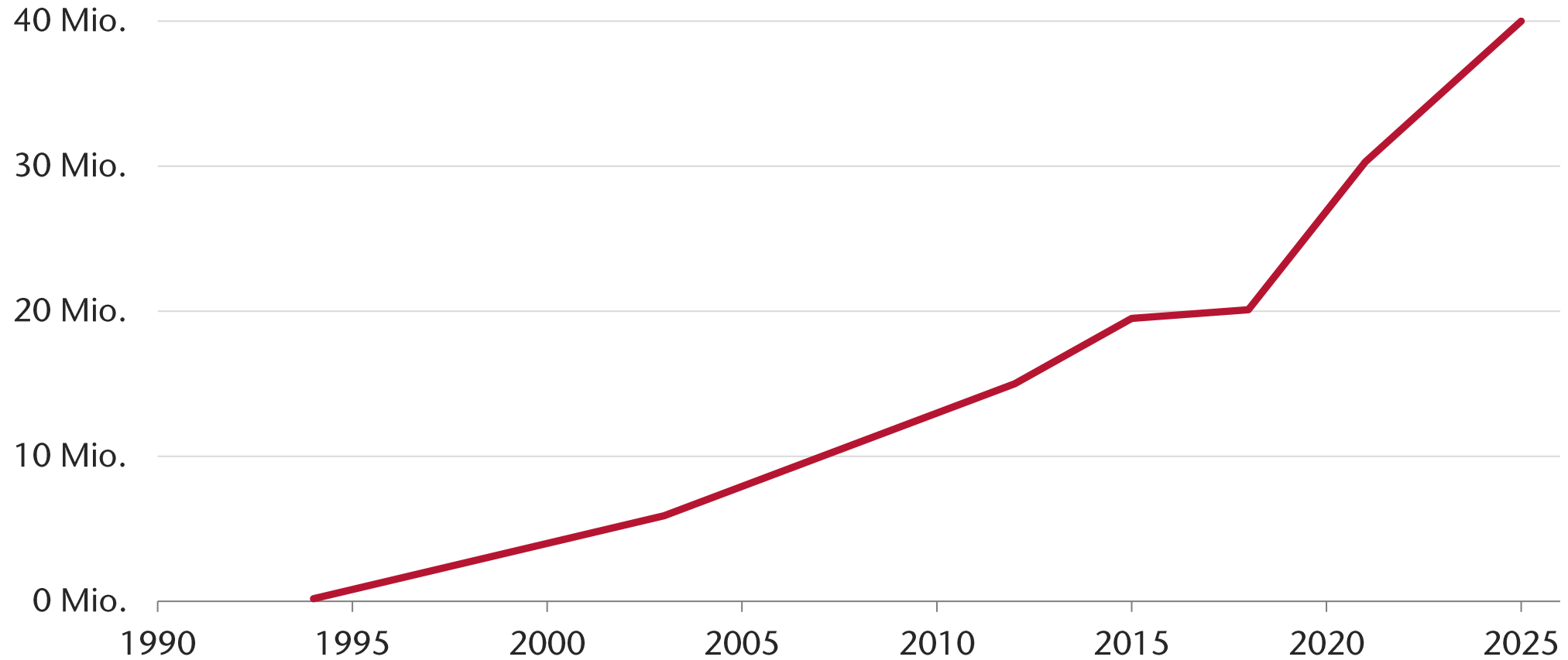
Politowski et al.: Game industry problems: An extensive analysis of the gray literature. Information and Software Technology 134, 2021

Historisches: Softwarekrise



Grund: Zunehmende Komplexität der Software

Linux-Kernel: Zeilen Quellcode



Zeilenzahlen der Kernel-Versionen 1.0, 2.6.0, 3.2, 4.1, 4.14, 5.11 und 6.14 (kernel.org; Zusammenstellung nach Wikipedia „Linux kernel“)

