



# Einfachheit in Softwareprojekten

Stefan Rook

[stefan.rook@akquinet.de](mailto:stefan.rook@akquinet.de)

# Theorie der Einfachheit.

## Begriff

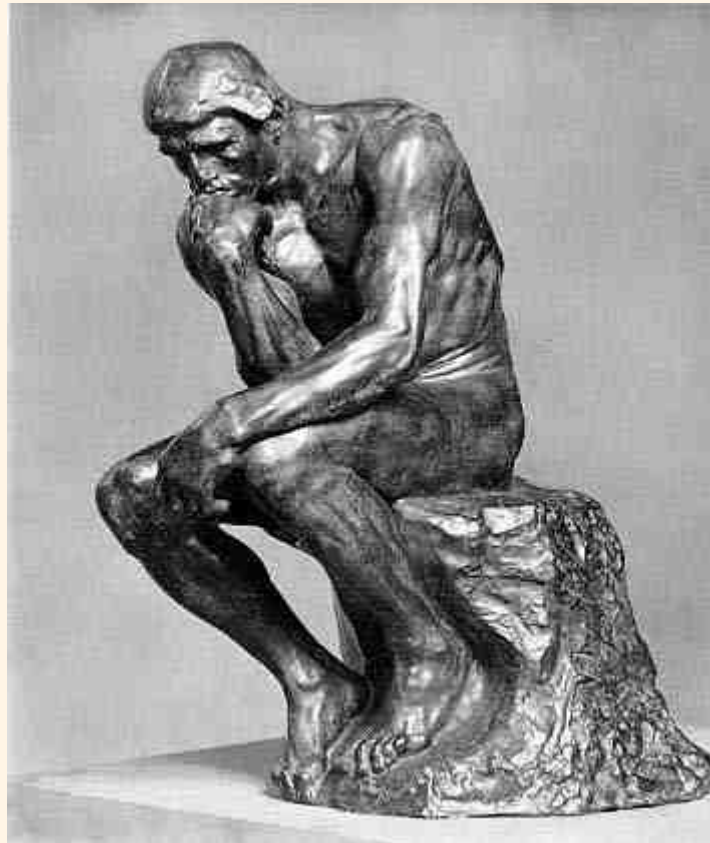
Einfachheit ist schwer definierbar.

Es ist aber in konkreter Situation und Kontext leicht,  
*x ist einfacher als y*

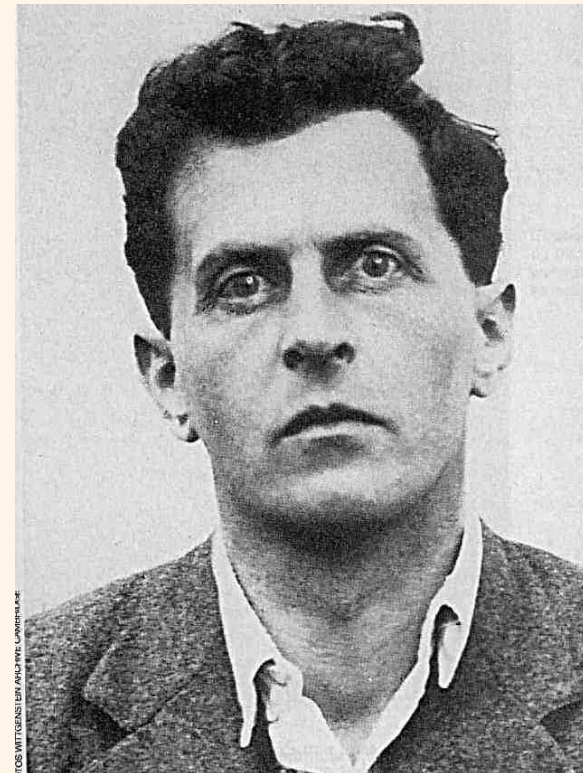
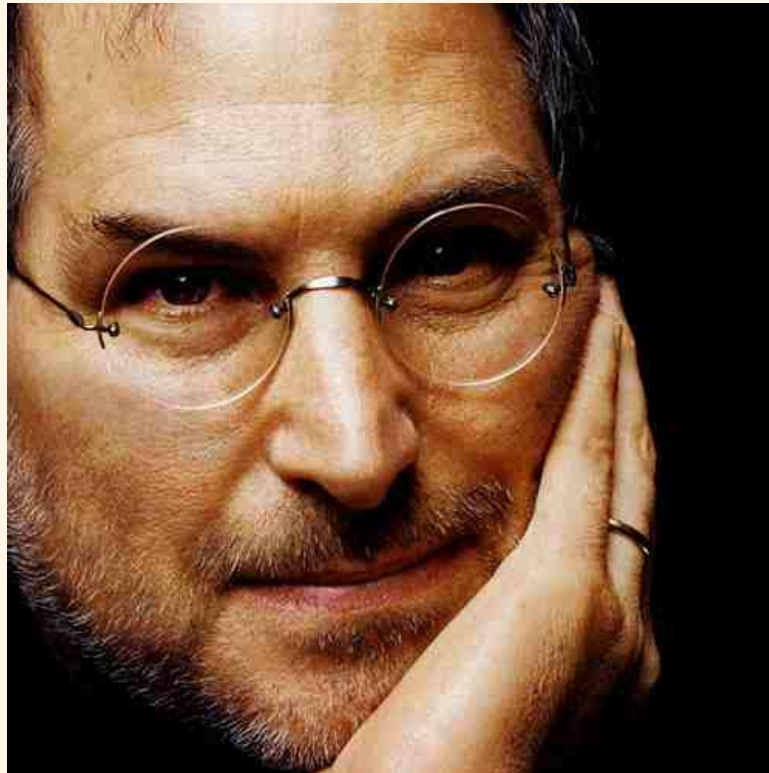
festzustellen.

