Obsoleszenz durch Software
Wie immaterielle Güter Material entwerten

Prof. Dr. Lorenz Hilty
Forschungsgruppe Informatik und Nachhaltigkeit
www.ifi.uzh.ch/isr

Universität Zürich, Institut für Informatik
www.ifi.uzh.ch

Empa, Abteilung Technologie und Gesellschaft
www.empa.ch/tsl

Hauptthesen des Vortrags

1. **Die heutige digitale Technologie bietet riesiges Potenzial für nachhaltiges Wirtschaften, denn**
    1.1. sie kann die Material- und Energieeffizienz vieler Prozesse verbessern („enabling effect“ ↗)
    1.2. sie ist selbst sehr genügsam und wird immer genügsamer („own footprint“ ↘)

2. **Diese Technologie wird so eingesetzt, dass sie dissipative Materialflüsse beschleunigt.**
    2.1. Allgemeine Gründe: Rebound-Effekte, Wachstumszwänge, falsche Preissignale, ...
    2.2. Spezielle Gründe: Geschäftsstrategien der IT-Industrie

Hauptthesen des Vortrags

1. **Die heutige digitale Technologie bietet riesiges Potenzial für nachhaltiges Wirtschaften, denn**

   1.1. sie kann die Material- und Energieeffizienz vieler Prozesse verbessern („enabling effect“ ↗)

   1.2. sie ist selbst sehr genügsam und wird immer genügsamer („own footprint“ ↘)

   – Historische Entwicklung der Energieeffizienz (”Koomey’s Law”)
   – Historische Entwicklung der Materileffizienz (“Moore’s Law”)
   – Software/Hardware-Trennung (A. Turing, J. v. Neumann)

Folien 5-7

2. **Diese Technologie wird so eingesetzt, dass sie dissipative Materialflüsse beschleunigt.**

   2.1. Allgemeine Gründe: Rebound-Effekte, Wachstumszwänge, falsche Preissignale, ...

   2.2. Spezielle Gründe: Geschäftsstrategien der IT-Industrie

   – Software-induzierte Obsoleszenz
   – Programmierte Obsoleszenz
   – Ausnutzung von Informations-Asymmetrien

Folien 9-11

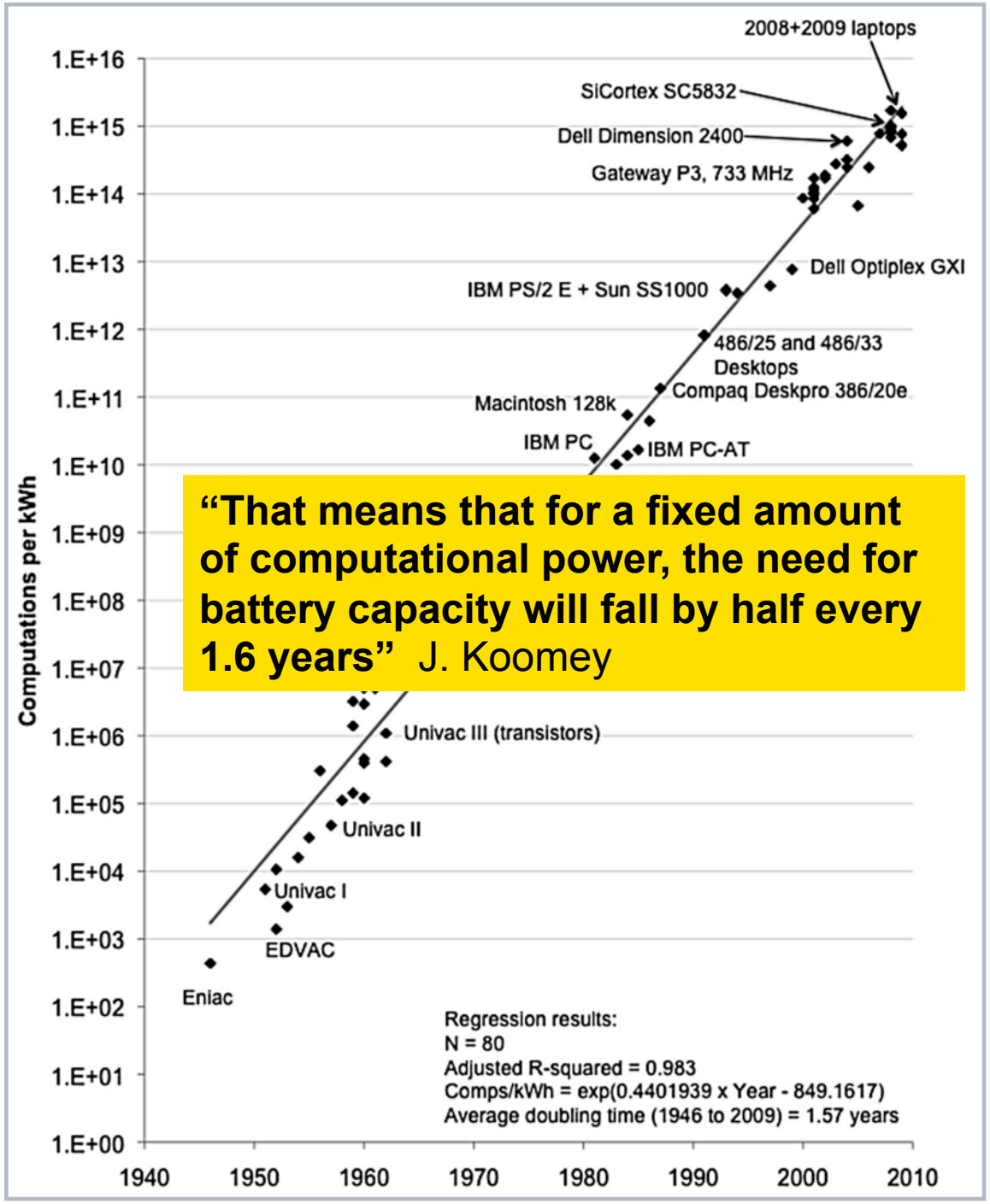
Historische Entwicklung der Energieeffizienz

