

Green Coding @ PTA

Impulsvortrag Smart Production Network

Dr. Frank Gredel, PTA GmbH

28. Juni 2022

01 Motivation / Warum Green Coding?

02 Wo kann man ansetzen?

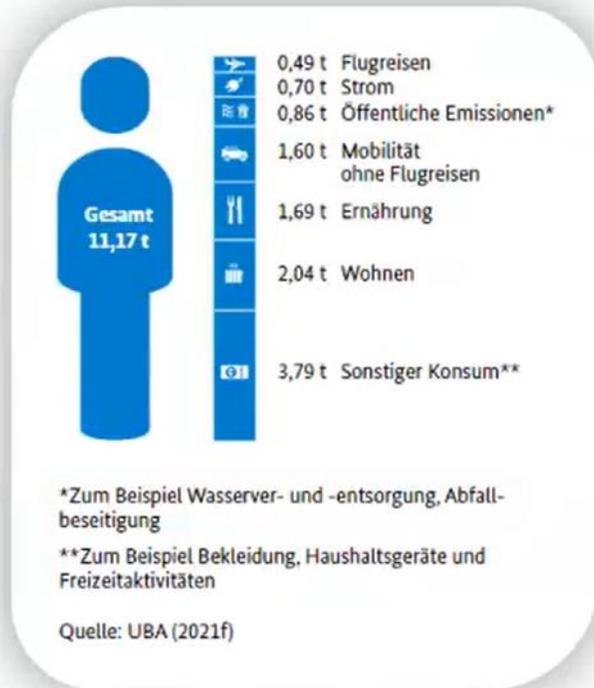
03 Fazit



zum Einordnen ...

| Typ | CO ₂ Emission | Bäume / Jahr | CO ₂ Emission / Jahr |
|--|-----------------------------------|--------------|--|
| Baum | 10 kg/Baum | 1 | 1 Baum bindet ~10 kg/Jahr (alters- und ortabhängig) |
| Barbecue mit 8 Personen | 18 kg/Abend | 9 | 5 Abende/Jahr= 90 kg |
| Energie | Kohle 1000 g/kWh Mix 401 g/kWh | 350 140 | 3 Personen Haushalt (3500 kWh/Jahr) = 3.5 t 3 Personen Haushalt (3500 kWh/Jahr) = 1.4 t |
| Laptop 8h (138 Watt/h) | 442 g/Arbeitstag | 8 | 200 Arbeitstage im Jahr = 88 kg |
| Streaming | 100 g/h | 7 | 2h/Tag x 365 = 73 kg |
| google.com | 0.2 g/Click | 41Mio | 5,6 Mrd. Suchen/Tag = 1.400 t/Tag = 408.800 t/Jahr |
| Training AI Netzwerk (GPT-3,175b params) | 85t/Training | 8500 | |
| Bitcoin Mining 2019 | 22m t/Jahr | 2.2 Mrd | |
| Schwarzwald | Anzahl der Bäume | 4 Mio | |

... digitale Emissionen pro Person ...



Digitale Emissionen einer Person (Deutschland)

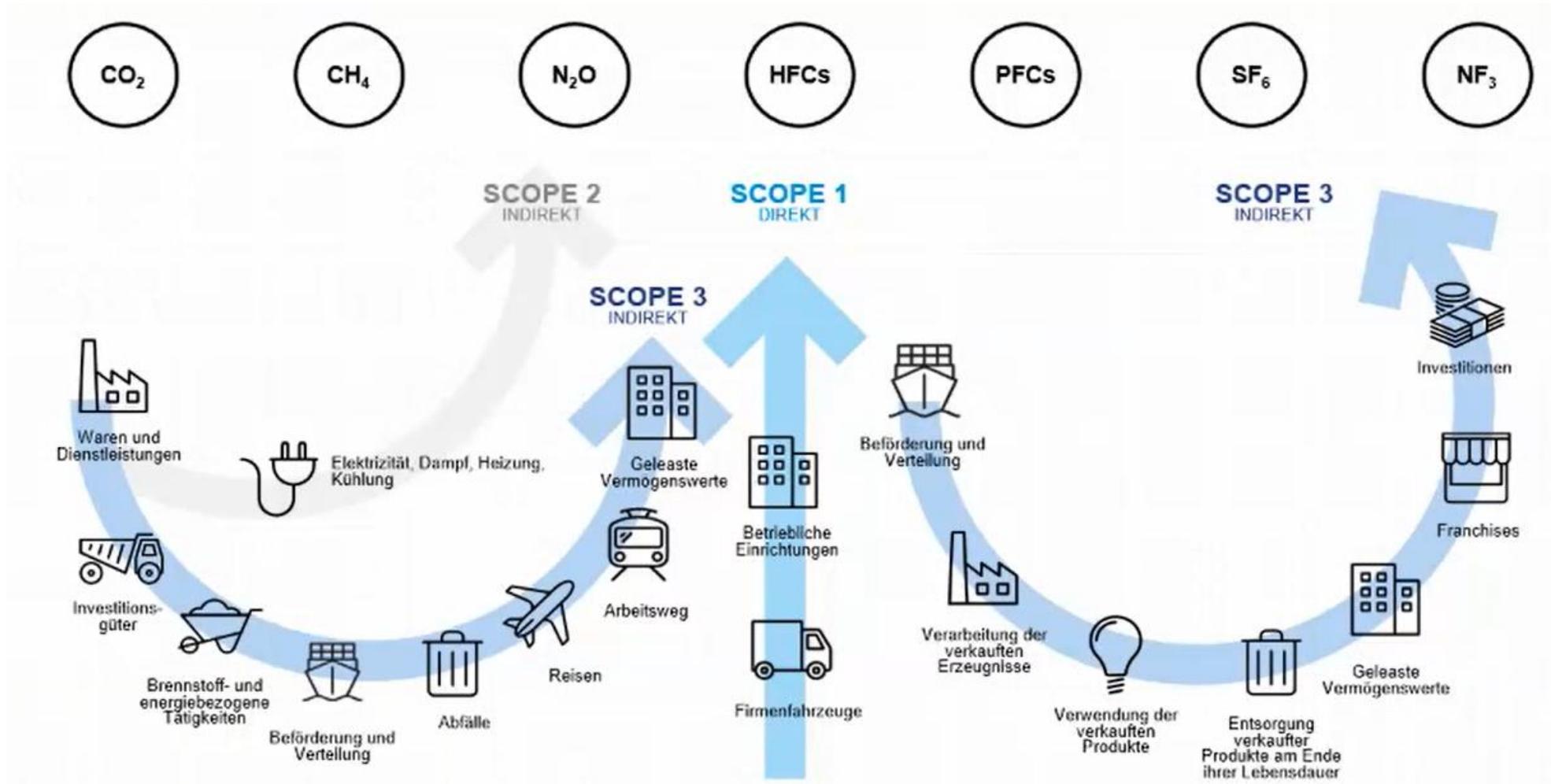
$$11,17t \times 4\% = 0,44 t / \text{Jahr} \\ = 1,2 \text{ kg} / \text{Tag}$$

