Softwareproblem.
Tut mir leid, da kann man nix machen.
(Spoiler: man kann)

Lange Nacht der Universitäten, 29./30. Mai 2026

Gidon Ernst
Vertretung Softwartechnik, Universität Augsburg
Juniorprofessor Softwareverifikation, LMU Munich
https://www.sosy-lab.org/people/ernst/

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

uniA

Disclaimer

some content in this presentation
has been altered* to fit the narrative

*but only a little bit

9/9

0800 Arctan started { 1.2700 9.037 847 025
1000 " stopped - arctan ✓ 9.037 846 995 correct
13 Dec (032) MP - MC ~~1.5821447000~~ ~~2.130476415 (-3)~~ 4.615925059 (-2)
(033) PRO 2 2.130476415
correct 2.130676415

Relays 6-2 in 033 failed special speed test
In relay " 11.00 test.
Relays changed

Relay 2145
Relay 3370

1100 Started Cosine Tape (Sine check)
1525 Started Mult + Adder Test.

1545 

Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

~~1545~~ 1630 Arctangent started.
1700 closed down.

9/9

0800 Arctan started
1000 " stopped - arctan ✓ { 1.2700 9.037 847 025
9.037 846 795 correct
13 sc (032) MP - MC ~~1.4821447000~~ ~~2.1304776415 (-3)~~ 4.615925059(-2)
(033) PRO 2 2.130476415
correct 2.130676415
Relays 6-2 in 033 failed special speed test
In relay " 11.00 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Multy Adder Test.

1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
~~1700~~ 1630 Arctangent started.
1700 closed down.

Relay 2145
Relay 3370

"first computer bug"


Gerd Eist
@erdgeist
Computer-Fehler, kann man nichts machen. Außer das kaputte Datum mit Edding auf dem Display zu korrigieren 😂
Translate Tweet
13 Bakenberg Strand
Kuhle
Dranske Lancken
Dranske
Friedhof
Wittower Straße
H Dranske Lancken
11.04.2012 Bus stop

Gerd Eist
@erdgeist

Computer-Fehler, kann man nichts machen. Außer das kaputte Datum mit Edding auf dem Display zu korrigieren 😂

Translate Tweet


Gerd Eist
@erdgeist
Computer-Fehler, kann man nichts m das
kaputte Datum mit Edding auf de
korrigieren 😂
Translate Tweet
pro-coder 3000
now with 100% vibe
11.04.2022
Bus stop
H
Dranske Lancken
11.04.2022
Bus stop

Windows 11 ist ein kompletter Verkehrsunfall

Windows 11 hat den Ruf, ziemlich fehlerhaft zu sein. Wir wollten wissen: Was genau ist aktuell kaputt?

22.05.2026, 18:16 Uhr Lesezeit: 23 Min. | c't Magazin

Beyond 12,500 former Nokia employees, who else is Microsoft laying off ? (July 17, 2014)

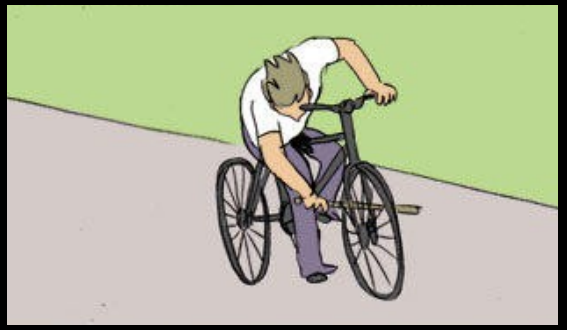
https://www.zdnet.com/article/beyond-12500-former-nokia-employees-who-else-is-microsoft-laying-off

[...] a number of Windows engineers,
primarily **dedicated testers**, will no longer be needed

Beyond 12,500 former Nokia employees, who else is Microsoft laying off ? (July 17, 2014)

https://www.zdnet.com/article/beyond-12500-former-nokia-employees-who-else-is-microsoft-laying-off

[...] a number of Windows engineers,
primarily **dedicated testers**, will no longer be needed



Beispiel: Ein Programm

https://github.com/torvalds/linux/blob/master/crypto/rsa.c#L51

```
/*
 * RSAEP function [RFC3447 sec 5.1.1]
 * c = m^e mod n;
 */
static int _rsa_enc(const struct rsa_mpi_key *key, MPI c, MPI m)
{
    /*
     * Even though (1) in RFC3447 only requires 0 ⩽ m ⩽ n - 1,
     * we are slightly more conservative and require 1 < m < n - 1.
     * This is in line with SP 800-56Br2, Section 7.1.1.
     */
    if (rsa_check_payload(m, key→n))
        return -EINVAL;

    /* (2) c = m^e mod n */
    return mpi_powm(c, m, key→e, key→n);
}
```

Quiz

Wie viele Fehler sind typischerweise in 1000 Zeilen Code?

Quiz

Wie viele Fehler sind typischerweise in 1000 Zeilen Code?

“mehrere” (± 5)

https://www.heise.de/security

Quiz

Wie viele Fehler sind typischerweise in 1000 Zeilen Code?

“mehrere” (± 5)

https://www.heise.de/security

Agenda

- Fallstudie: Crowdstrike

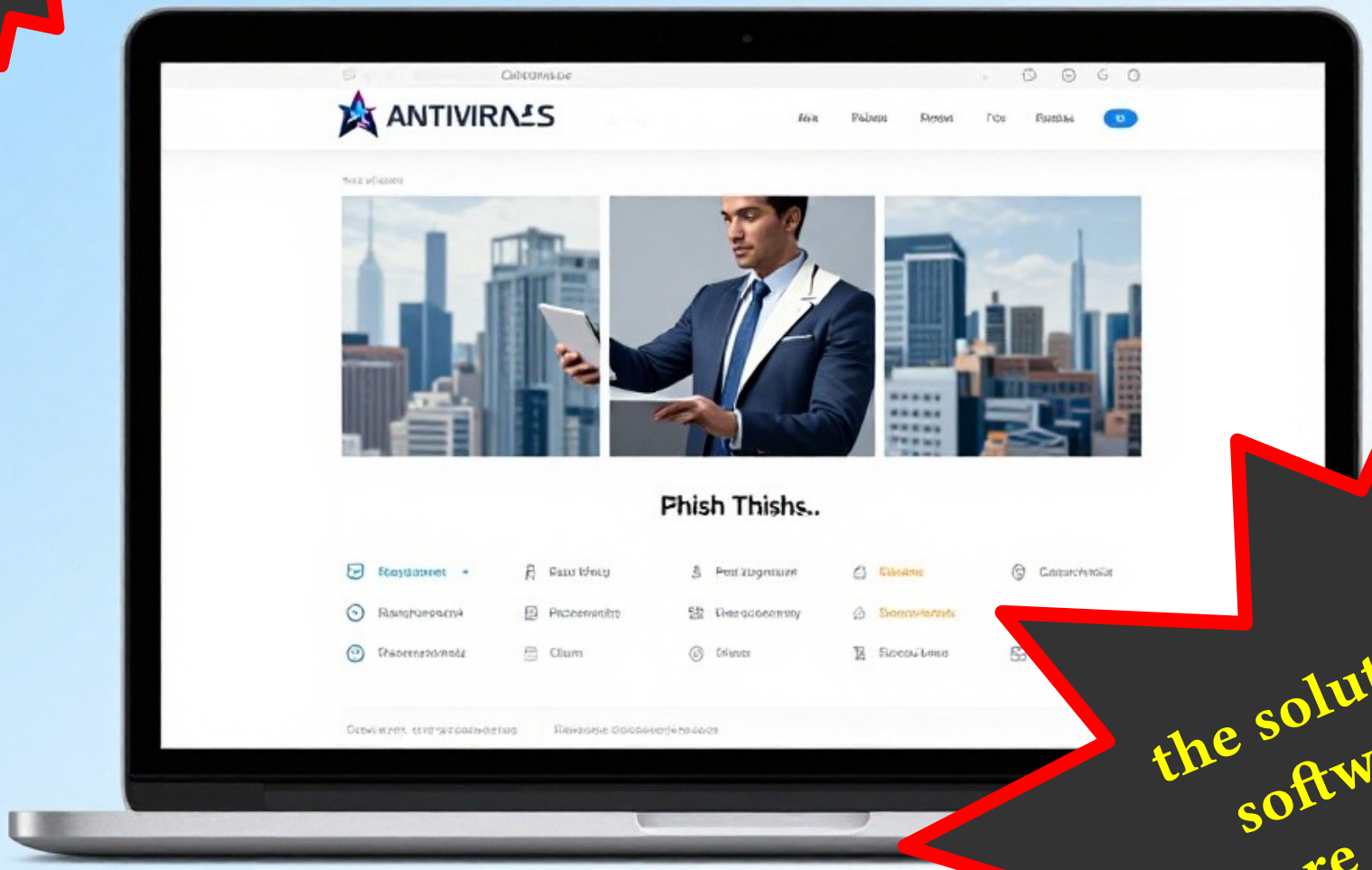
- Wie sollte man Software entwickeln?
- Verifikation als Technik der Qualitätssicherung