Rechenoperationen pro Energieeinheit

Verdoppelung der Energieffizienz von Prozessoren durchnittlich alle 1.57 Jahre seit dem ersten programmierbaren elektronischen Computer ENIAC 1946 (“Koomey’s Law”).

ENIAC leistete knapp 1000 Rechenoperationen pro Kilowattstunde (mit Elektronenröhren als Schaltelemente).

Heutige handelsübliche Mikroprozessoren leisten rund **10 Billiarden (10^{16})** Rechenoperationen pro Kilowattstunde.



“That means that for a fixed amount of computational power, the need for battery capacity will fall by half every 1.6 years” J. Koomey

Quelle: Koomey, J., Berard, S., Sanchez, M. & Wong, H. (2011): Implications of Historical Trends in the Electrical Efficiency of Computing. Annals of the History of Computing, IEEE, 33 (3): 46-54

Historische Entwicklung der Materialeffizienz

Transistoren pro Chipfläche

Verdoppelung der Packungsdichte integrierter Schaltungen ca. alle 2 Jahre seit dem ersten Ein-Chip-Prozessor „Intel 4004“ 1971 („Moore‘s Law“).

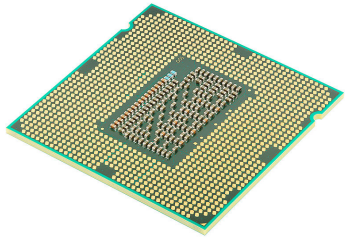
Leistung pro Masse

Bis heute haben Prozessorleistung und Speicherplatz **pro Kilogramm Mikrochips** um einen Faktor von rund 2 Milliarden zugenommen.

1971

Intel 4004
2300
Transistoren

2011



Intel CORE i7 3960X
2,27 Milliarden
Transistoren

Software-Hardware-Trennung

Die Turing-Maschine (TM)

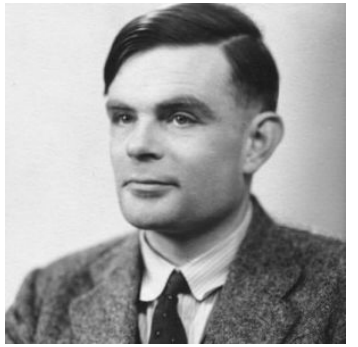
Zu jeder berechenbaren mathematischen Funktion existiert eine TM, die die Funktionsargumente als Input einliest und den Funktionswert als Output ausgibt.

Es gibt eine TM, die jede andere TM simulieren kann. Diese spezielle TM wird „Universelle Turing-Maschine“ (UTM) genannt.

Die UTM bekommt neben den eigentlichen Funktionsargumenten als zusätzlichen Input eine **Beschreibung** derjenigen Turing-Maschine, die sie simulieren soll, um sie auf die Funktionsargumente azuwenden.