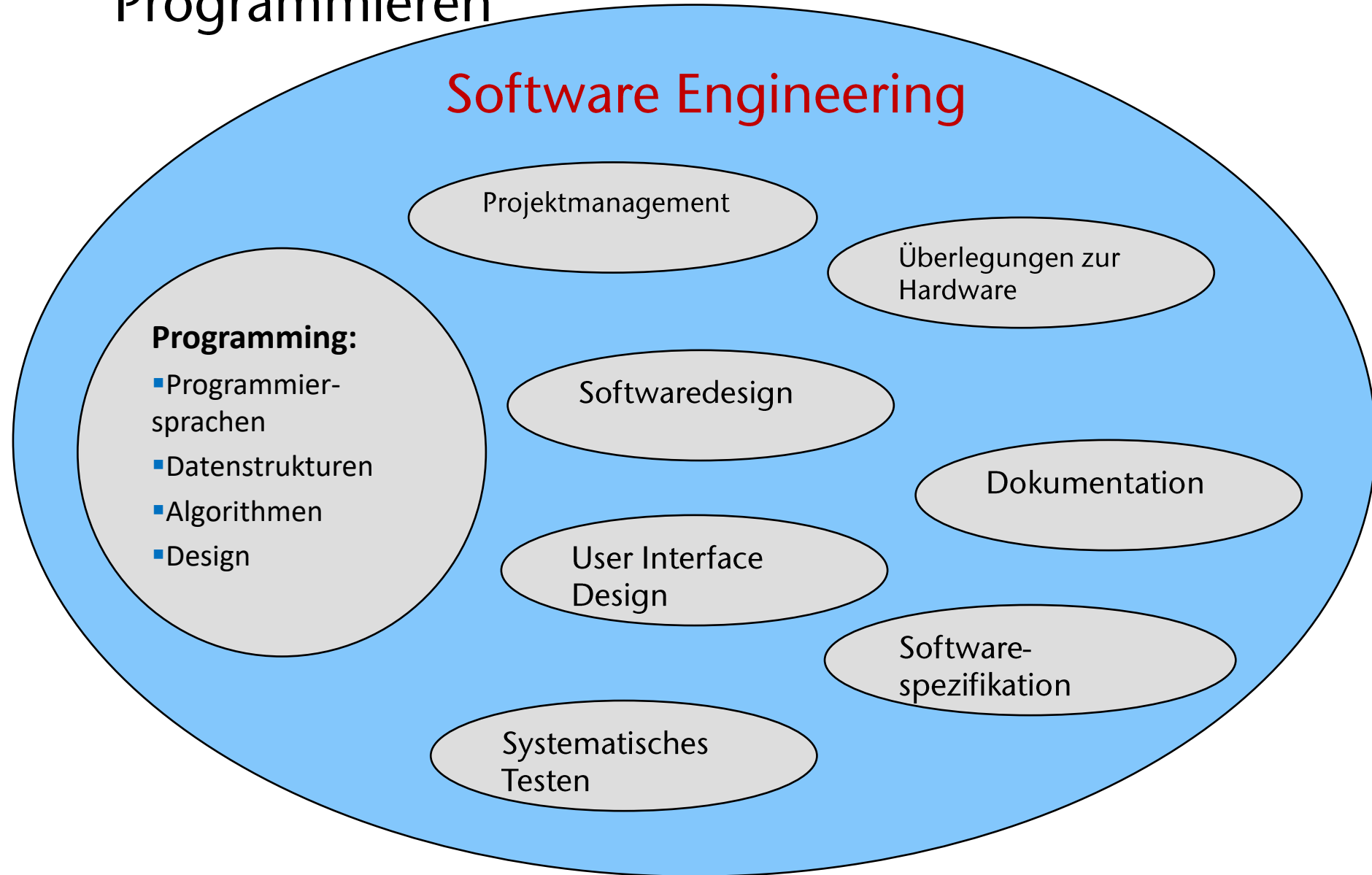
Und wie verhindert man jetzt das Scheitern?

- Die Lösung (?): Software Engineering

The application of a **systematic, disciplined, quantifiable** approach to **development, operation, and maintenance** of software; that is, the **application of engineering to software**.

(IEEE Standard Computer Dictionary, 1990)

Hinweis: Software-Engineering ist mehr als Programmieren



Womit verbringen Entwickler ihre Zeit?