Simple != Easy

~~Theorie der Einfachheit.~~



## Die Jobs/Wittgenstein-Episode





Relativ zur Konkurrenz:

- Kann weniger.
- Kostet mehr.
- Verkauft sich besser.

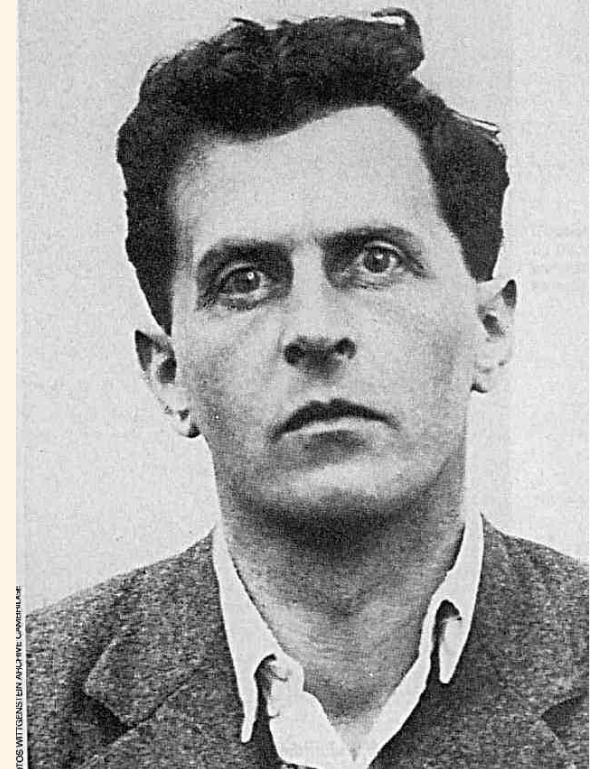
„Einfachheit ist wünschenswert – auch in Softwareprojekten.“



## Philosophie

*„Was gesagt werden kann,  
kann auch einfach gesagt werden.“*

Wittgenstein



## Die Kaizen-Episode



Weniger → Einfacher

Das überlegte Weglassen führt zu Einfachheit.

## Perfektion

*"Ein Ding ist nicht dann perfekt, wenn man nichts mehr hinzufügen kann, sondern dann, wenn man nichts mehr weglassen kann."*

Antoine de Saint-Exupéry

Kaizen: Verschwendung (Muda) eliminieren

改善

無駄

Verschwendung: Geht nicht direkt in das Produkt ein.

- Anforderungsanalyse
- Nightly-Build
- Compile-Target im Build-Skript
- Automatisierte Tests, die nie fehlschlagen
- Testen, um Fehler zu finden
- Anforderungen aufschreiben
- Metriken über das System erheben
- ...

Anforderungsanalyse

Nightly-Build

Compile-Target im Build-Skript

hinterher Testen

**Verschwendung:  
Geht nicht direkt in das Produkt ein.**

...

Anforderungen aufschreiben

Metriken über das System erheben

## Die Dueck/Cohen/Stewart-Episode





## Technik: Hindernislisten

Hindernis	Seit wann?	Wann beseitigt?
<del>Kunde nicht erreichbar.</del>	01.10.2007	02.10.2007
Akzeptanztests kaputt.	02.10.2007	
<del>Baulärm.</del>	02.10.2007	02.10.2007
...		

## Umgang mit Hindernislisten

- Probleme lange offen?
  - Problem im Team
  
- Keine neuen Probleme?
  - Problem im Team

Hindernis	Seit wann?	Wann beseitigt?
<del>Kunde nicht erreichbar.</del>	01.10.2007	02.10.2007
Akzeptanztests kaputt.	02.10.2007	
<del>Baulärm.</del>	02.10.2007	02.10.2007
...		

## Gewöhnung

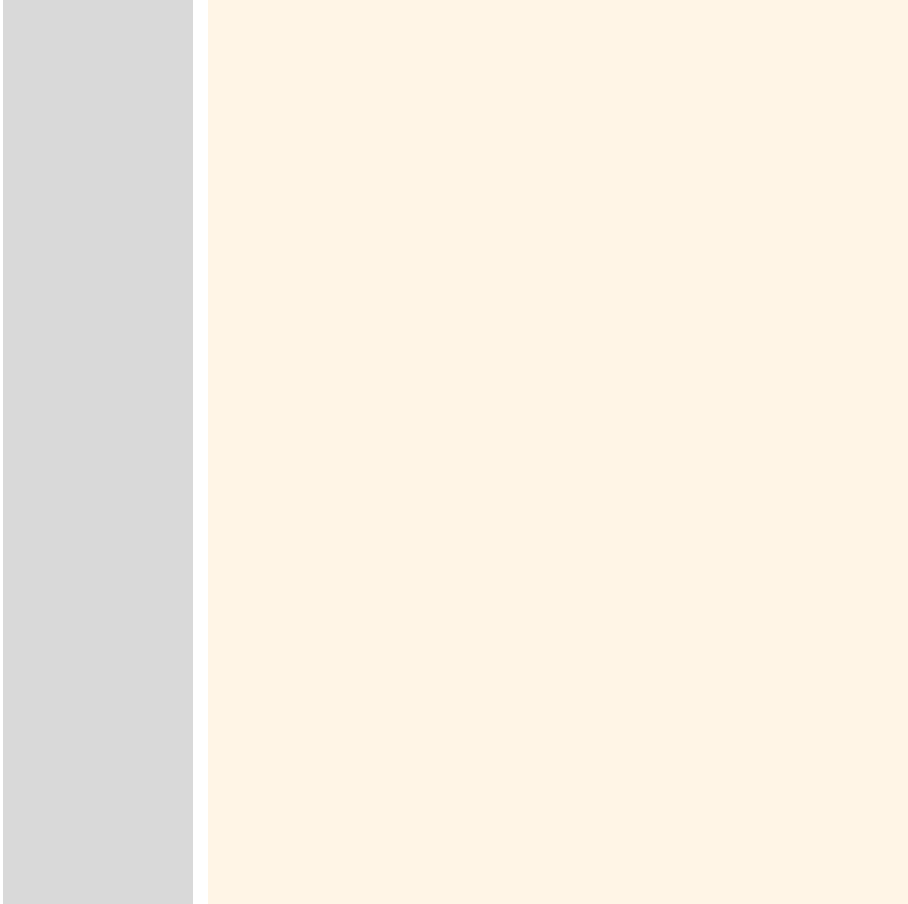
„Man kann das nicht besser erklären.  
Es ist nun mal schwer.  
Man gewöhnt sich daran, es nicht zu  
verstehen, so dass es sich bald so normal  
anfühlt, dass man es zu verstehen glaubt.“