- Was ändert sich mit moderner KI?

such wow

Antivos & Phish Protection

much trust-worthy!!



the solution to software is more software

https://freegen.app/

https://www.crowdstrike.com

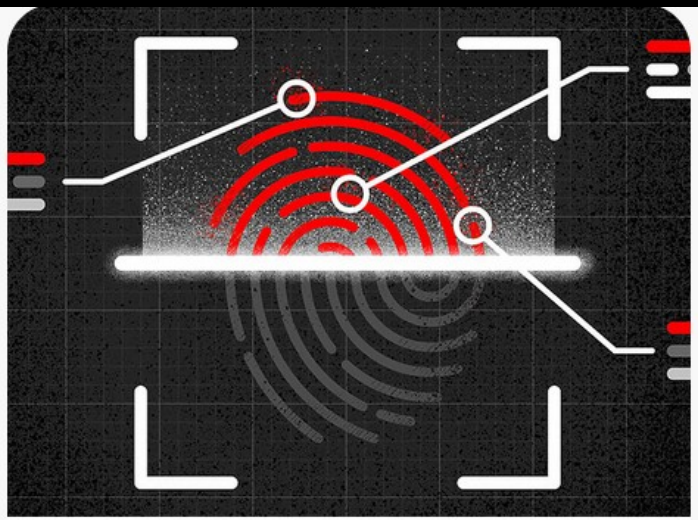
CrowdStrike Falcon® Cloud Security

Secure your cloud

Stop cloud breaches with unified agent and agentless protection. Secure AI with new AI model scanning and AI security dashboard.

Latest news →

Discover Cloud Security →



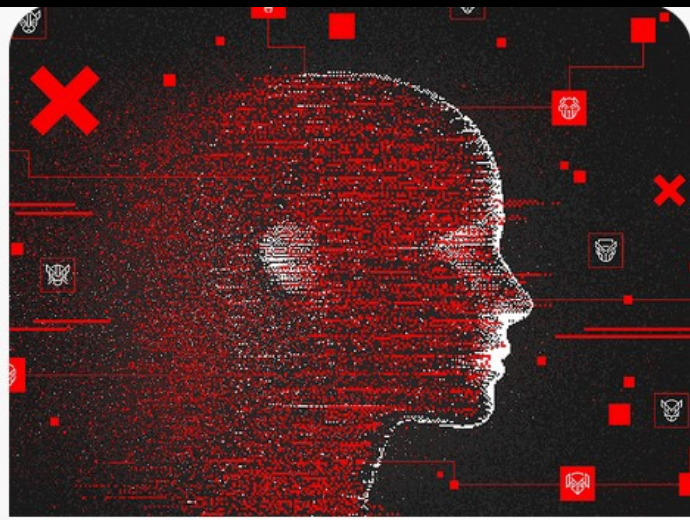
CrowdStrike Falcon® Next-Gen Identity Security

Stop identity attacks

Stop breaches faster with unified security for every identity — human, non-human, AI, and SaaS.

Read blog →

Discover Identity Security →



CrowdStrike Falcon® Next-Gen SIEM

The next SOC era starts here

The AI-native engine of the modern SOC, built to stop breaches — not just log them.

Latest news →

Discover Next-Gen SIEM →

Getty Images

19. Juli 2024

https://www.youtube.com/watch?v=HPAyFCtpnBM


WICHTIGES PROBLEM (aber nicht deins) ↓
SORRY :(
Fehler ist aufgetreten.
:(
Wir arbeiten nicht daran.
Versprochen.
Vielleicht morgen. ¯_(ツ)_/¯
SOFTWAREPROBLEM.
TUT MIR LEID, DA KANN MAN NIX MACHEN.
ERST KAFFEE DANN SUPPORT ♡

Crowdstrike Ausfall --- Folgen

- kaputtes Update auf 8,5 Mio PCs eingespielt
- IT Ausfall
  - 4,6% aller weltweiten Flüge gestrichen
  - digitale Zahlung geht nicht (z.B. 350 Tegut Fillialen schließen)
  - kritische Infrastuktur lahmgelegt (z.B. Krankenhäuser)
- Schaden: > 10 Mrd USD
- Crowdstrike Jahresumsatz 2026: ~ 5 Mrd USD

Crowdstrike Ausfall --- Warum eigentlich??

Technisch

- unsichere Integration ins Betriebssystem
- unsichere Programmiersprache (C, C++)

Organisatorisch

- unzureichende Qualitätssicherung bei der Entwicklung
- keine gestaffelte Auslieferung

Crowdstrike Ausfall --- Warum eigentlich??

Technisch

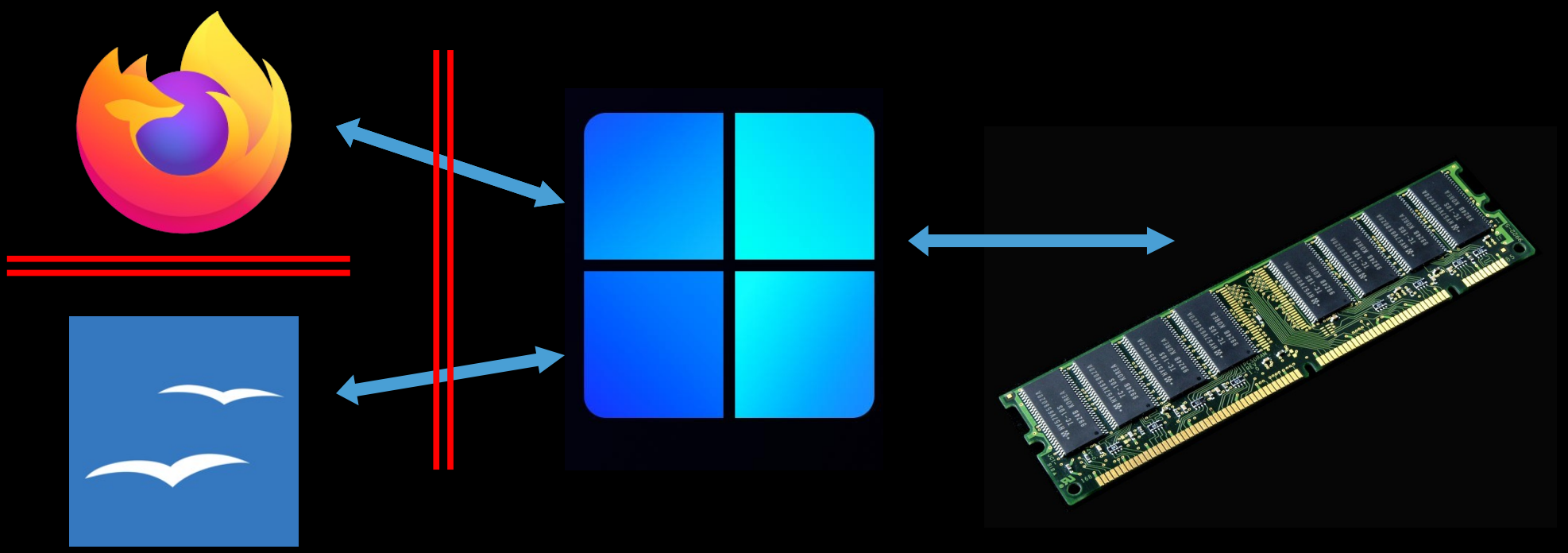
- unsichere Integration ins Betriebssystem
- unsichere Programmiersprache (C, C++)

Organisatorisch

- unzureichende Qualitätssicherung bei der Entwicklung
- keine gestaffelte Auslieferung