Arbeitszeit von 78 Berufsentwicklern

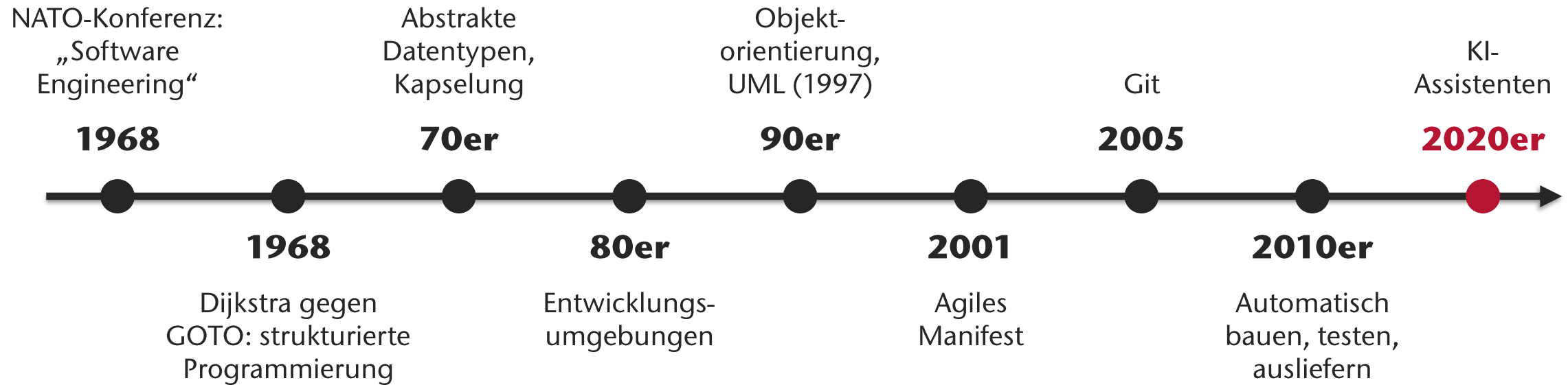
Code verstehen
58 %

alles andere
42 %

im Code navigieren, Code ändern, Sonstiges

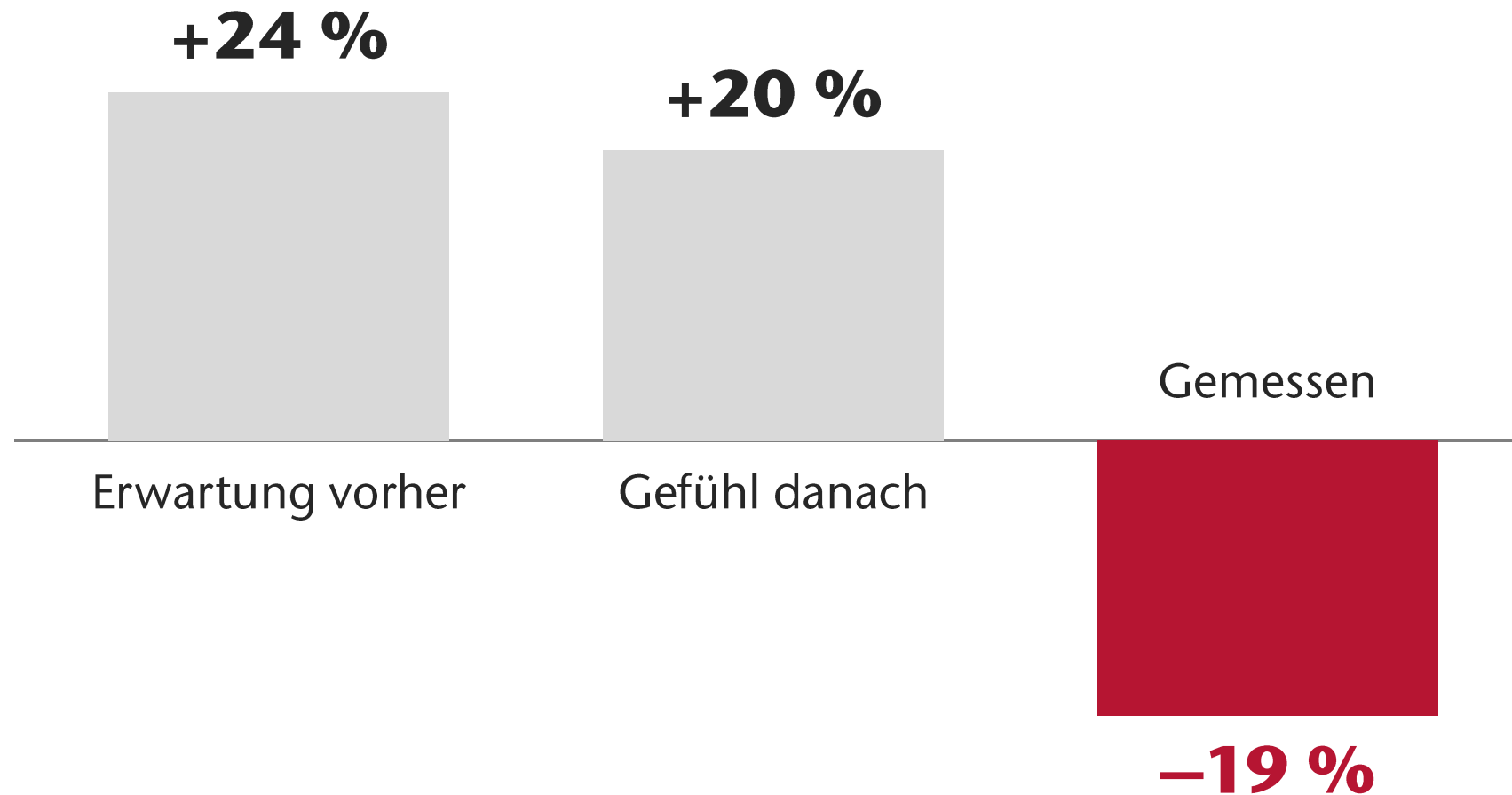
Xia et al.: Measuring Program Comprehension: A Large-Scale Field Study with Professionals. IEEE Trans. Software Eng. 44(10), 2018