Gunther Dueck



## „Komplizität“

Cohen/Stewart

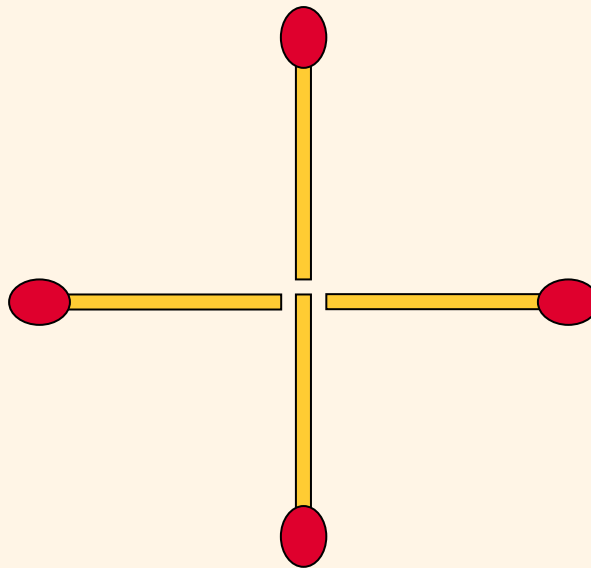


## Die Johannes-Kepler-Episode



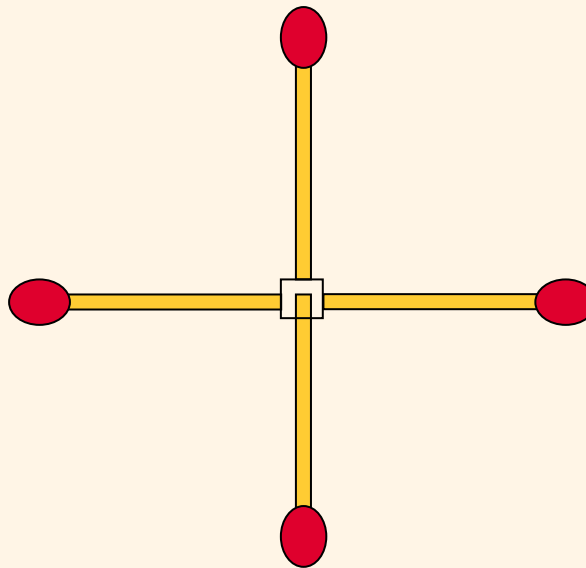
## Eine kleine Aufgabe zum Nachdenken

- Legen Sie **ein** Streichholz um, so dass ein Quadrat entsteht!



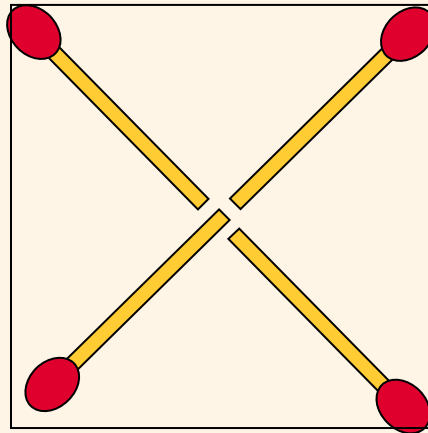
## 1. Lösung zur Aufgabe

- Legen Sie ein Streichholz um, so dass ein Quadrat entsteht!



## 2. Lösung zur Aufgabe

- Legen Sie ein Streichholz um, so dass ein Quadrat entsteht!