→ Wir können mit sprachlichen Ausdrücken beliebige Maschinen erschaffen! (→ Begriff des Programms)

→ John von Neumann schuf eine Architektur, die der UTM für praktische Zwecke sehr nahe kommt (Von-Neumann-Architektur).



Alan Turing (1912-1954)

John von Neumann (1903-1959)

Eine einzige Maschine physisch konstruieren, unendlich viele Maschinen nutzen!

Hauptthesen des Vortrags

1. **Die heutige digitale Technologie bietet riesiges Potenzial für nachhaltiges Wirtschaften, denn**

   1.1. sie kann die Material- und Energieeffizienz vieler Prozesse verbessern („enabling effect“ ↗)

   1.2. sie ist selbst sehr genügsam und wird immer genügsamer („own footprint“ ↘)

   – Historische Entwicklung der Energieeffizienz (”Koomey’s Law”)
   – Historische Entwicklung der Materileffizienz (“Moore’s Law”)
   – Software/Hardware-Trennung (A. Turing, J. v. Neumann)

✔

Folien 5-7

2. **Diese Technologie wird so eingesetzt, dass sie dissipative Materialflüsse beschleunigt.**

   2.1. Allgemeine Gründe: Rebound-Effekte, Wachstumszwänge, falsche Preissignale, ...

   2.2. Spezielle Gründe: Geschäftsstrategien der IT-Industrie

   – Software-induzierte Obsoleszenz
   – Programmierte Obsoleszenz
   – Ausnutzung von Informations-Asymmetrien

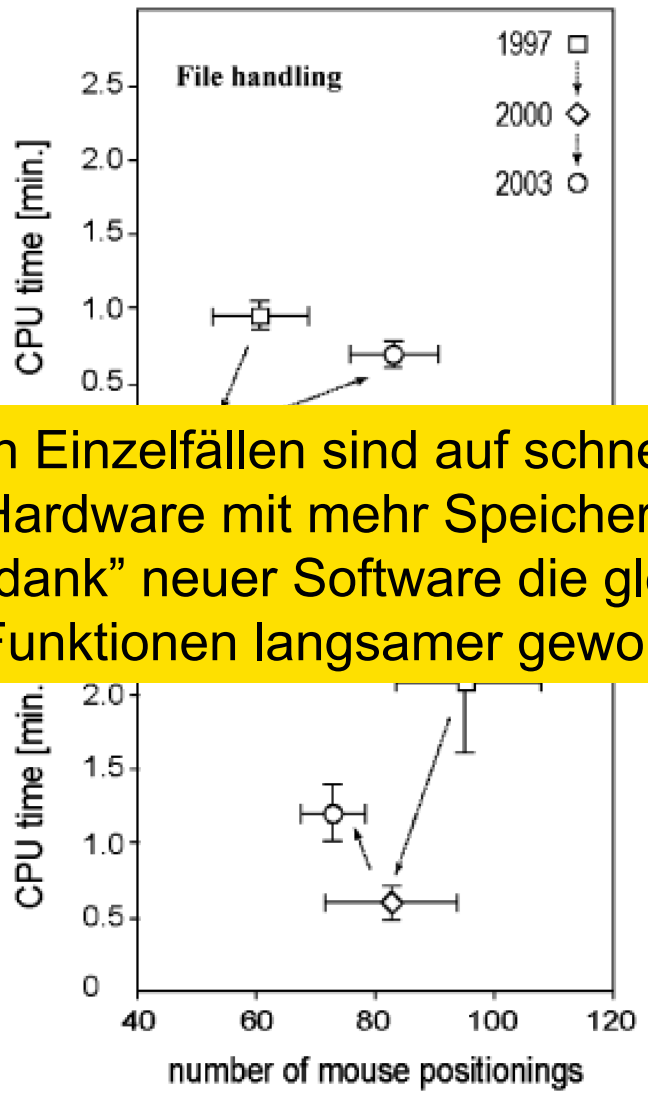
Folien 9-11

Software-induzierte Obsoleszenz

- Software wäre das perfekt nachhaltige Produkt: Software nutzt sich nicht ab, sie ist so immateriell wie ein Roman oder ein Musikstück.
- Auch Hardware nutzt sich nicht ab, wenn sie Software ausführt, sie kann 10 Jahre und länger funktionieren und immer neue Software zum Leben erwecken.