Was man tun kann: Sichere Architekturen

Anwendungen Betriebssystem Hardware

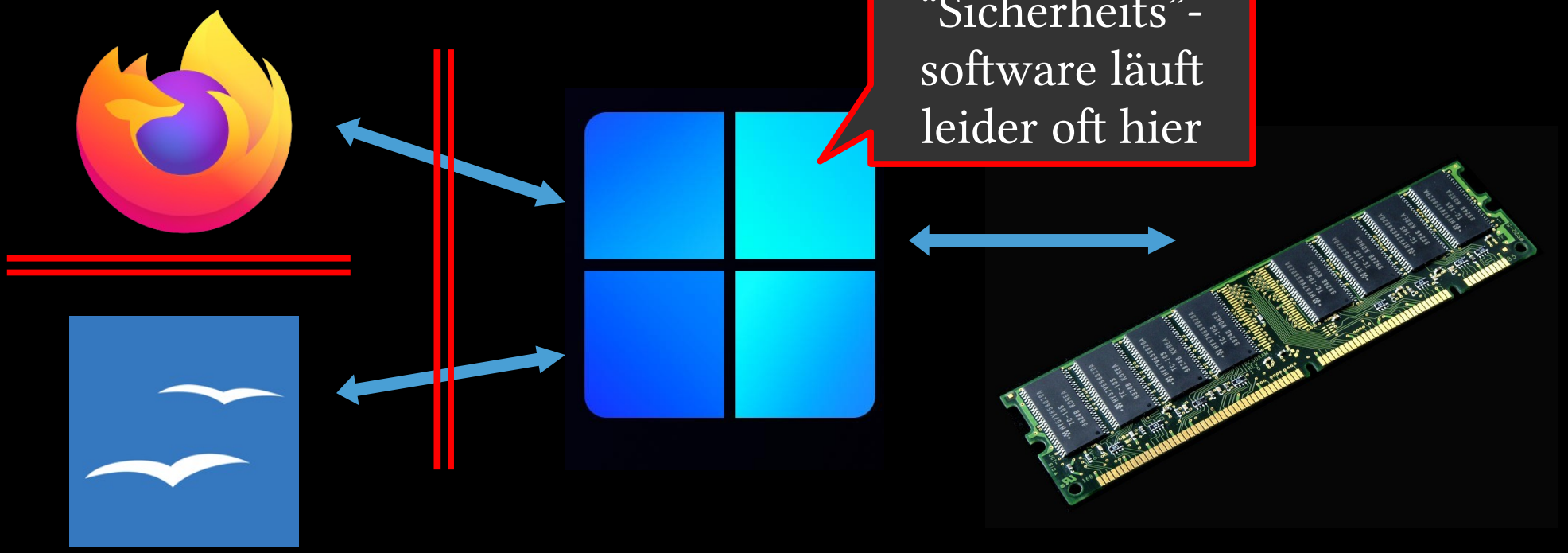


Was man tun kann: Sichere Architekturen

Anwendungen **Betriebssystem** **Hardware**

Was man tun kann: Sichere Architekturen

Anwendungen **Betriebssystem** **Hardware**



Apollo (1969, …)

Margaret Hamilton
(Direktorin der Software-Abteilung NASA/MIT)

When I first came up with the term "Software Engineering", no one had heard of it before, at least in our world. It was an ongoing joke for a long time.

robuste Architektur verhindert Abbruch der Mondlandung (!)

Crowdstrike Ausfall --- Warum eigentlich??

Technisch

- unsichere Integration ins Betriebssystem
- unsichere Programmiersprache (C, C++)

Organisatorisch

- unzureichende Qualitätssicherung bei der Entwicklung
- keine gestaffelte Auslieferung

Pseudo-Code

Demo

```
make 20 buckets b0 ... b19

count := read number of items

for i = 0 ... count-1 do
    item := read the next item
    put item into bucket bi
```

White House urges developers to avoid C and C++, use 'memory-safe' programming languages

News By Les Pounder published February 28, 2024

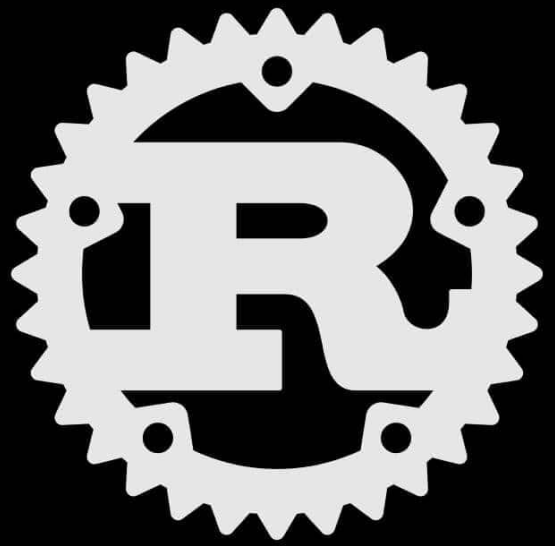
told you so

The languages may pose a security risk when used in critical systems.

https://www.tomshardware.com

Was man tun kann: Rust (ab 2012)

https://www.rust-lang.org/



- Graydon Hoare: *technology from the past come to save the future from itself*

- Ziel: **schnelle** aber **sichere** Programme
- Design: **pragmatisch** aber **prinzipienbasiert**

- Community!

Crowdstrike Ausfall --- Warum eigentlich??

Technisch

- unsichere Integration ins Betriebssystem
- unsichere Programmiersprache (C, C++)

Organisatorisch

- unzureichende Qualitätssicherung bei der Entwicklung
- keine gestaffelte Auslieferung

TESTS WON’T FAIL

IF YOU DON’T WRITE ANY TESTS

Was man tun kann:

Testen!

https://sqlite.org/testing.html

SQLite

Small. Fast. Reliable.
Choose any three.

Home | About | Documentation | Download | License | Support | Purchase | Search

How SQLite Is Tested

▸ Table Of Contents

1. Introduction

The reliability and robustness of SQLite is achieved in part by thorough and careful testing.

As of version 3.42.0 (2023-05-16), the SQLite library consists of approximately 155.8 KSLOC of C code. (KSLOC means thousands of "Source Lines Of Code" or, in other words, lines of code excluding blank lines and comments.) By comparison, the project has 590 times as much test code and test scripts - 92053.1 KSLOC.

1.1. Executive Summary

- Four independently developed test harnesses
- 100% branch test coverage in an as-deployed configuration
- Millions and millions of test cases
- Out-of-memory tests
- I/O error tests
- Crash and power loss tests
- Fuzz tests
- Boundary value tests
- Disabled optimization tests
- Regression tests
- Malformed database tests
- Extensive use of assert() and run-time checks
- Valgrind analysis
- Undefined behavior checks
- Checklists

Was man tun kann:

Testen!

https://sqlite.org/testing.html

SQLite

Small. Fast. Reliable. Choose any three.

Home About Documentation Download License Support Purchase Search

How SQLite Is Tested

▸ Table Of Contents

1. Introduction

The reliability and robustness of SQLite is achieved in part by thorough and careful testing.

As of version 3.42.0 (2023-05-16), the SQLite library consists of approximately 155.8 KSLOC of C code. (KSLOC means thousands of "Source Lines Of Code" or, in other words, lines of code excluding blank lines and comments.) By comparison, the project has 590 times as much test code and test scripts - 92053.1 KSLOC.

As of version 3.42.0 (2023-05-16), the SQLite library consists of approximately 155.8 KSLOC of C code. (KSLOC means thousands of "Source Lines Of Code" or, in other words, lines of code excluding blank lines and comments.) By comparison, the project has 590 times as much test code and test scripts - 92053.1 KSLOC.

- Crash and power loss tests
- Fuzz tests
- Boundary value tests
- Disabled optimization tests
- Regression tests
- Malformed database tests
- Extensive use of assert() and run-time checks
- Valgrind analysis
- Undefined behavior checks
- Checklists