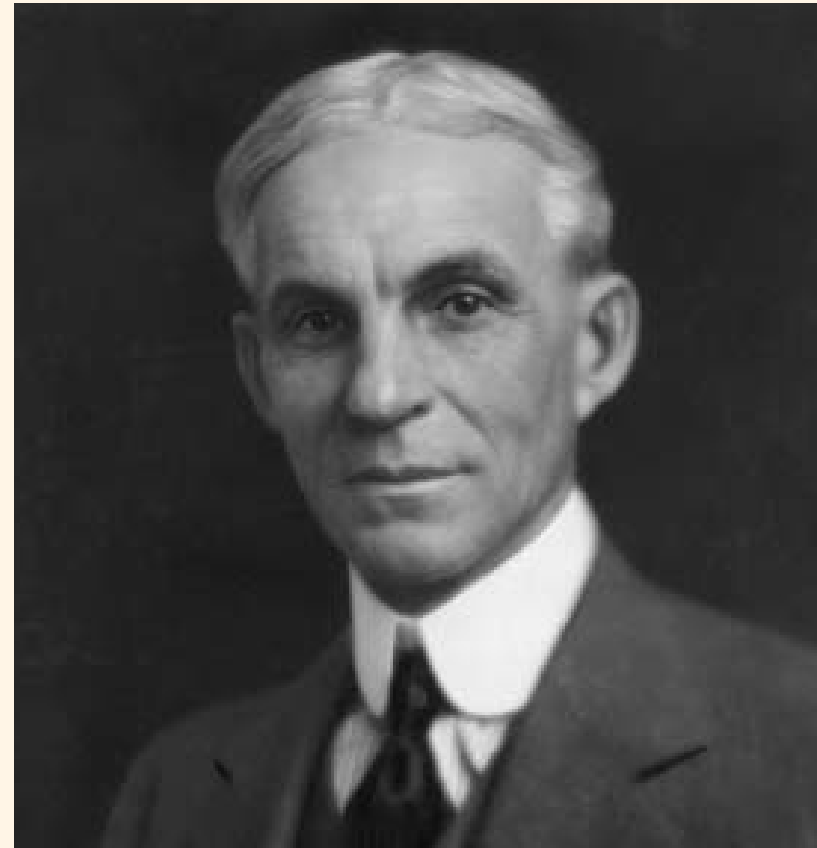


## Problem, Kontext, Lösung

- Können wir Problem oder Kontext anders wahrnehmen?
- Können wir Problem oder Kontext ändern?



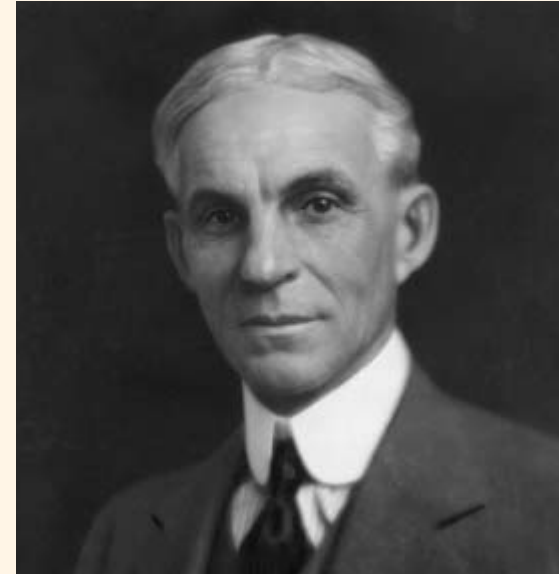
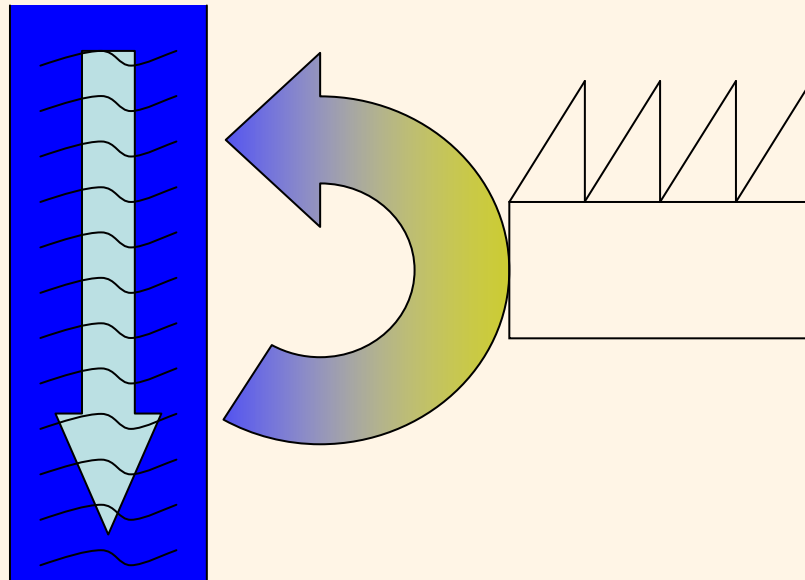
## Die Henry-Ford-Episode





## Eat Your Own Dogfood

- Henry Ford: „Jedes Unternehmen muss soviel Wasser flussaufwärts einleiten, wie es flussabwärts entnimmt.“



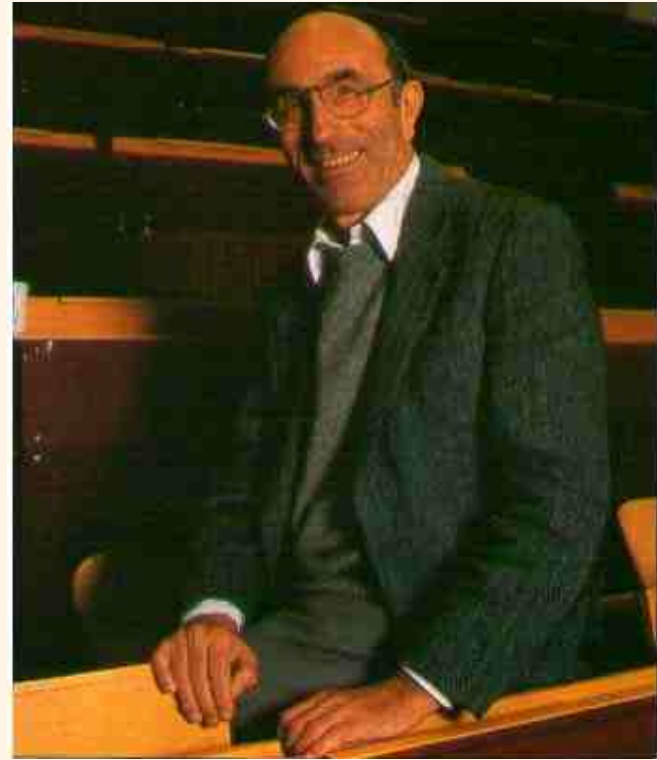
## Qualitätszyklen reduzieren Regelungsbedarf

- ...und vereinfachen damit Prozesse.