Aber:

- Die Software wird so weiterentwickelt, dass die Ansprüche an Prozessorleistung und Speicherplatz von Version zu Version zunehmen, **insbesondere mit dem Effizienzfortschritt der Hardware Schritt halten.**
- Software-Produkte sind immer fehlerhaft und enthalten u.a. Sicherheitslücken, die nach und nach von Hackern entdeckt werden. → Der Hersteller kann die “Unterstützung” durch Sicherheits-Updates beenden und damit die Außer-betriebnahme der Software (und indirekt von Hardware) damit erzwingen.
- Peripheriegeräte werden obsolet, wenn sie von einer neuen Version des Betriebssystems “nicht mehr unterstützt” werden.



In Einzelfällen sind auf schnellerer Hardware mit mehr Speicherplatz “dank” neuer Software die gleichen Funktionen langsamer geworden.

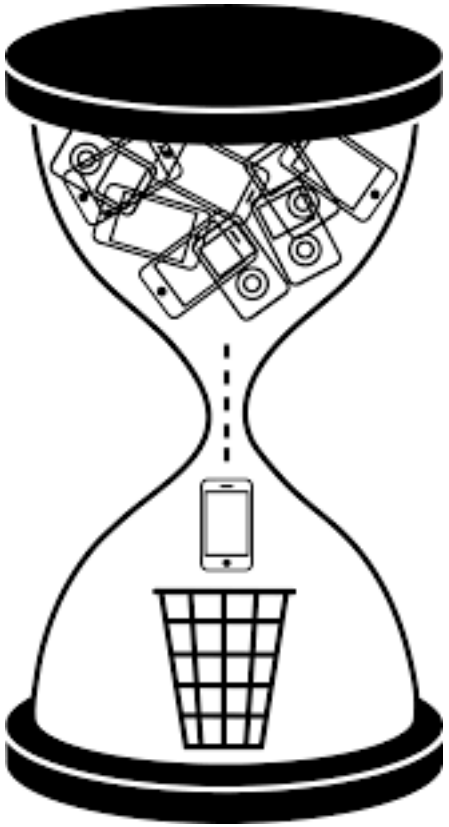
Hilty, L. M.; Köhler, A.; von ...; Ruddy, T.: Rebound Effects of Progress in Information Technology Assessment and Ethics of Science, 1 (4) 20... 007/s10202-005-0011-2

Programmierte Obsoleszenz

Jedes durch Software gesteuerte Gerät bietet Möglichkeiten zur Realisierung geplanter Obsoleszenz durch explizite Anweisungen im Programmcode („programmierte Obsoleszenz“).

- Geräten, die aus Sicherheitsgründen am Tropf von Software-Updates hängen, können jederzeit manipuliert werden, z.B. indem nach einem Update die Leistung gedrosselt wird.

- Weitere bekannte Formen der programmierten Obsoleszenz:
  - Zähler, die nach einer bestimmten Zahl (z.B. von Druckvorgängen oder Akku-Ladezyklen) zu Ausfall oder Qualitätsverschlechterung führen.
  - Hardwarekomponenten (z.B. Ersatzteile) anderer Hersteller werden von der Software erkannt und abgelehnt, ohne dass es dafür technische Gründe gäbe.



Ausnutzung von Informations-Asymmetrien

Von **Informations-Asymmetrie** in einem Markt spricht man dann, wenn die Information über ein Gut zwischen Anbieter- und Nachfragerseite ungleich verteilt ist.

Bei **Software** ist die Informations-Asymmetrie normalerweise besonders hoch, da der Anbieter sehr viel mehr Information über das Produkt besitzt als der Nachfrager. Selbst bei Offenlegung der Quellprogramme sind hoher Aufwand und spezielles Fachwissen erforderlich, um daraus Produkteigenschaften abzuleiten.

Software als Erfahrungs- und Vertrauensgut

– Deshalb fallen Softwareprodukte fallen in die Kategorie der *Vertrauensgüter:* Der Nachfrager hat praktisch keine Möglichkeit, sich im Voraus über die Qualität des Gutes zu informieren, er kann sich lediglich (wie z.B. auch bei der Dienstleistung eines Arztes) auf eigene oder fremde Erfahrung verlassen und muss dem Anbieter in einem gewissen Ausmaß vertrauen.

– Der Anbieter kann die Orientierungslosigkeit des Kunden z.B. ausnutzen, um ihn an sich zu binden („user lock-in“) und zu einem fortgesetzten Konsum der von ihm angebotenen Güter und Dienstleistungen zu nötigen oder zumindest zu „nudgen“.