| Aktivität | CO ₂ e |
|-------------------------|---------------------------|
| Atmen für einen Tag | <u>1-2kg</u> |
| 10 Km Auto fahren | <u>2kg</u> |
| Rose zum Valentinstag | <u>350g</u> |
| Produktion 150g Steak | <u>5,4kg</u> |
| Flug München – New York | <u>661kg</u> (pro Person) |



Der IT-Sektor ist verantwortlich für 4% der globalen Treibhausgas Emissionen

Skalierungseffekte und Scope 3 Emissionen



Beispiel für Scope 3 Emission in der IT

Microsoft's expected carbon emissions in 2020



Microsoft's expected 16 million metric tons of carbon emissions in 2020, around 75% were in Scope 3.

Given the wide range of scope 3 activities, this higher percentage of the total is probably typical for most IT organizations.



01

Motivation

KLIMAWANDEL

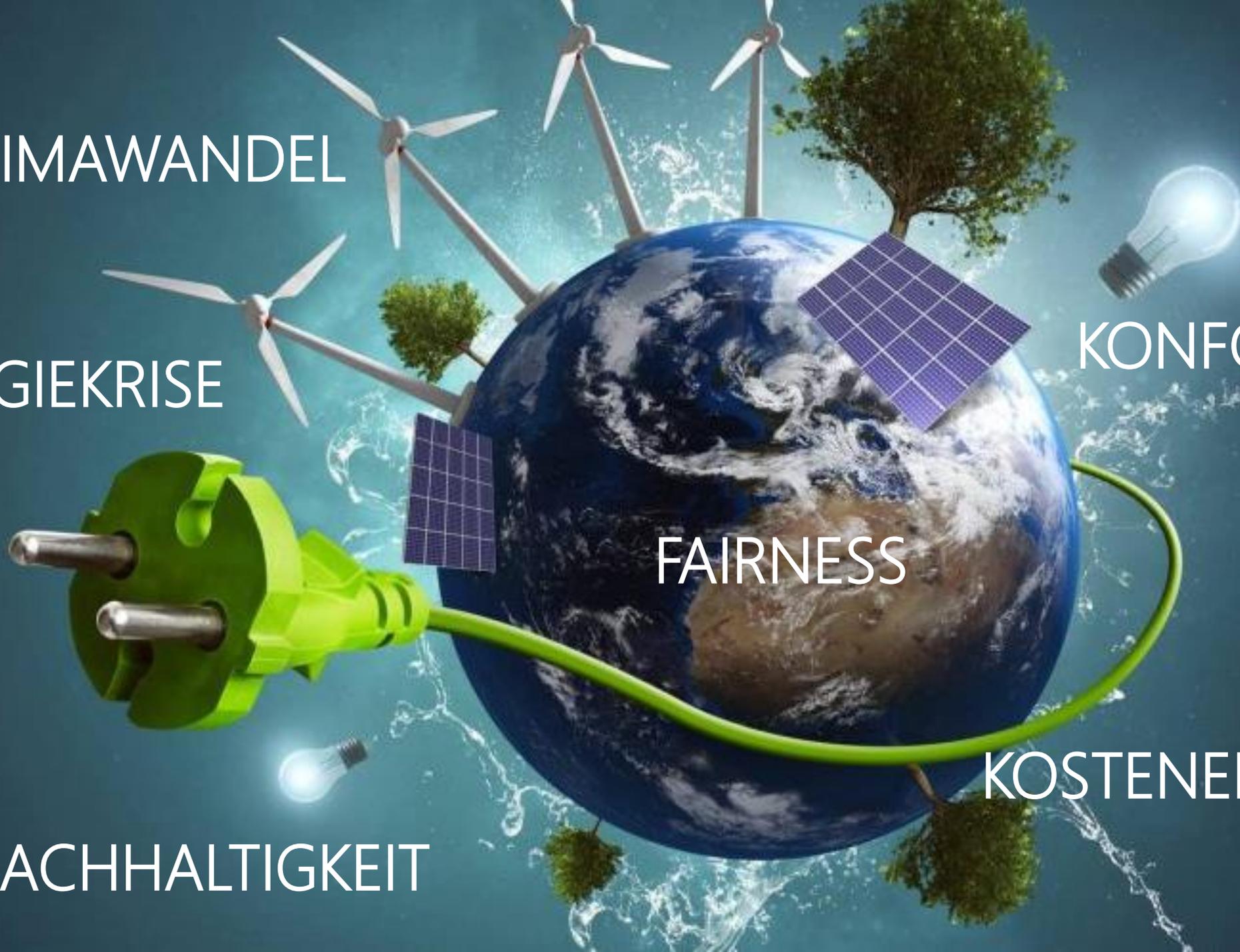
ENERGIEKRISE

KONFORMITÄT

FAIRNESS

KOSTENEFFIZIENZ

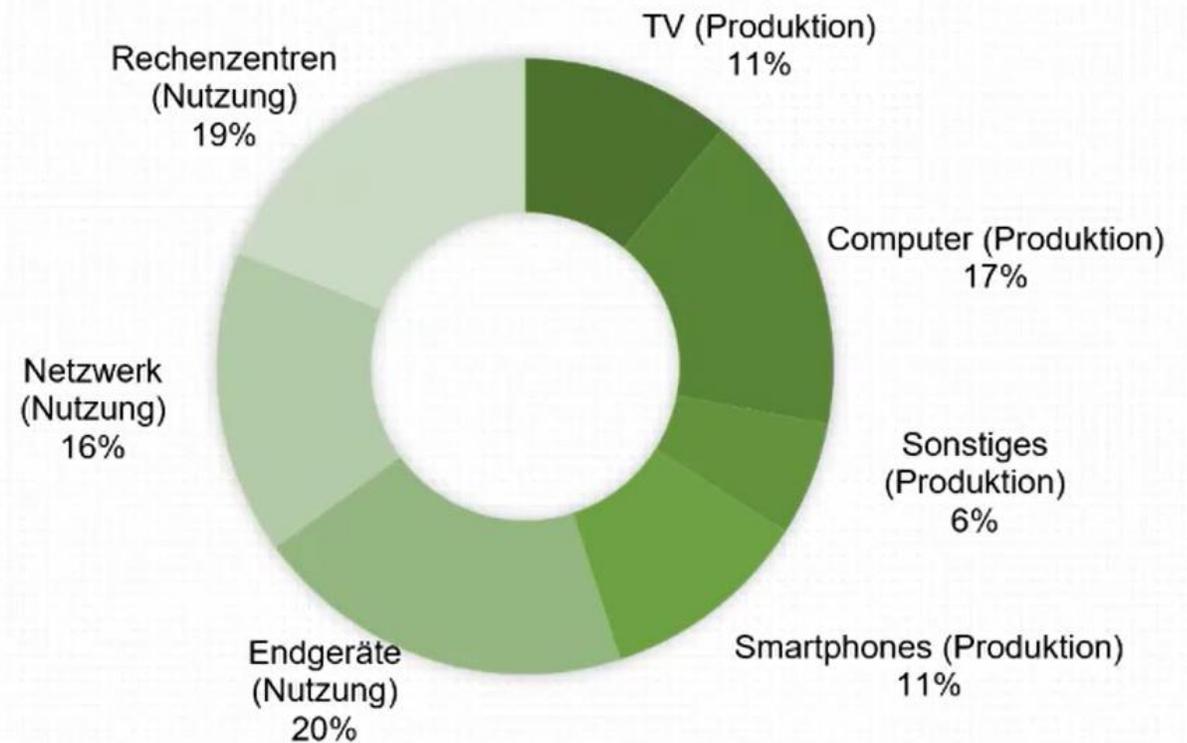
NACHHALTIGKEIT



Was ist Green Coding?

- **GreenIT** ist kein neuer Begriff und konzentriert sich primär auf die Hardware und die Optimierung von Rechenzentren – heute häufig in der Cloud