### **Beispiele**

- Verantwortung für Wartung im Entwicklungsteam belassen.
- ...

## Die Niklas-Luhmann-Episode



Qualitätszyklus unmöglich?

# Vertrauen

in Menschen

**„Vertrauen. Ein Mechanismus der Reduktion sozialer Komplexität.“  
Niklas Luhmann**

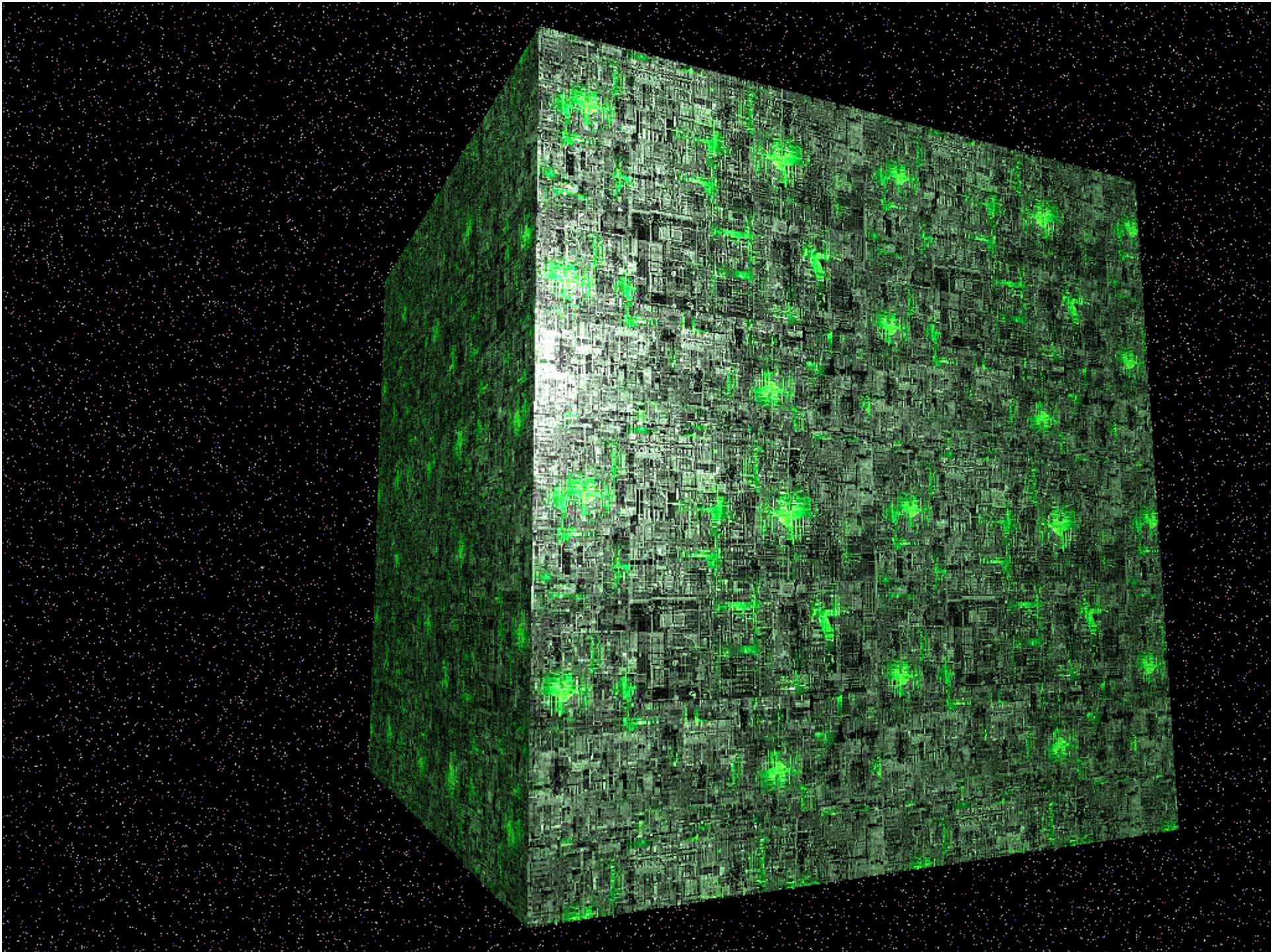