Hauptthesen des Vortrags

1. **Die heutige digitale Technologie bietet riesiges Potenzial für nachhaltiges Wirtschaften, denn**

   1.1. sie kann die Material- und Energieeffizienz vieler Prozesse verbessern („enabling effect“ ↗)

   1.2. sie ist selbst sehr genügsam und wird immer genügsamer („own footprint“ ↘)

   – Historische Entwicklung der Energieeffizienz (”Koomey’s Law”)
   – Historische Entwicklung der Materileffizienz (“Moore’s Law”)
   – Software/Hardware-Trennung (A. Turing, J. v. Neumann)

✔

Folien 5-7

2. **Diese Technologie wird so eingesetzt, dass sie dissipative Materialflüsse beschleunigt.**

   2.1. Allgemeine Gründe: Rebound-Effekte, Wachstumszwänge, falsche Preissignale, ...

   2.2. Spezielle Gründe: Geschäftsstrategien der IT-Industrie

   – Software-induzierte Obsoleszenz
   – Programmierte Obsoleszenz
   – Ausnutzung von Informations-Asymmetrien

✔

Folien 9-11

Pessimistischer Ausblick: Das Problem könnte sich ausweiten

Smarte Dinge / Internet der Dinge

– Einer der Haupttrends der Digitalisierung besteht darin, dass mehr und mehr Dinge des Alltags (nicht nur IT-Geräte) durch eingebettete Prozessoren **“smart”**, also durch Software gesteuert und häufig auch internetfähig werden.

– Dies betrifft z.B. Haushaltgeräte, Spielsachen, Wohnungseinrichtung (“Internet der Dinge”, “Cyber-Physical Systems”).

– Alle diese Dinge sind damit potenziell von Obsoleszenz durch Software betroffen.

Konsum wird steuerbar – ohne Umweg über Werbung

– Die Externalisierung der Kontrolle über viele Dinge des Alltags kann zu einem generellen Instrument der Konsumsteuerung werden.

Optimistischer Ausblick: Lösungsansätze

Recht auf Reparatur

– Die EU bereitet ein „Recht auf Reparatur“ für Elektrogeräte vor: Reparaturen müssen mit allgemein zugänglichen Werkzeugen ausführbar sein; Reparaturanleitungen müssen offen zugänglich sein.

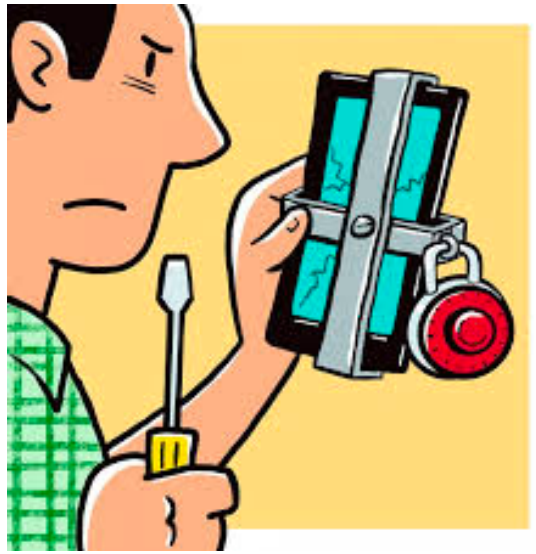
– In 16 US-Bundesstaaten gibt es ein solches „Right to repair“ bereits für Autos. Kern ist die **Gleichstellung der Reparateure**: keine „autorisierten Werkstätten“; jeder, der das Wissen und die Fertigkeiten hat, kann ein Auto reparieren.

– Würde man das konsequent für IT-Geräte umsetzen, müsste der Quellcode offengelegt werden, denn die Software ist ja die eigentliche Maschine. (Diese Konsequenz ist aber eher nicht zu erwarten.)

Materielle Selbstbestimmung

– Ist es akzeptabel, dass Sachgüter, die jemand als Eigentum erworben hat, dauerhaft unter der Kontrolle des Herstellers bleiben und von diesem jederzeit entwertet werden können?

– Analog zum Recht auf **informationelle** Selbstbestimmung, das aus dem Persönlichkeitsrecht heraus entwikcelt wurde, könnte ein „Recht auf **materielle** Selbstbestimmung“ aus dem Eigentumsrecht entwickelt werden.



Texte zum Vortrag

Zusammengestellt auf

https://files.ifi.uzh.ch/hilty/tutzing

Shortlink: http://tiny.uzh.ch/Zy

Kontakt:

Prof. Dr. Lorenz M.Hilty
Universität Zürich, IFI
Binzmühlestrasse 14
CH-8050 Zürich

hilty@ifi.uzh.ch