- Das ist jedoch nicht genug ...
- **55%** der Emissionen wird durch die zugrunde liegende Software beeinflusst

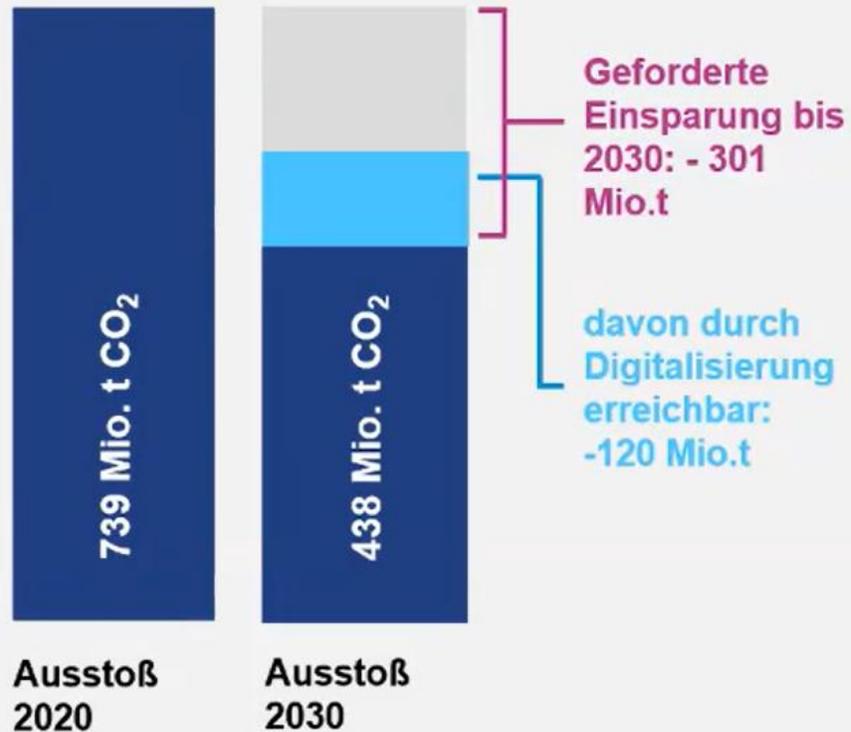
➔ **GREEN CODING**



Source: https://theshiftpoint.org/wp-content/uploads/2019/03/Executive-Summary_Lean-ICT-Report_EN_lowdef.pdf

Beitrag und Kosten der Digitalisierung

Das CO₂-Einsparungspotenzial der Digitalisierung



Die CO₂-Kosten der Digitalisierung

Digitale Emissionen

- sind 2021 für 4% der globalen Emissionen verantwortlich
- **steigen** um 9% jährlich
- werden 2040 14% der **weltweiten** gesamt Emissionen ausmachen
- werden bis 2030 6% der deutschen Emissionen ausmachen, so viel wie heute die **Stahlindustrie**



02

Wo kann man ansetzen?

Wo setzt Green Coding an?

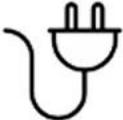
Reduce emissions

 Green energy

 Green industry & agriculture

 Green transport

Reduce consumption

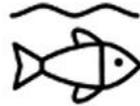
 Energy consumption

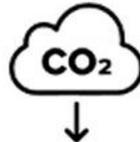
 Goods consumption

 Transport consumption (esp. goods)