Was man tun kann: Verifikation

Demo

CPA

SoSy-Lab
Software Systems

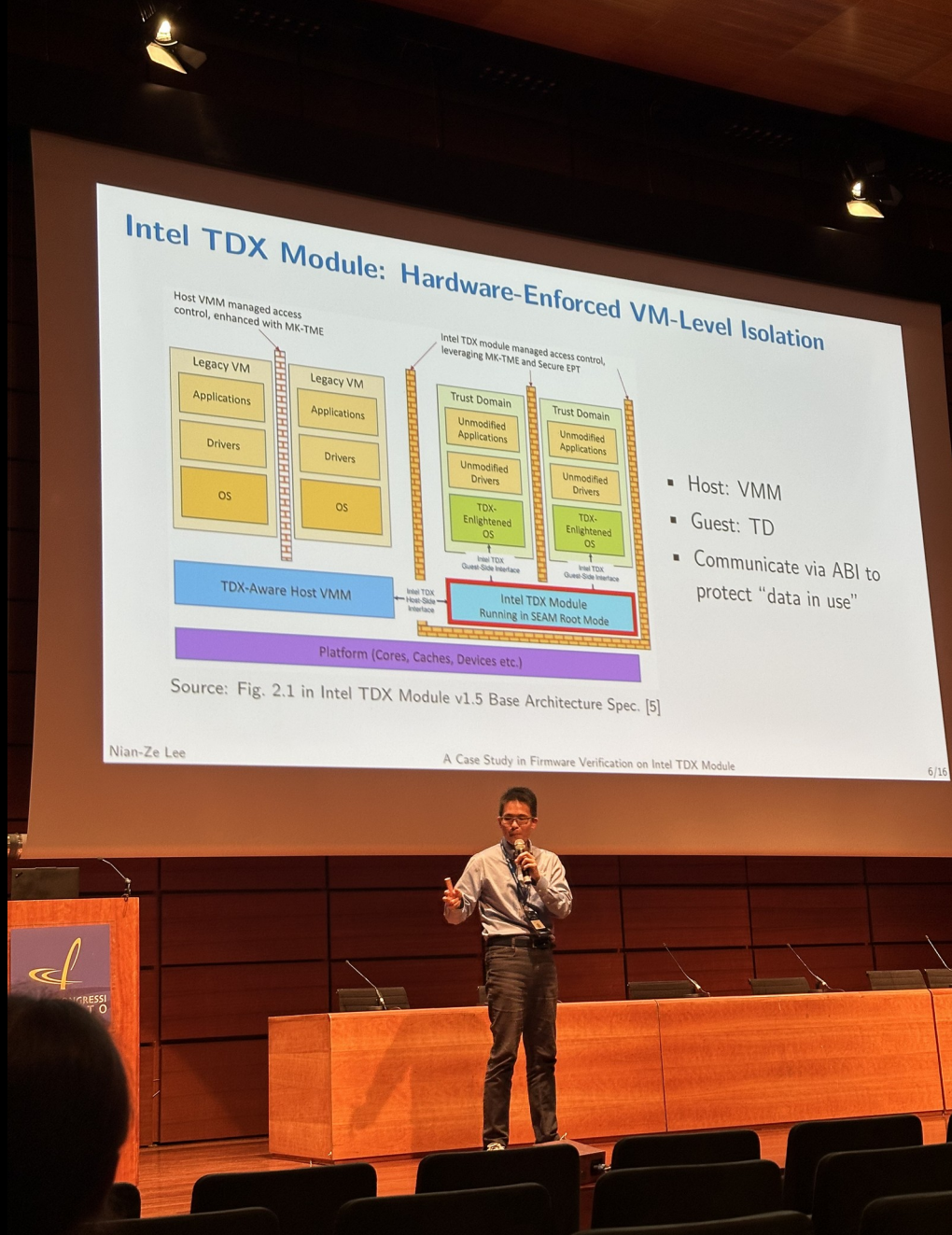
Gödel-Medaille 2014 für
Dirk Beyer et al

CPAchecker in der Praxis

Dirk Beyer, Po-Chun Chien, Bo-Yuan Huang, Nian-Ze Lee and Thomas Lemberger:

A Case Study in Firmware Verification: Applying Formal Methods to Intel TDX Module

ETAPS 2026 distinguished paper



Crowdstrike Ausfall --- Warum eigentlich??

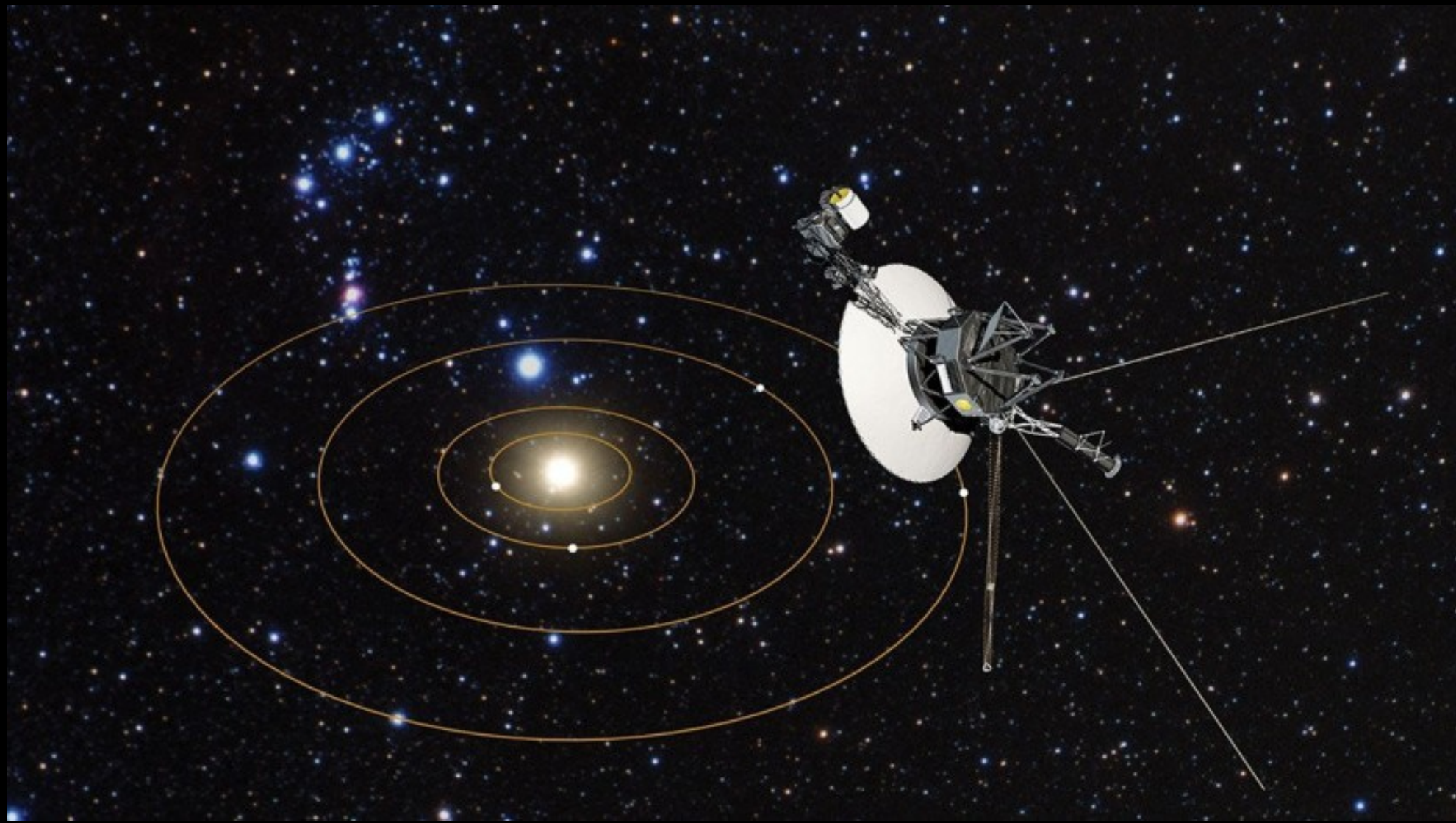
Technisch

- unsichere Integration ins Betriebssystem
- unsichere Programmiersprache (C, C++)

Organisatorisch

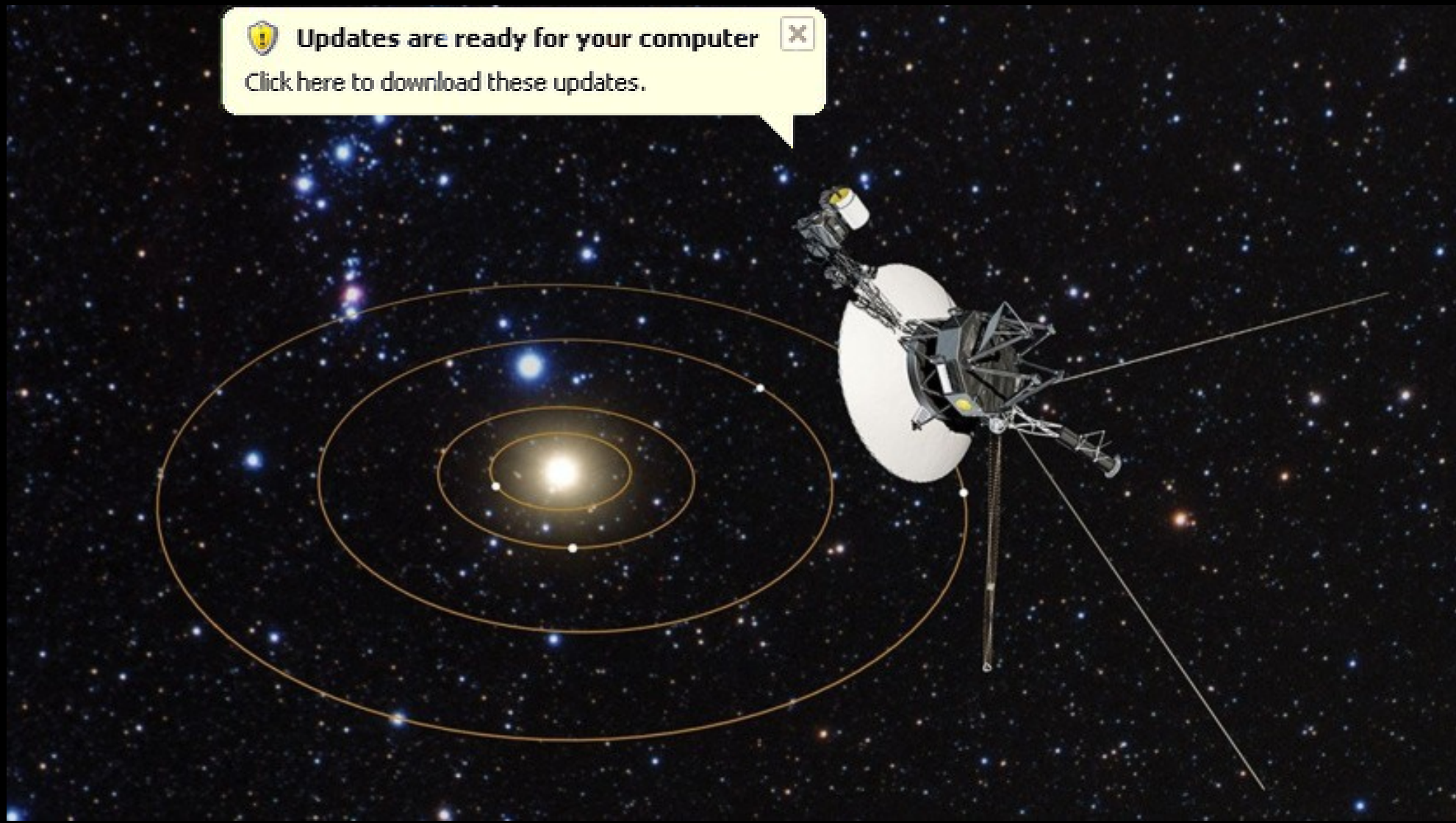
- unzureichende Qualitätssicherung bei der Entwicklung
- keine gestaffelte Auslieferung