# Die Borg-Episode



by Pitman Lee



# BORG



Quelle: Mikano - Borg im Space Park Bremen



„Wir sind die Borg.  
Widerstand ist  
zwecklos.  
Sie werden  
assimiliert.“

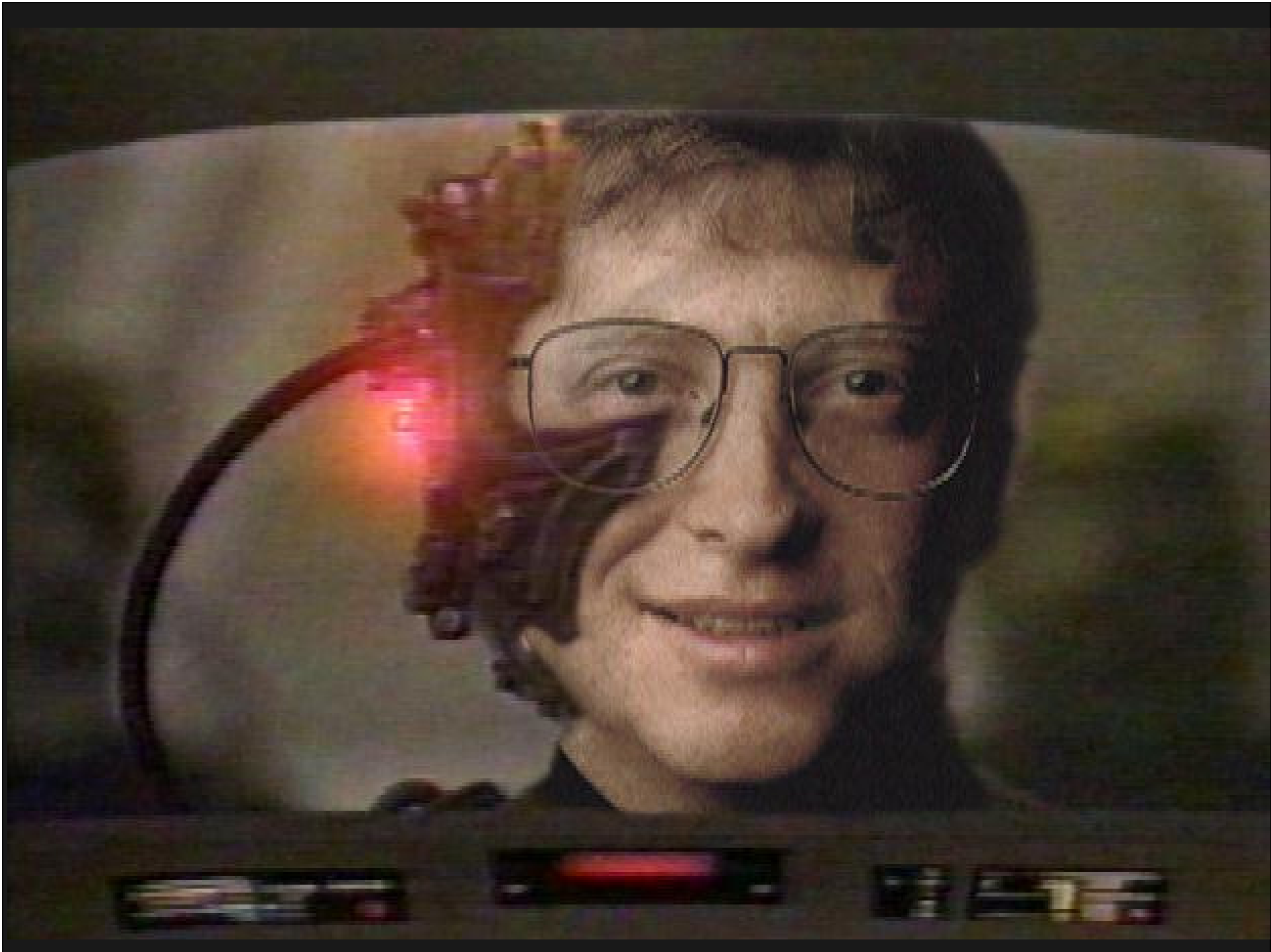
Quelle: Mikano - Borg im Space Park Bremen