Conversion & storage

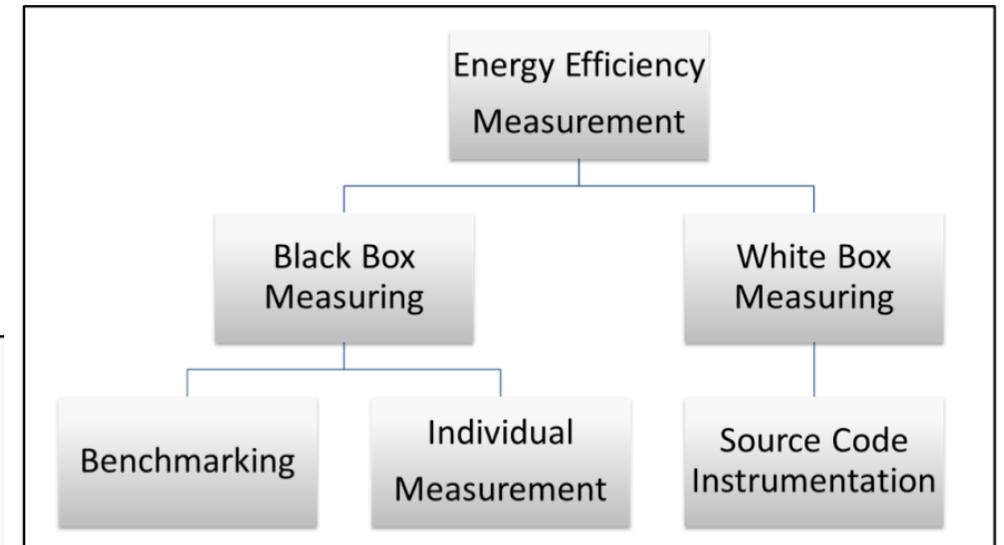
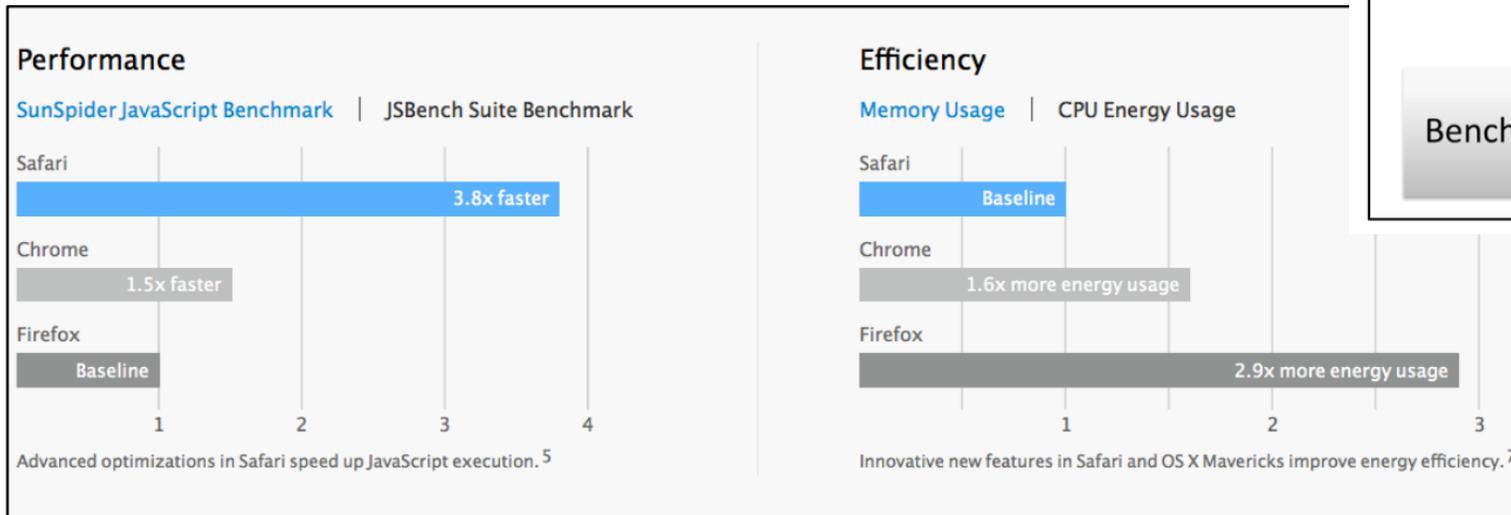
 Photosynthesis
Trees & phytoplankton

 CO₂ storages
oceans & whales,
wetlands, marshlands

 CO₂e removal
technologies

Herausforderung: Messbarkeit

Energy Efficiency = Useful Work Done / Used Energy



Green Coding Ansätze

Green Plattform

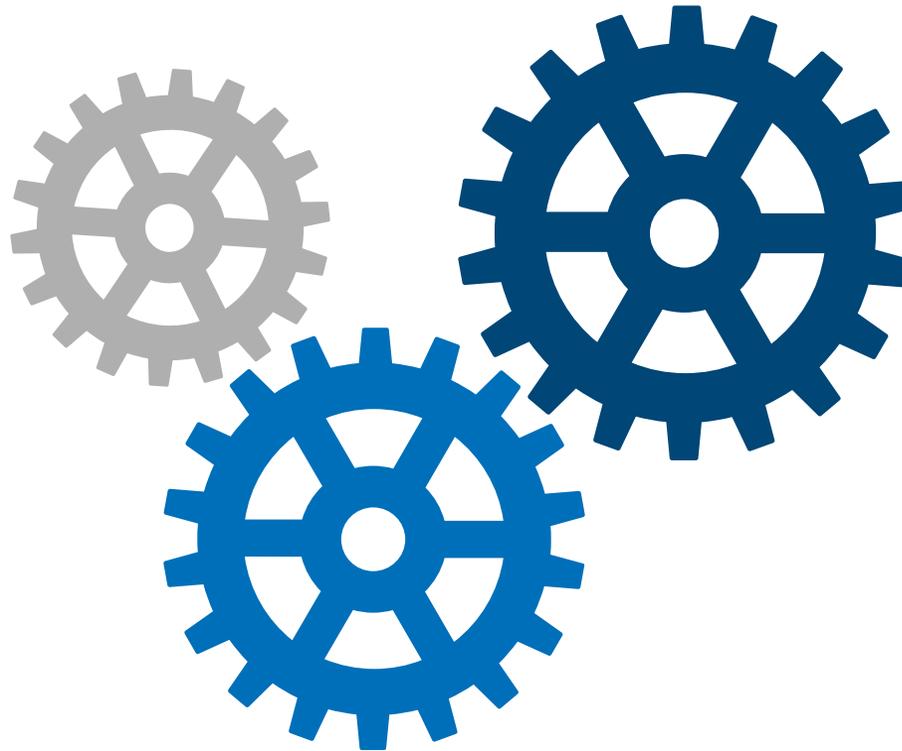
Optimale Auslastung

- Oft sind die Systeme zu groß definiert
- Falsche Konfiguration vermeiden
- Versteckte Infrastruktur bedenken

Green Methodik

Agiles Vorgehen

- Feedback berücksichtigen
- Wiederverwendbare Ergebnisse



Green Logik

Best Practice Leitlinien mit Energiesparkriterien erweitern

- Nutzerorientierter visueller Inhalt
- Zero Waste Code
- Ressourcen mit geringem CO₂-Fußabdruck
- Nutzungshäufigkeit und Nähe.