Voyager I und II (Start: 1977)



NASA, ESA, and G. Bacon (STScI)

Voyager I und II (Start: 1977)



NASA, ESA, and G. Bacon (STScI)

Agenda

- Fallstudie: Crowdstrike

- **Wie sollte man Software entwickeln?**
- Verifikation als Technik der Qualitätssicherung

- Was ändert sich mit moderner KI?

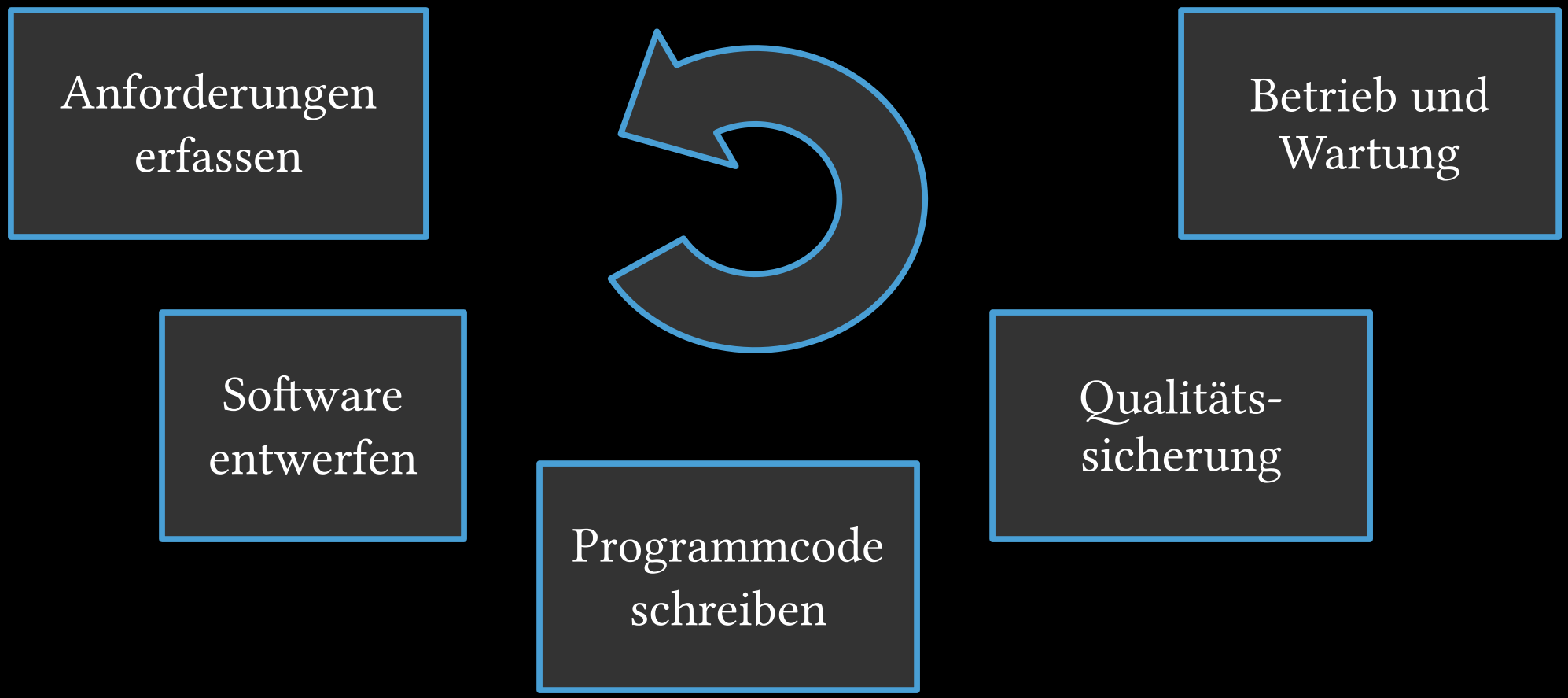
Ein “Erfindungs-verwaltungssystem” bitte

“der Kunde”

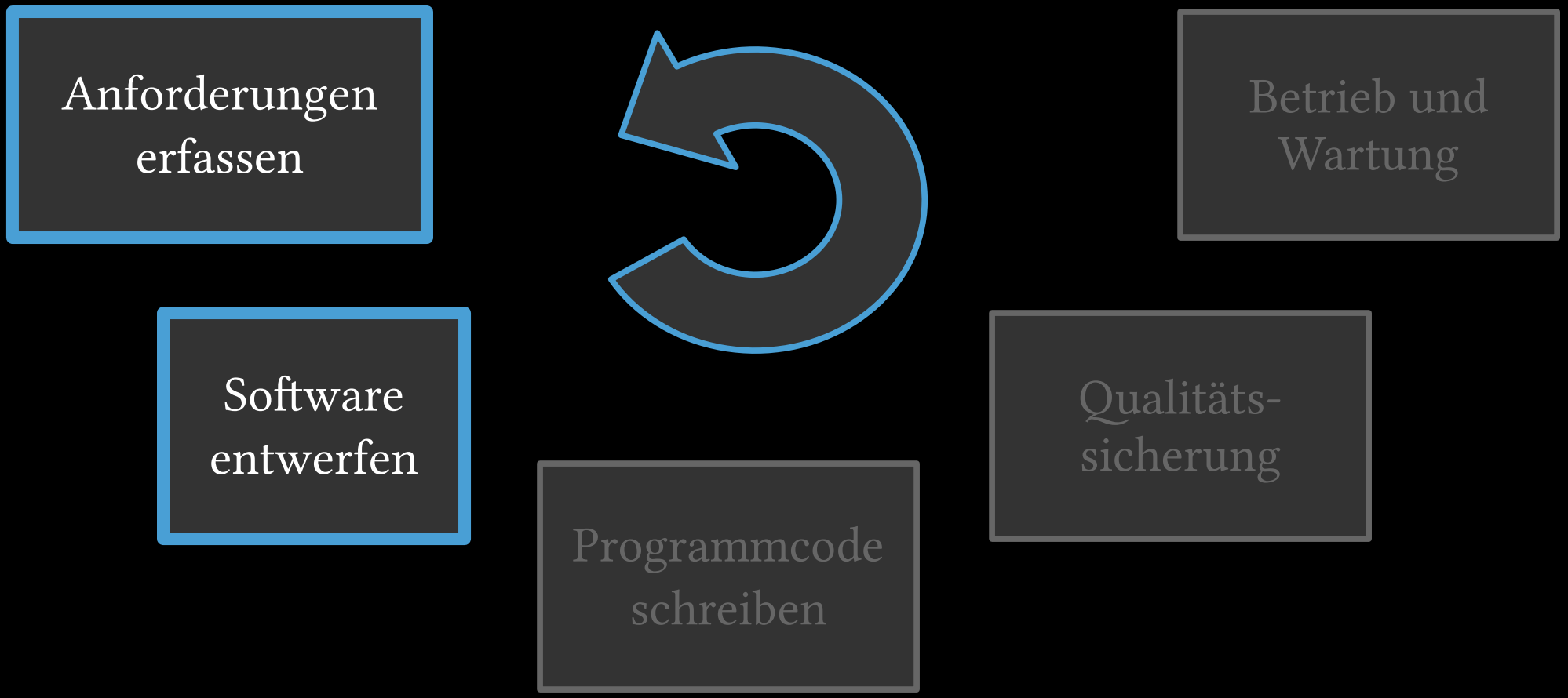
Coole Idee!
Was soll es denn können?
Wie soll es denn funktionieren?