Einige historische Meilensteine



Macht KI Entwickler schneller?

schneller (+) oder langsamer (–) mit KI-Werkzeugen



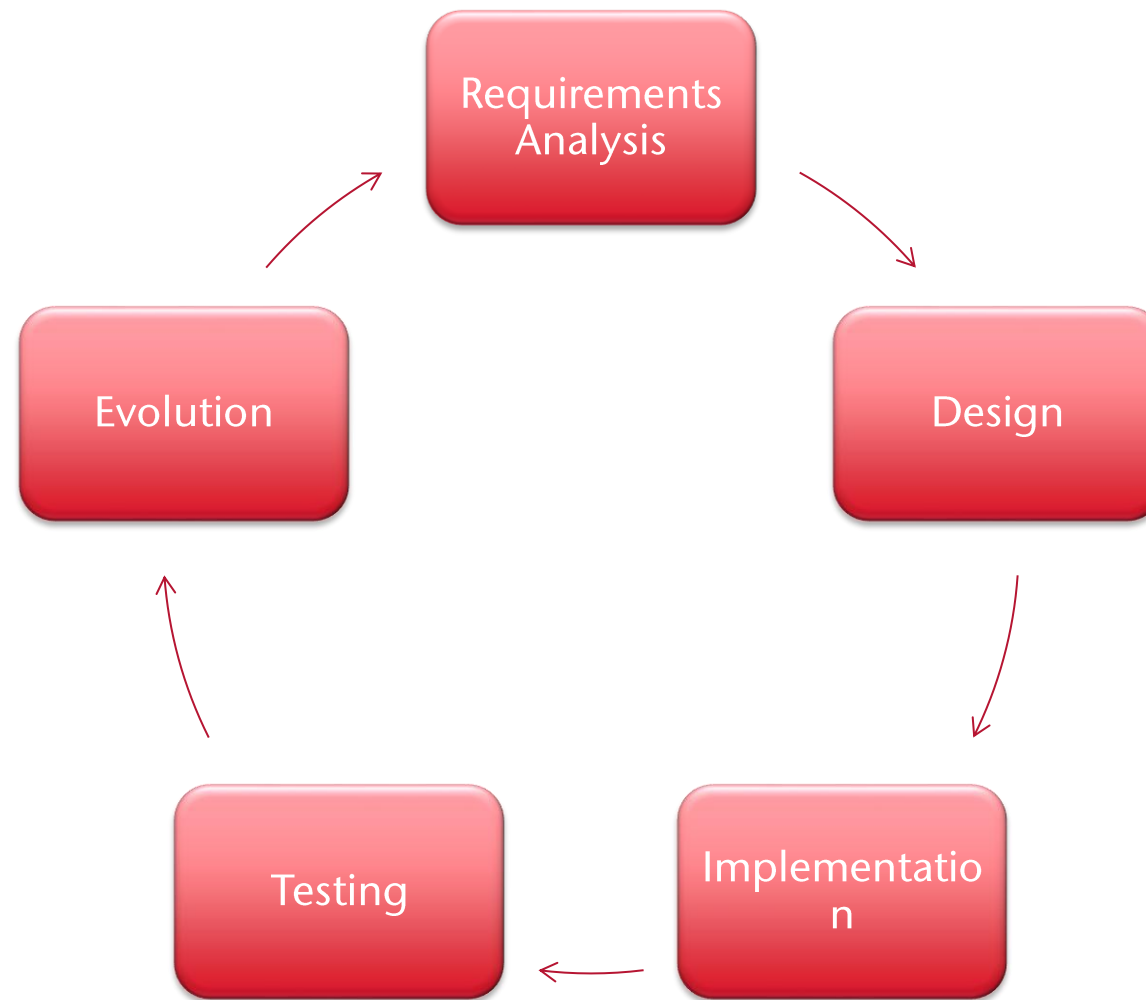
Becker et al. (METR): Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity, 2025

Die Anwendung der Prinzipien des Software Engineering auf die Entwicklung digitaler Medien.