<https://www.gft.com/de/de/technology/greencoding>

Green Coding Ansätze

Green Plattform

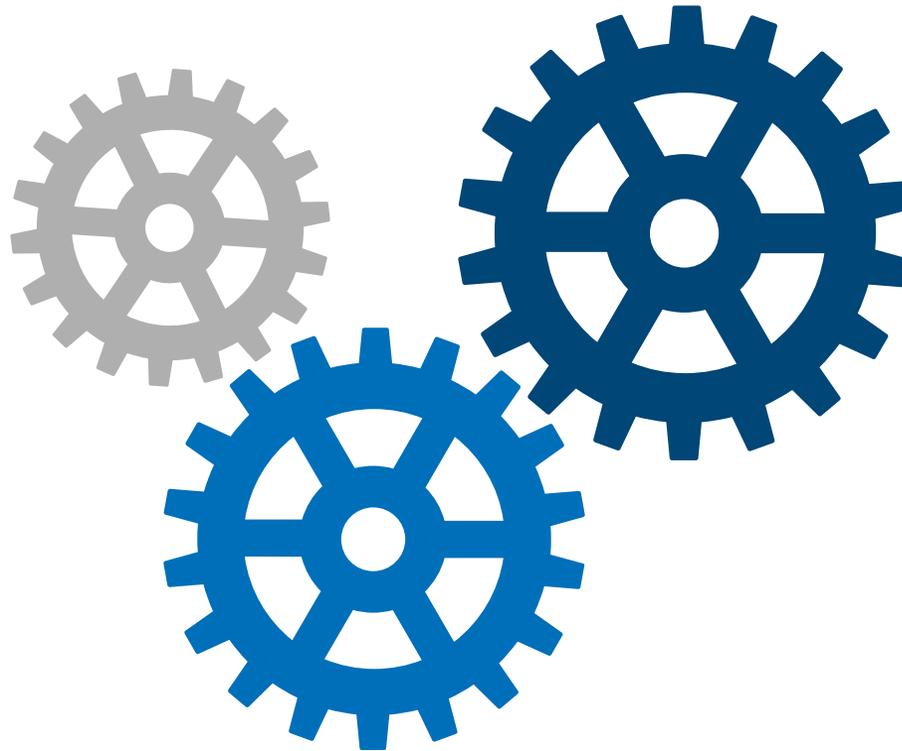
Optimale Auslastung

- Oft sind die Systeme zu groß definiert
- Falsche Konfiguration vermeiden
- Versteckte Infrastruktur bedenken

Green Methodik

Agiles Vorgehen

- Feedback berücksichtigen
- Wiederverwendbare Ergebnisse



Green Logik

Best Practice Leitlinien mit
Energiesparkriterien erweitern

- Nutzerorientierter visueller Inhalt
- Zero Waste Code
- Ressourcen mit geringem CO2-Fußabdruck
- Nutzungshäufigkeit und Nähe.

<https://www.gft.com/de/de/technology/greencoding>

Beispiel 1 Programmiersprachen haben bereits einen großen Einfluss auf Energieeffizienz und Geschwindigkeit

| ENERGY | | TIME | | MB | |
|----------------|-------|----------------|-------|----------------|-------|
| (c) C | 1.00 | (c) C | 1.00 | (c) Pascal | 1.00 |
| (c) Rust | 1.03 | (c) Rust | 1.04 | (c) Go | 1.05 |
| (c) C++ | 1.34 | (c) C++ | 1.56 | (c) C | 1.17 |
| (c) Ada | 1.70 | (c) Ada | 1.85 | (c) Fortran | 1.24 |
| (v) Java | 1.98 | (v) Java | 1.89 | (c) C++ | 1.34 |
| (c) Pascal | 2.14 | (c) Chapel | 2.14 | (c) Ada | 1.47 |
| (c) Chapel | 2.18 | (c) Go | 2.83 | (c) Rust | 1.54 |
| (v) Lisp | 2.27 | (c) Pascal | 3.02 | (v) Lisp | 1.92 |
| (c) Ocaml | 2.40 | (c) Ocaml | 3.09 | (c) Haskell | 2.45 |
| (c) Fortran | 2.52 | (v) C# | 3.14 | (i) PHP | 2.57 |
| (c) Swift | 2.79 | (v) Lisp | 3.40 | (c) Swift | 2.71 |
| (c) Haskell | 3.10 | (c) Haskell | 3.55 | (i) Python | 2.80 |
| (v) C# | 3.14 | (c) Swift | 4.20 | (c) Ocaml | 2.82 |
| (c) Go | 3.23 | (c) Fortran | 4.20 | (v) C# | 2.85 |
| (i) Dart | 3.83 | (v) F# | 6.30 | (i) Hack | 3.34 |
| (v) F# | 4.13 | (i) JavaScript | 6.52 | (v) Racket | 3.52 |
| (i) JavaScript | 4.45 | (i) Dart | 6.67 | (i) Ruby | 3.97 |
| (v) Racket | 7.91 | (v) Racket | 11.27 | (c) Chapel | 4.00 |
| (i) TypeScript | 21.50 | (i) Hack | 26.99 | (v) F# | 4.25 |
| (i) Hack | 24.02 | (i) PHP | 27.64 | (i) JavaScript | 4.59 |
| (i) PHP | 29.30 | (v) Erlang | 36.71 | (i) TypeScript | 4.69 |
| (v) Erlang | 42.23 | (i) Jruby | 43.44 | (v) Java | 6.01 |
| (i) Lua | 45.98 | (i) TypeScript | 46.20 | (i) Perl | 6.62 |
| (i) Jruby | 46.54 | (i) Ruby | 59.34 | (i) Lua | 6.72 |
| (i) Ruby | 69.91 | (i) Perl | 65.79 | (v) Erlang | 7.20 |
| (i) Python | 75.88 | (i) Python | 71.90 | (i) Dart | 8.64 |
| (i) Perl | 79.58 | (i) Lua | 82.91 | (i) Jruby | 19.84 |

Figure 2:
The energy, time and memory required by different languages for a benchmark problem^a

^aEnergy efficiency by programming language: <https://greenlab.di.uminho.pt/wp-content/uploads/2017/10/sleFinal.pdf>, October 2017