“die Entwicklerin”

Entwicklung als kontinuierlicher Prozess



Entwicklung als kontinuierlicher Prozess



Anforderungen

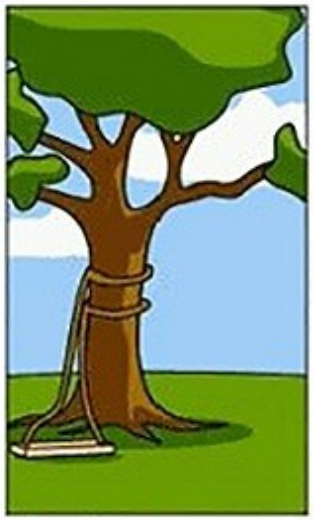
Erwin Wurm

Truck (2015)

How the customer explained it

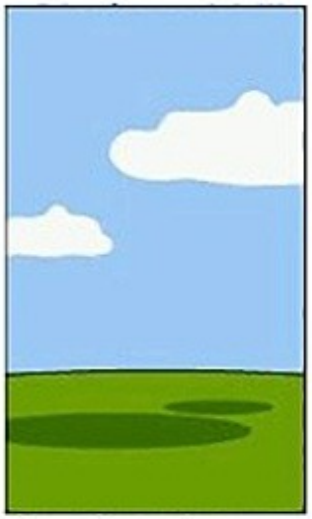
How the project leader understood it

How the engineer designed it

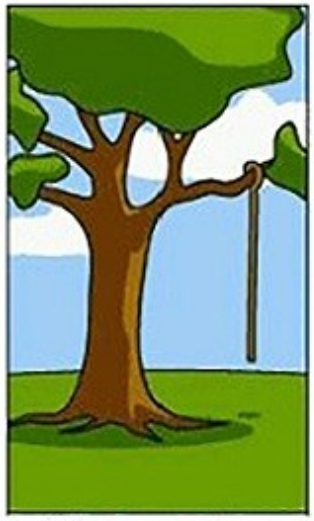


How the programmer wrote it

How the sales executive described it

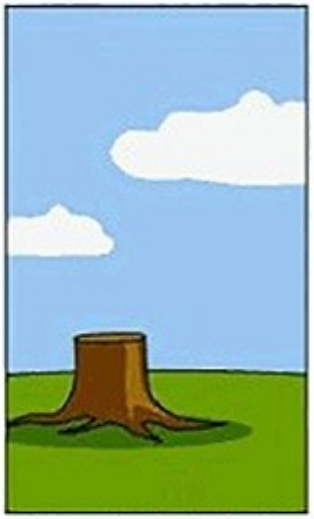


How the project was documented



What operations installed

How the customer was billed

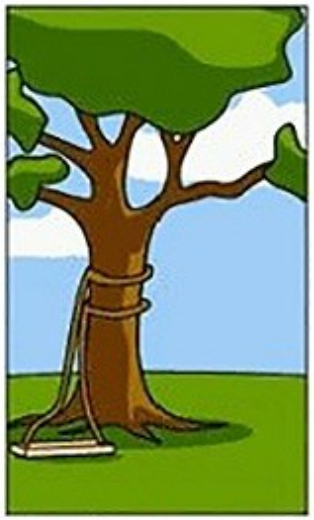


How the helpdesk supported it

How the customer explained it

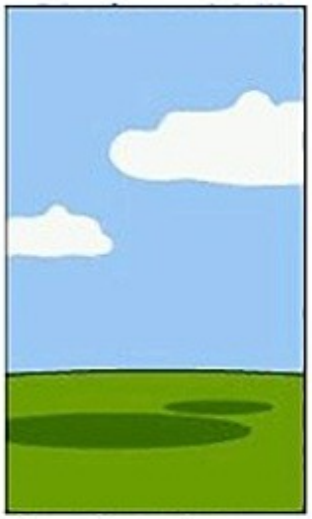
How the project leader understood it

How the engineer designed it

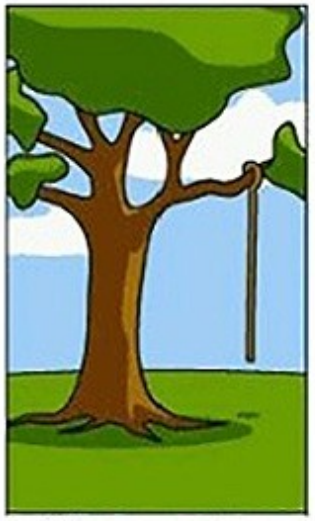


How the programmer wrote it

How the sales executive described it

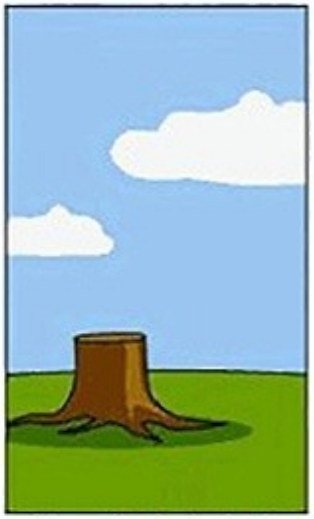


How the project was documented



What operations installed

How the customer was billed



How the helpdesk supported it

What the customer really needed

Entwicklung als kontinuierlicher Prozess

Anforderungen erfassen

Software entwerfen

Programmcode schreiben

Qualitäts-sicherung

Betrieb und Wartung

Grace Hopper (ca 1960)
Computerpioneerin
Designerin von COBOL

Quiz

(Wo) kommt COBOL heute noch zum Einsatz?

Legacy-Software (COBOL)

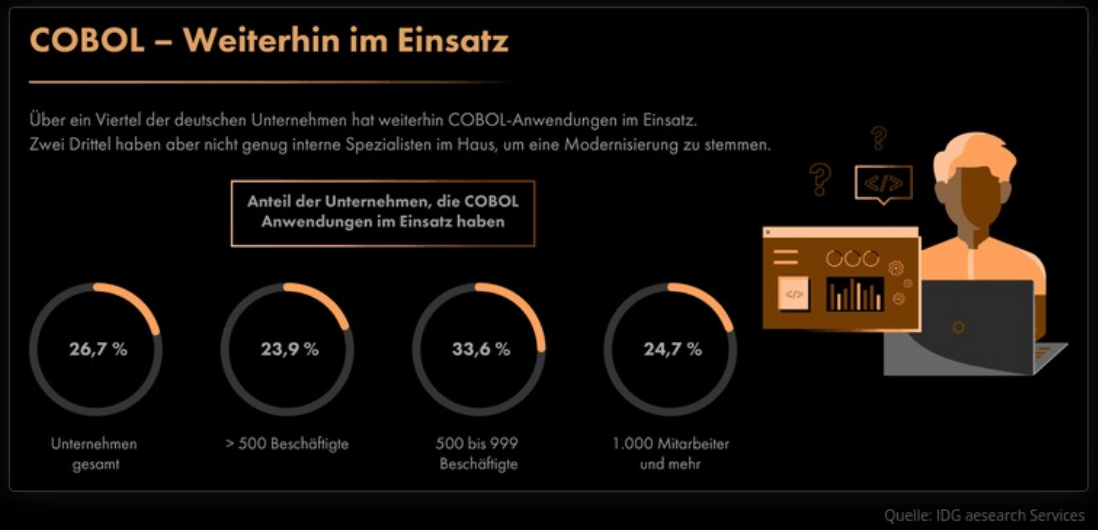
https://www.it-finanzmagazin.de/banken-versicherer-und-ihr-cobol-problem-134184/

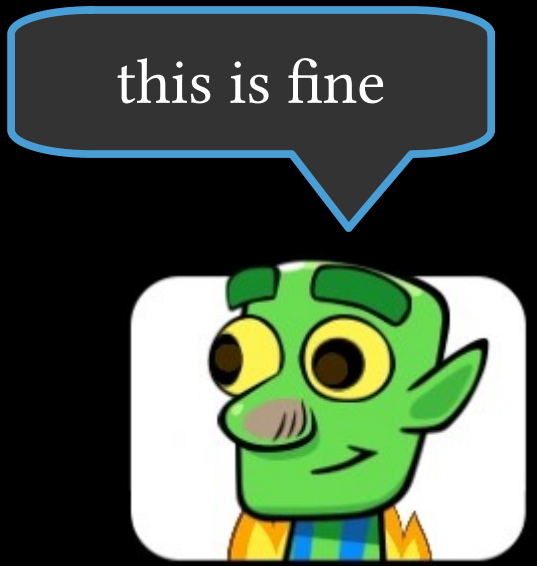
50% aller Banken basieren auf COBOL

80% aller Kreditkartentransaktionen

95% aller Bankautomatentransaktionen

800 Mrd Zeilen COBOL Code weltweit





Agenda

- Fallstudie: Crowdstrike

- Wie sollte man Software entwickeln?
- **Verifikation als Technik der Qualitätssicherung**

- Was ändert sich mit moderner KI?