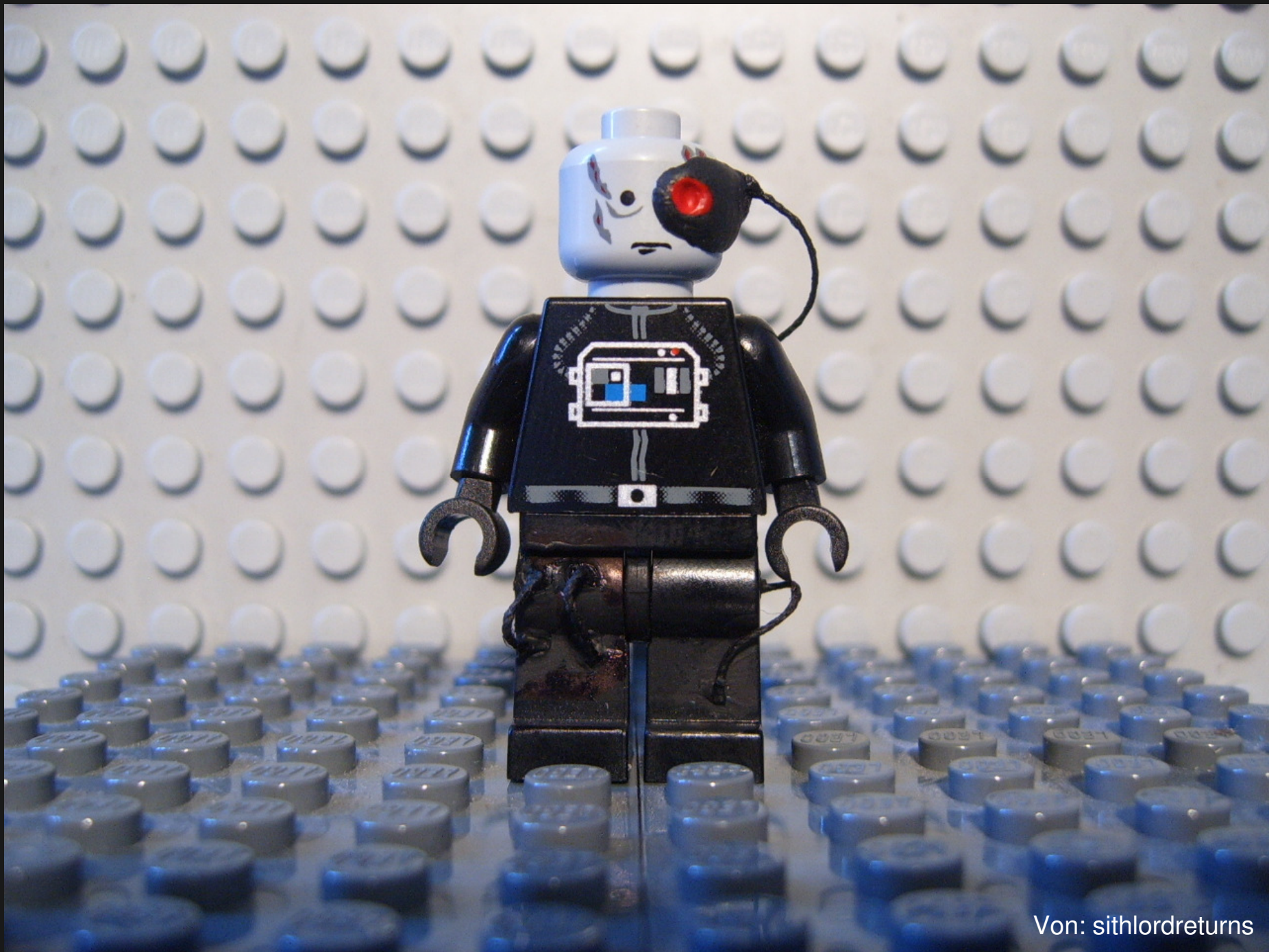
**BORG == Verkomplizierer**



**CNN  
LIVE**

*Oli Volt 4 gimps.de*

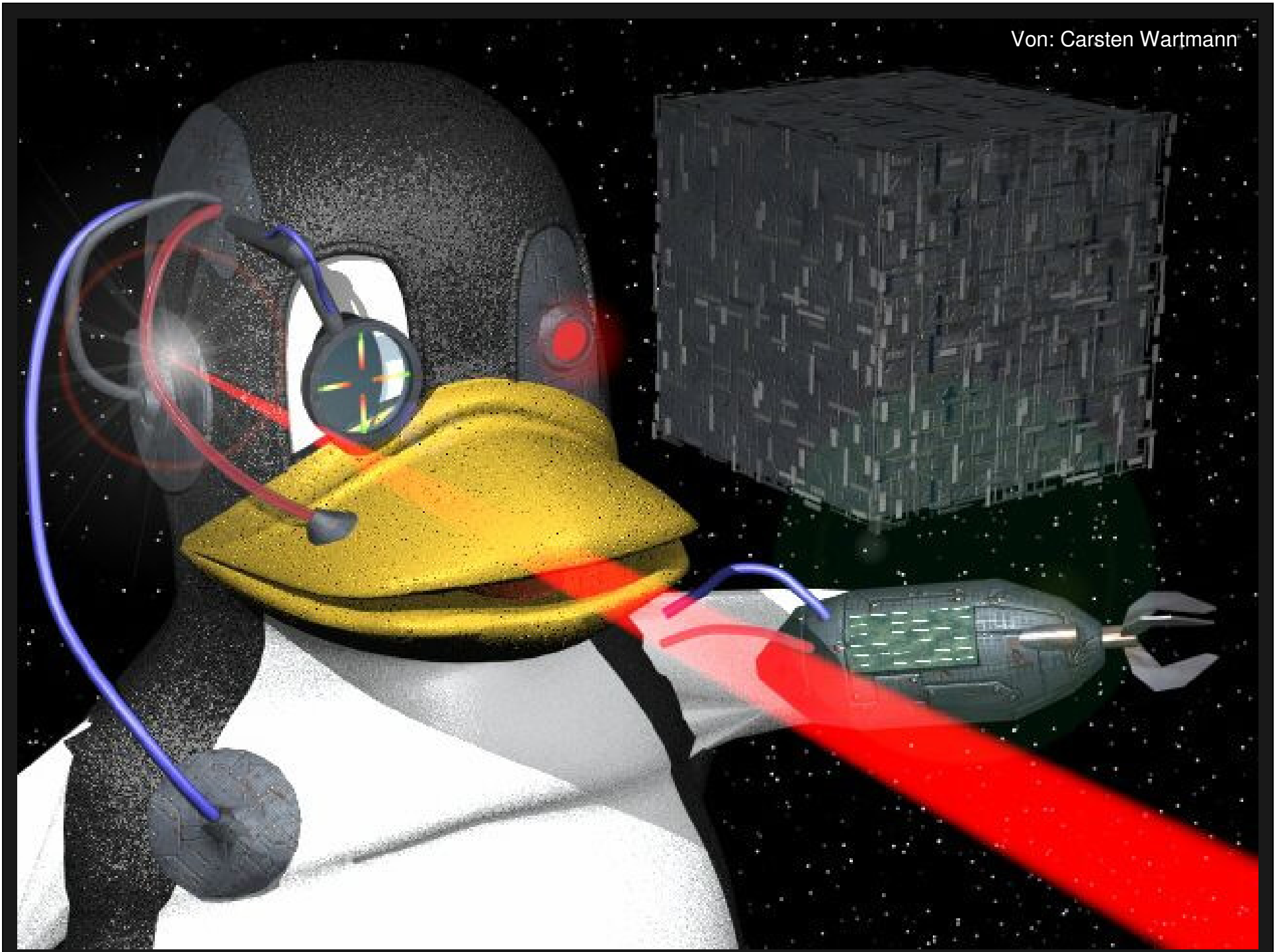




Von: sithlordreturns



Von: Carsten Wartmann





**Warum**

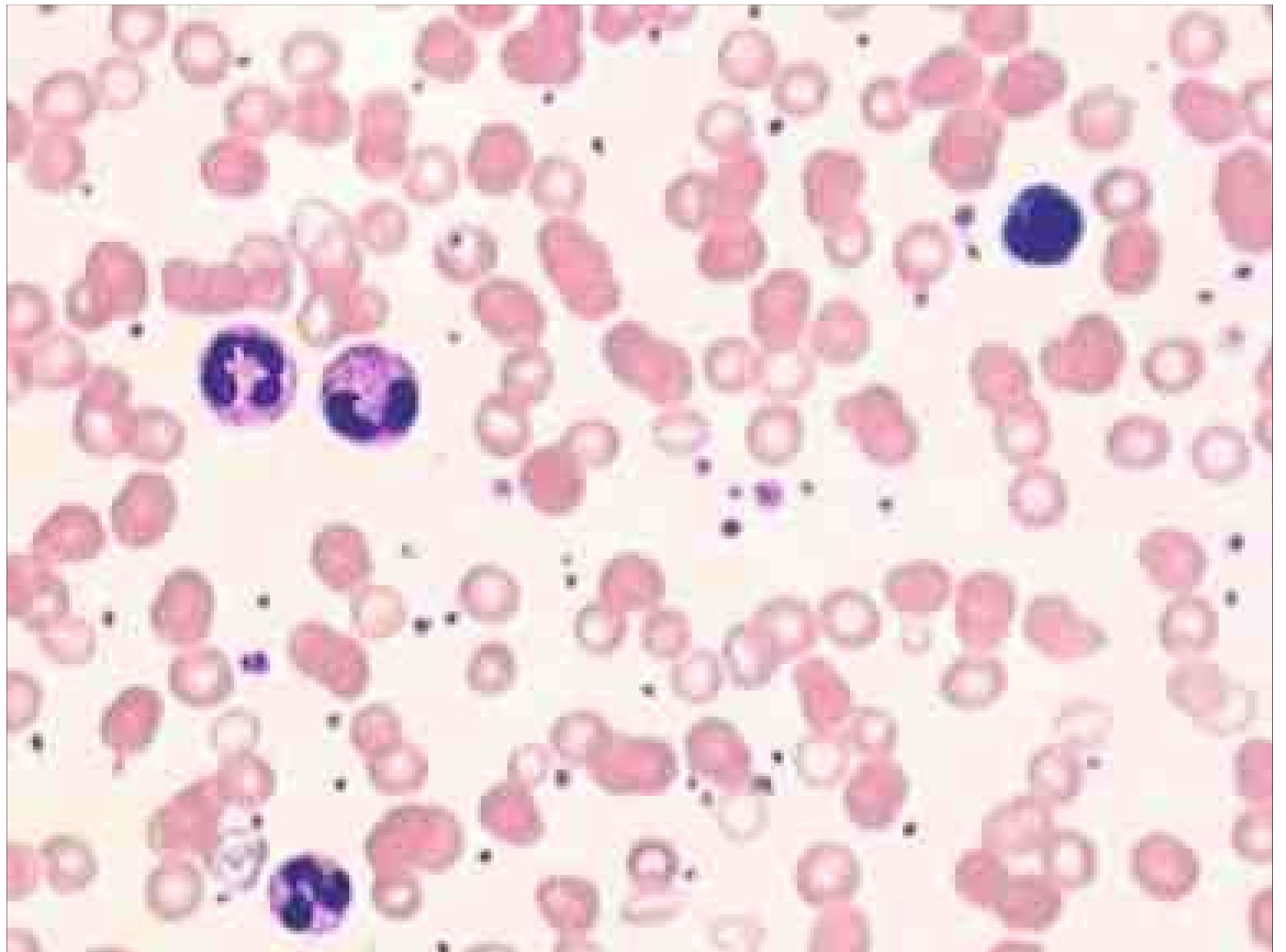
**sind**

**die**

**Borg**

**so**

**erfolgreich?**



Aus der Werbung eines großen deutsche IT-Dienstleiders

„Softwaresysteme sind komplex.

Die Fähigkeit, sie voll und ganz zu durchdenken, ist rar.“

# Borg

Gemeinde Das kommt bestimmt.  
Kreis Das kostet fast nichts.

**Friedliche Borg?  
Schlechtes Zeichen!**



## „Was kann ich tun?“

- Hindernislisten an der Wand
- Kaizen / Babysteps
- Qualitätszyklen
- Wie muss das Problem / der Kontext aussehen, damit die Lösung einfach wird?
- Vertrauen statt Kontrolle
- Sich den Gegenwind um die Nase wehen lassen



Vielen Dank für die Aufmerksamkeit

Noch Fragen?



Schulung *verlängerte Werkbank*  
**agile Softwareentwicklung**  
Festpreisprojekte Coaching  
RCP **Systemintegration** Eclipse  
h3270 Hostintegration  
*Scrum* Refactoring testgetriebene Entwicklung  
Hibernate SAP-Netweaver **OpenSource**  
Ajax JBoss/JEMS Groovy  
*it-agile*  *eXtreme Programming*

## Risiken der Einfachheit

- Probleme werden früh sichtbar.
- Man kann sie in der Regel noch reparieren.
- Der eigene Ruf wird möglicherweise beschädigt.

## Risiken der Einfachheit

- Probleme werden früh sichtbar.
- Man kann sie in der Regel noch reparieren.
- Der eigene Ruf wird möglicherweise beschädigt.

## Risiken der Komplexität

- Probleme werden spät sichtbar.
- Man kann sie häufig nicht mehr reparieren.
- Der eigene Job ist in Gefahr.