Beispiel 2- Energieoptimierung Datenbanken

- Einfluss des Arbeitsspeichers zum Stromverbrauch => abhängig von Verhältnis Größe DB zu RAM
- Einfluss der Datenbankgröße zum Stromverbrauch => linear
- Einfluss von Indizes auf Stromverbrauch => nicht zwangsweise positiver Effekt; gut, wenn Ausschluss von Datensätzen möglich
- Einsatz von SSDs => größere Blockgröße führt zur Reduktion des Stromverbrauchs; ideal mit FlashDB

<http://www.odpms.org/wp-content/uploads/2013/11/GreenITandDB.pdf>

| Cache Green Checklist | | |
|--------------------------|--|---|
| | Energieoptimierung | Auswirkung |
| Arbeitsspeicher | Erweiterung des Arbeitsspeichers um mindestens ~3 GB. | Abhängig vom Abfrage-Typ werden bis zu 53 % Energie gespart. Empfehlenswert bei Sortierungen und Verknüpfungen (JOINS). |
| Indexstrukturen | Anlegen eines Index bei Range-Bedingungen (bei niedriger Selektivität der Spalte der Range-Bedingung). | Spart bis zu 82 % bei stark einschränkenden Range-Bedingungen. |
| Solid-State-Drive | Verwendung einer SSD statt einer HDD. | Spart bis zu 18 % Energie und 17 % Laufzeit. |
| | Bei Nutzung einer SSD die Datenbank-Blockgröße auf 64 KB einstellen (bei OLAP-Workload). | Spart zusätzlich 7 % Energie und 10 % Laufzeit. |
| Prozessor-Nutzung | Dynamische Spannung- und Frequenzanpassung des Prozessors (z.B. <i>Enhanced Halt State</i> -Funktion) im BIOS aktivieren | Spart durchschnittlich 3 % Energie. |
| | <i>Enhanced Halt State</i> -Funktion im BIOS aktivieren für I/O-lastige Abfragen. | Spart bis zu 9 % Energie. Je niedriger die Prozessorauslastung desto größer ist die Energieeinsparung. |

Green Coding Ansätze

Green Plattform

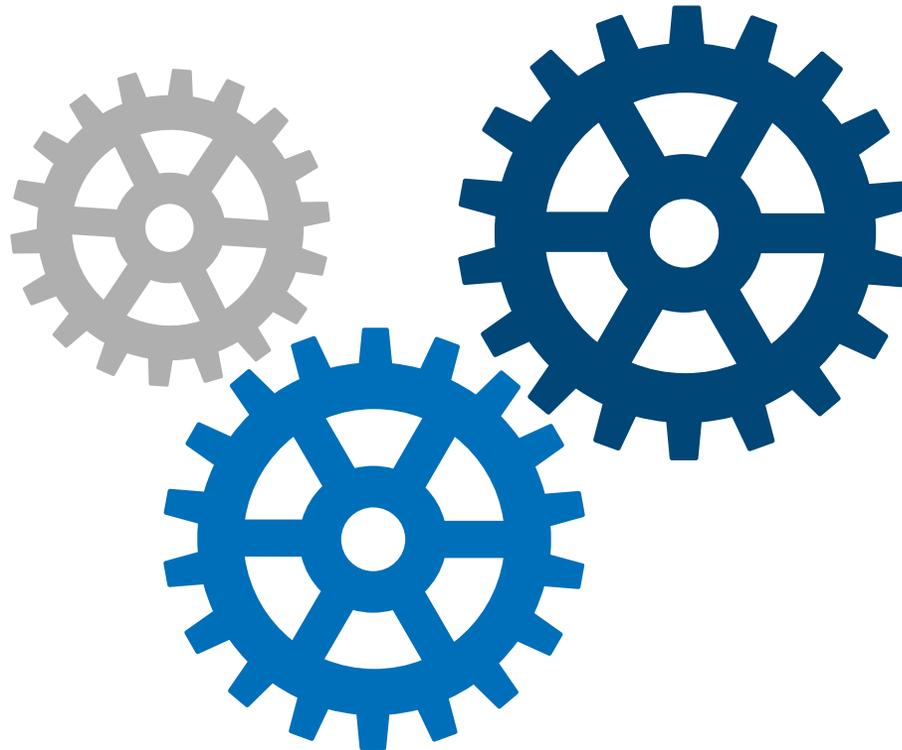
Optimale Auslastung

- Oft sind die Systeme zu groß definiert
- Falsche Konfiguration vermeiden
- Versteckte Infrastruktur bedenken

Green Methodik

Agiles Vorgehen

- Feedback berücksichtigen
- Wiederverwendbare Ergebnisse



Green Logik

Best Practice Leitlinien mit Energiesparkriterien erweitern

- Nutzerorientierter visueller Inhalt
- Zero Waste Code
- Ressourcen mit geringem CO2-Fußabdruck
- Nutzungshäufigkeit und Nähe
- Komprimierung

<https://www.gft.com/de/de/technology/greencoding>

Beispiel – Einfluss der Programmierung

- Die Programmierung kann – bei identischem Ergebnis - Lasten reduzieren
- Im Beispiel reduziert der Code Aufrufe/Aktivitäten der CPU/GPU – Reduktion des Energieverbrauchs der CPU sowie der Grafikkarte

N Renderings vs. 1 Rendering



```
/// <summary>
/// Adds the additional data to this collection.
/// </summary>
/// <remarks>
/// Caution: If data is bound to the user interface, each addition leads to additional effort for re-rendering.
/// This leads to lower performance and higher power consumption.
/// </remarks>
/// <param name="additionalData">Additional data</param>
0 references | 0 changes | 0 authors, 0 changes
private void AddData(IEnumerable<T> additionalData)
{
    foreach (T item in additionalData)
    {
        base.Add(item);
    }
}
```