Man könnte doch “genau hinschauen”

```
make 20 buckets b₀ ... b₁₉

count := read number of items

for i = 0 ... count-1 do
    item := read the next item
    put item into bucket bᵢ
```

wann geht das gut?

Man könnte doch “genau hinschauen”

```
make 20 buckets b₀ ... b₁₉

count := read number of items

for i = 0 ... count-1 do
    item := read the next item
    put item into bucket bᵢ
```

i < 20

Man könnte doch “genau hinschauen”

```
make 20 buckets b0 ... b19

count := read number of items

for i = 0 ... count-1 do
    item := read the next item
    put item into bucket bi
```

i < count

i < 20

Man könnte doch “genau hinschauen”

```
make 20 buckets b₀ ... b₁₉

count := read number of items

for i = 0 ... count-1 do
    item := read the next item
    put item into bucket bᵢ
```

count ⩽ 20

i < count

i < 20

Man könnte doch “genau hinschauen”

```
make 20 buckets b0 ... b19

count := read number of items

for i = 0 ... count-1 do
    item := read the next item
    put item into bucket bi
```

nicht garantiert!

count ⩽ 20

i < count

i < 20

Verifikationstools

Anforderungen
(kein Absturz, …)

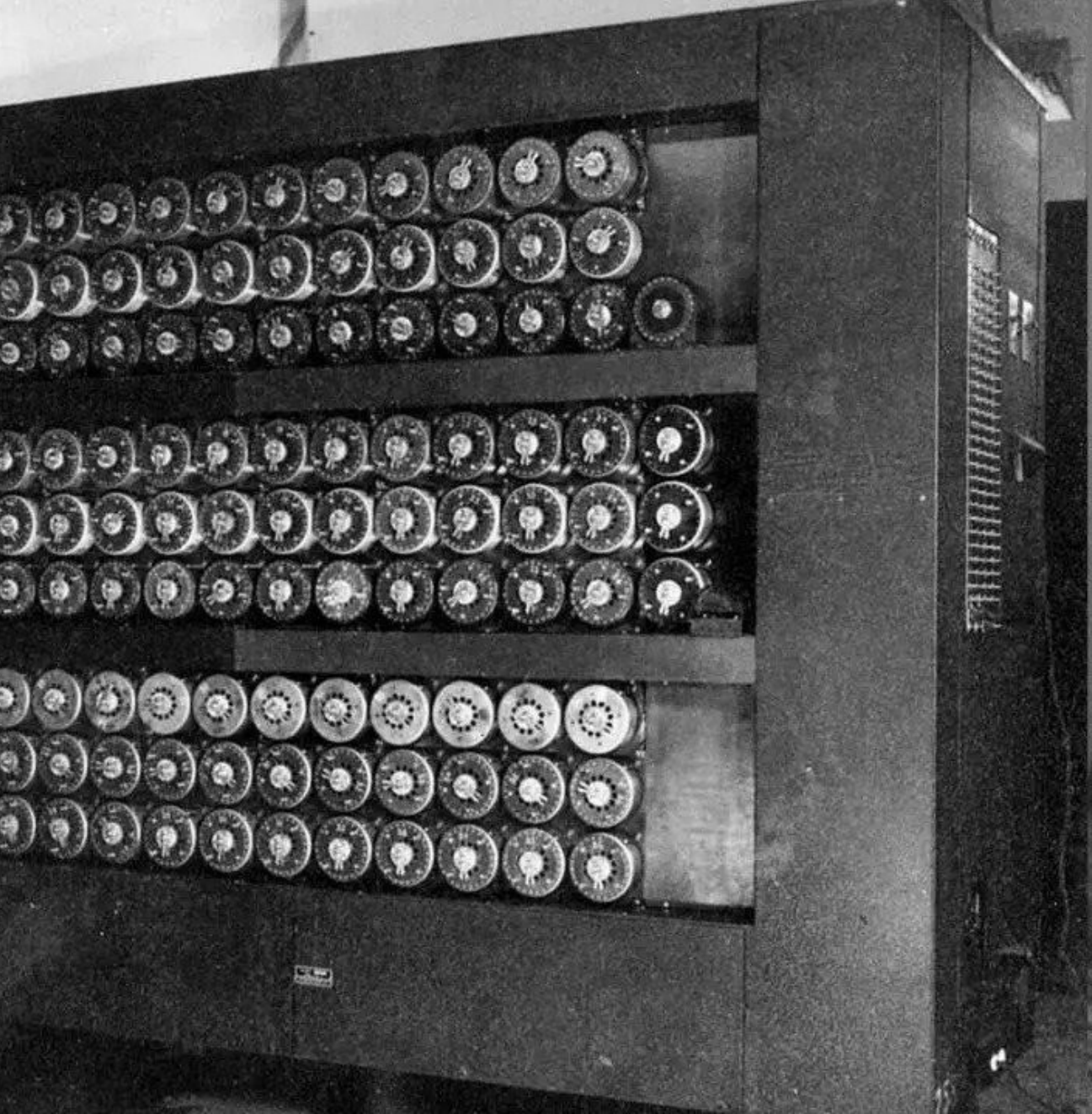
crowdstrike.c

```c
#include <stdio.h>

int read_item_count() { return 30; }
int read_item(int i) { return i; }

int main()
{
    int b[20]; // 20 buckets

    int count = read_item_count();

    for(int i = 0; i < count; i++) {
        printf("storing item %d\n", i);
        b[i] = read_item(i);
    }
}
```

Korrektheitsbeweis

Fehlerbeschreibung

Timeout



Herausforderungen (Theorie)

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. Turing.

[Received 28 May, 1936.—Read 12 November, 1936.]

Es kann kein perfektes Verifikationstools geben

Beweise sind fundamental schwierig

Herausforderungen & Erfolge (Praxis)

- große Programme (> Mio Zeilen Code)
- komplexe Berechnungen und Datenstrukturen

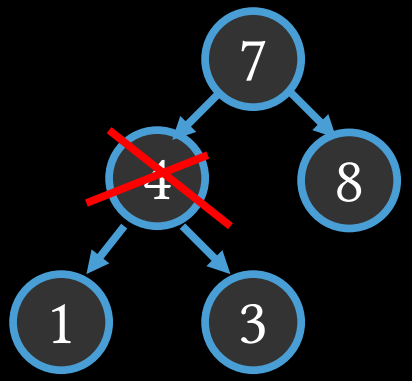
- Microsoft: Treiberverifikation (viel weniger Bluescreens!)
- Google: kontinuierliche Fehlersuche in Linux
- Apple: Verifikation von Crypto-Code (in München!)
- Amazon: Verifikation von Cloud-Systemen

Metro 14 Paris

Meine Forschung

Spezifikation und Verifikation komplexer Systeme

Automatisierung von Beweisen mit Datenabstraktion



=

[1,3,4,7,8]

Wie sieht das konkret aus?

[Ernst 2026 – under submission]

Algorithm 1 (Verification Conditions) *The weakest precondition is computed by structural recursion on the program for an arbitrary ψ:*

$$wp(\varphi \mid \psi) \iff (\varphi \Rightarrow \psi) \tag{2}$$

$$wp((c_1 \sqcup c_2); \kappa \mid \psi) \iff wp(c_1; \kappa \mid \psi) \wedge wp(c_2; \kappa \mid \psi) \tag{3}$$

$$wp((c_1; c_2); \kappa \mid \psi) \iff wp(c_1; (c_2; \kappa) \mid \psi) \tag{4}$$

$$wp(A; \kappa \mid \psi) \iff \big(\forall\, \overline{x}'.\ step(A, \overline{x}, \overline{x}') \Rightarrow wp(\kappa' \mid unfold(\psi, \overline{x}, \overline{x}'))\big) \tag{5}$$

$$wp(\texttt{iter}(c); \kappa \mid \psi) \iff \textit{exists formula } \iota,\ \textit{closure operator } \mathcal{H}_\iota \textit{ for } \iota, \tag{6}$$
$$\textit{and fresh placeholder } \omega \textit{ so that}$$
$$(\iota \Rightarrow \psi) \wedge \forall\, \overline{x}.\ wp(\kappa \mid \iota) \wedge wp(c; \omega(\overline{x}) \mid \mathcal{H}_\iota(\omega(\overline{x})) \Rightarrow \iota)$$

Agenda

- Fallstudie: Crowdstrike

- Wie sollte man Software entwickeln?
- Verifikation als Technik der Qualitätssicherung

- **Was ändert sich mit moderner KI?**

Meta-CEO Zuckerberg: "KI wird besseren Code schreiben als Entwickler"

Laut CEO Nadella schreibt KI rund 30 Prozent des Microsoft-Codes. Zuckerberg schätzt den Anteil bei Meta höher und glaubt an das Potenzial von KI-Code.