Unterschiede zwischen Software und anderen digitalen Medien

Klassische Software	Digitale Medien
Oft für den Arbeitsplatz	Eher zur Unterhaltung
Vor allem Code	Code plus Inhalte: Grafik, Sound, Story, Leveldesign
Oft klar umrissene Benutzergruppe	Unvorhersehbar – ein Hit kann über Nacht aus 100 Benutzern 100 000 machen
Anforderungen lassen sich erfragen	„Spaß“ lässt sich nicht erfragen – Prototypen und Playtests
Meist Maus, Tastatur, Touch	Viele Eingabegeräte: Controller, Touch, Sprache, Gesten, VR/AR
Design follows function	Function follows design

Der Software Development-Lifecycle



Berufsentwickler verbringen gut die Hälfte ihrer Arbeitszeit damit, vorhandenen Code zu verstehen.

?

Welche Folgerung ist am besten begründet? Eine Antwort ist richtig.

A

Wer schneller tippt, ist deutlich produktiver.

B

Lesbarer Code spart später mehr Zeit, als das sorgfältige Schreiben kostet.

C

Kommentare sind überflüssig, weil der Code ohnehin gelesen wird.

D

Mit KI-Werkzeugen muss man vorhandenen Code nicht mehr verstehen.

1 Eine Minute allein nachdenken

2 Stichworte notieren

3 Antworten sammeln, dann Auflösung

„Wenn ein Team jede Aufgabe einzeln durchdenkt und überall den schlimmsten Fall ansetzt, liegt die Gesamtschätzung auf der sicheren Seite.“

?

Wahr oder falsch? Begründen Sie in einem Satz.

1 Eine Minute allein nachdenken

2 Stichworte notieren

3 Antworten sammeln, dann Auflösung

Erfahrene Entwickler schätzten nach der Arbeit mit KI-Werkzeugen, sie seien 20 % schneller gewesen. Gemessen waren sie 19 % langsamer.

?

Welche Forderung aus der IEEE-Definition von Software Engineering zeigt dieses Ergebnis – und was heißt das für Ihr Projekt?

1 Eine Minute allein nachdenken

2 Stichworte notieren

3 Antworten sammeln, dann Auflösung

Lessons Learned



Software-Projekte scheitern selten an der Technik, sondern meist an Planung, Anforderungen und Kommunikation.



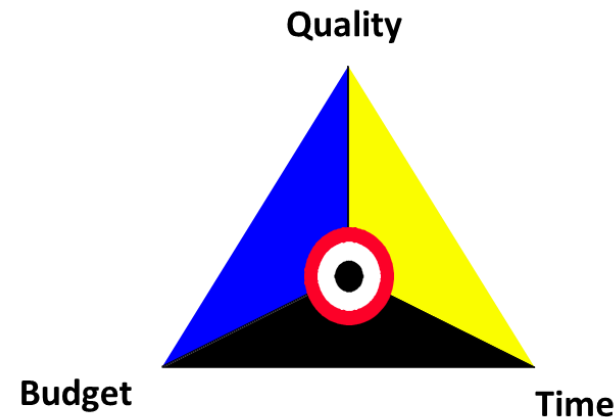
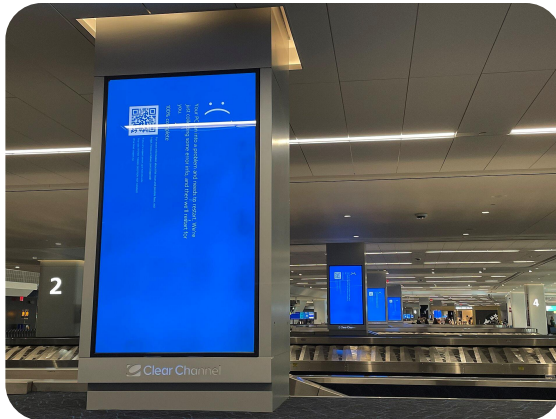
Software ist ein immaterielles Einzelstück. Deshalb schätzen wir zu optimistisch – Erfahrungswerte schlagen das Bauchgefühl.



Software Engineering heißt: systematisch, diszipliniert und messbar vorgehen. Bei Spielen kommt hinzu, dass sich Spaß nur durch Ausprobieren prüfen lässt.

- Zur Erinnerung

Task	Stunden
...	...
Nachbearbeiten der Vorlesung	28
...	...



Edward A. Murphy

“Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the universe trying to produce bigger and better idiots. So far, the universe is winning” - **Rick Cook**