- Für jedes neue Element
- UI-Measurement (CPU)
 - Re-Rendering (CPU/GPU)

```
/// <summary>
/// Adds the dataList to this collection without individual add notifications.
/// </summary>
/// <remarks>
/// The suppression of the individual messages leads to better performance and lower energy consumption.
/// </remarks>
/// <param name="dataList">The data list to be added.</param>
77 references | Marc Ambruster, 25 days ago | 1 author, 1 change | 1 incoming change | 1 work item
public void AddRange(IEnumerable<T> dataList)
{
    this.SuppressSingleAddNotification = true;

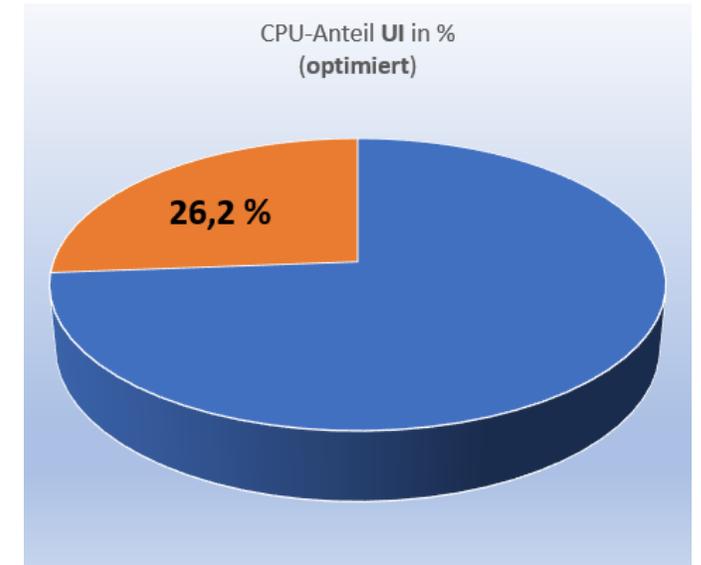
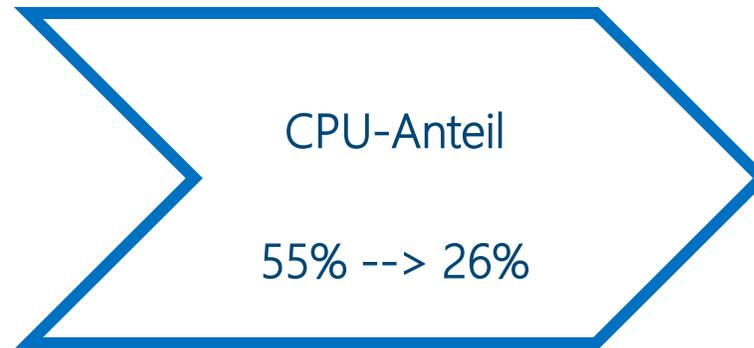
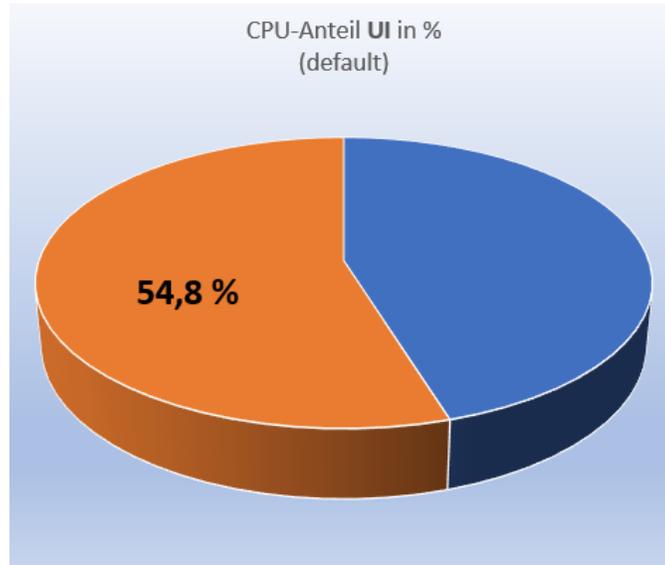
    // add items without notifying -> reduce UI effort
    foreach (T item in dataList)
    {
        base.Add(item);
    }

    this.SuppressSingleAddNotification = false;

    // notify collection changed only at the end of the action
    this.OnCollectionChanged(new NotifyCollectionChangedEventArgs(NotifyCollectionChangedAction.Reset));
}
```

- Measurement und Rendering unterdrücken und erst am Ende einmal aufrufen.
Nur 1x
- UI-Measurement (CPU)
 - Re-Rendering (CPU/GPU)

Beispiel – Einfluss der Programmierung



Beispiel - PTA EnergyBenchmark

```
·const int primCounter = 300_000;  
·EnergyBenchmarkResult start = EnergyBenchmark.Start() 1  
·Console.WriteLine(start.ToString());  
·Console.WriteLine($"Erstelle '{primCounter}' Primzahlen");  
·ArrayList result = GeneratePrimesNumber(primCounter);  
·EnergyBenchmarkResult end = EnergyBenchmark.Stop() 2  
·Console.WriteLine(end.ToString());  
·double restMW = start.RemainingCapacity - end.RemainingCapacity;  
·double spannungsDiff = start.Voltage - end.Voltage;  
·Console.WriteLine($"Verbrauchte Kapazität: {restMW} mw"); 3  
·Console.WriteLine($"Differenz Spannung: {spannungsDiff} V");  
·Console.WriteLine($"Restkapazität: {result.EstimatedChargeRemaining}");  
·Console.WriteLine($"Ausführungszeit: {result.ElapsedTime}");  
·Console.ReadKey();
```

```
Entladerate: 20353MW; Restkapazität: 19502MW; 14,439V; 29%  
Erstelle '300000' Primzahlen  
Entladerate: 33592MW; Restkapazität: 16796MW; 14,046V; 25%  
Verbrauchte Kapazität: 2706 mw  
Differenz Spannung: 0,3930000000000001 V
```

Beispiel – Trainingsoptimierung für KI



Researchers at Google and UC Berkeley found that AI developers can shrink a model's carbon footprint 1000-fold

Carbon Emissions and Large Neural Network Training

David Patterson¹, Joseph Gonzalez², Gauri Le¹, Chen Liang¹, Luis Miguel Martin¹, Daniel Rothchild¹, David So¹, Abul Taswar¹, and Jeff Dean¹
¹Stanford University, ²UC Berkeley, ³Microsoft, ⁴Google, ⁵Meta, ⁶MIT, ⁷Google.com, ⁸Twitter, ⁹OpenAI, ¹⁰Facebook AI Research

| Model | Transformer (Big) | Former (Transformer) | Transformer (Big) | Former (Transformer) |
|--|-------------------|----------------------|-------------------|----------------------|
| Number of Parameters (B) | 0.20 | 0.10 | 0.20 | 0.10 |
| CO ₂ Emissions (kg) | 85,000 | 85 | 85,000 | 85 |
| CO ₂ Emissions (kg/Parameter) | 425,000 | 850 | 425,000 | 850 |