(Bild: iX)

Anthropics neues KI-Modell könnte Cyberlandschaft "umwälzen"

Anthropic will mit Mythos Tausende teils kritische Software-Lücken entdeckt haben. Das BSI erwartet erhebliche Folgen für den Cybersektor.

10. April 2026 um 09:08 Uhr / Marc Stöckel

Bild: Sean Gallup/Getty Images

BSI-Chefin Claudia Plattner äußert sich zu Anthropics neuem KI-Modell Mythos.

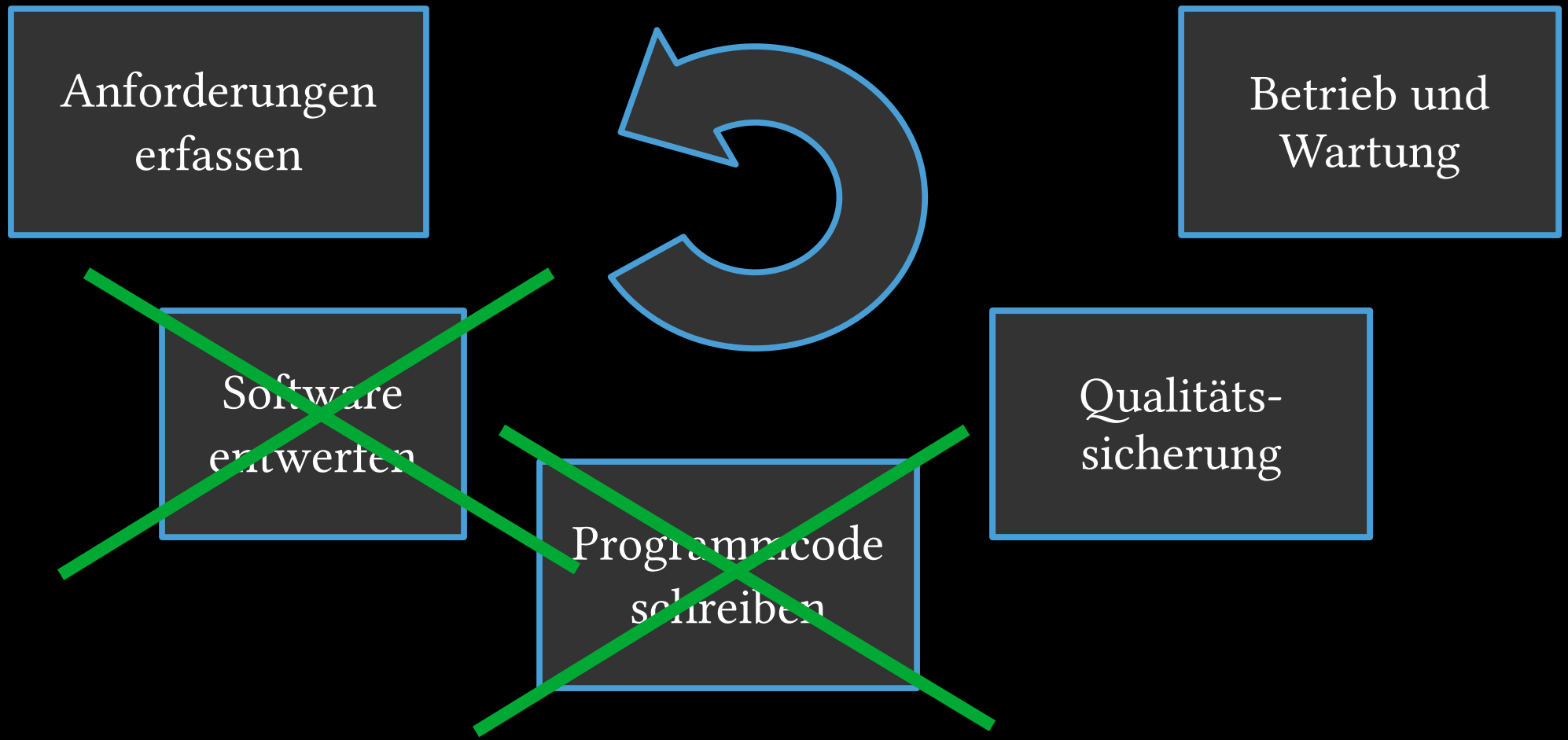
Wenn man dem Hype glaubt (und es ist was dran)

- **Sie/Ihr** könnt Software entwickeln – heute schon

- Entwickler schreiben oft selbst keinen Code mehr, sondern reviewen nur noch – heute schon

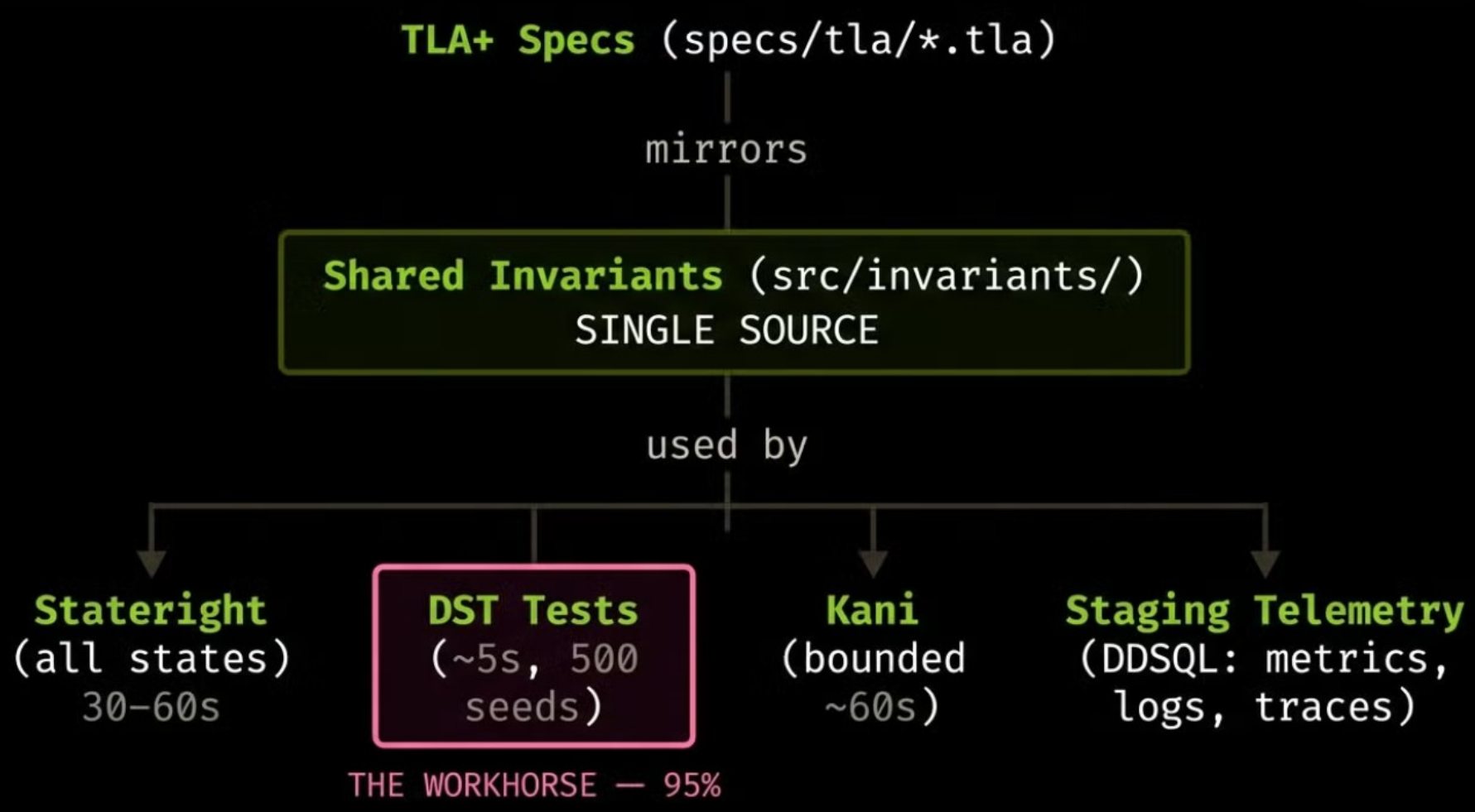
- Es wird keine geheimen Sicherheitslücken mehr geben
- KI unterstütz Verifikationstools

Wenn man dem Hype glaubt (und es ist was dran)



Code-Generierung und Verifikation mit KI

https://www.datadoghq.com/blog/ai/harness-first-agents/




Guten Morgen!
Softwareentwicklung ist eine Herausforderung
aber es gibt viele Techniken die dabei helfen