Green Coding Ansätze

Green Plattform

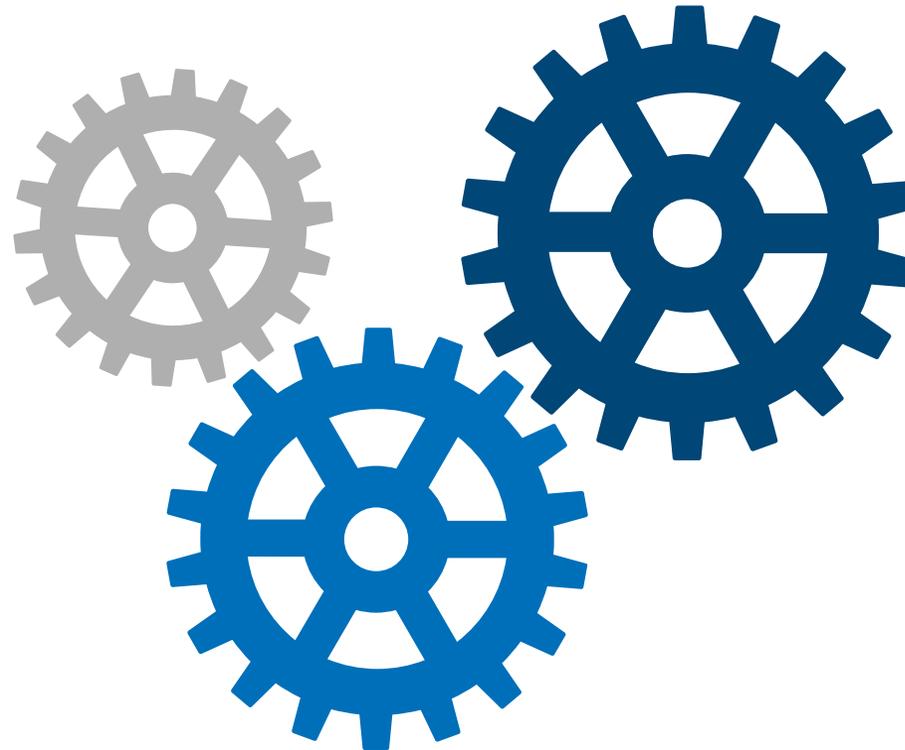
Optimale Auslastung

- Oft sind die Systeme zu groß definiert
- Falsche Konfiguration vermeiden
- Versteckte Infrastruktur bedenken

Green Methodik

Agiles Vorgehen

- Feedback berücksichtigen
- Wiederverwendbare Ergebnisse



Green Logik

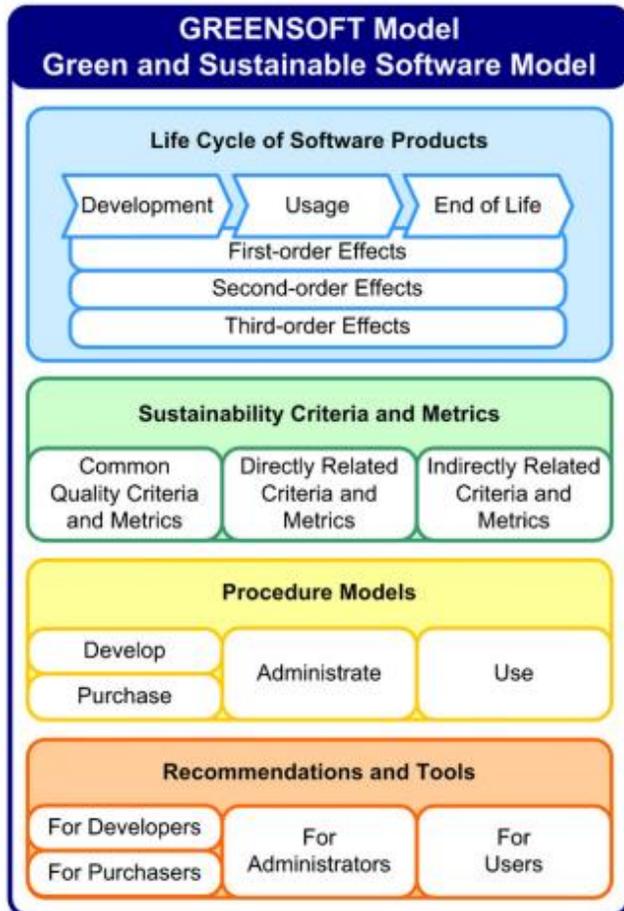
Best Practice Leitlinien mit Energiesparkriterien erweitern

- Nutzerorientierter visueller Inhalt
- Zero Waste Code
- Ressourcen mit geringem CO₂-Fußabdruck
- Nutzungshäufigkeit und Nähe.

<https://www.gft.com/de/de/technology/greencoding>

Vorgehensweise

S. Naumann et al. / Sustainable Computing: Informatics and Systems 1 (2011) 294–304



| | Development | Usage | End of Life | | |
|----------------------|--|---|---|---|----------|
| Third-order Effects | <ul style="list-style-type: none"> - ... - Changes in software development methods - Changes in corporate organizations - Changes in life style | <ul style="list-style-type: none"> - ... - Rebound effects - Changes of business processes | <ul style="list-style-type: none"> - ... - Demand for new software products | | |
| Second-order Effects | <ul style="list-style-type: none"> - ... - Globally distributed development - Telework - Higher motivation of team members | <ul style="list-style-type: none"> - ... - Smart grids - Smart metering - Smart buildings - Smart logistics - Dematerialization | <ul style="list-style-type: none"> - ... - Media disruptions | | |
| First-order Effects | <ul style="list-style-type: none"> - ... - Daily way to work - Working conditions - Business trips - Energy for ICT - Office HVAC - Office lighting | <ul style="list-style-type: none"> - ... - Manuals - Transportation - Packaging - Data medium - Download size | <ul style="list-style-type: none"> - ... - Accessibility - Hardware requirements - Software induced resource consumption - Software induced energy consumption | <ul style="list-style-type: none"> - ... - Backup size - Long term storage of data (due to legal issues) - Data conversion (for future use) - ... - Manuals - Data medium - Packaging | |
| | Development | Distribution | Usage | Deactivation | Disposal |

Fig. 2. Cradle-to-grave inspired pro

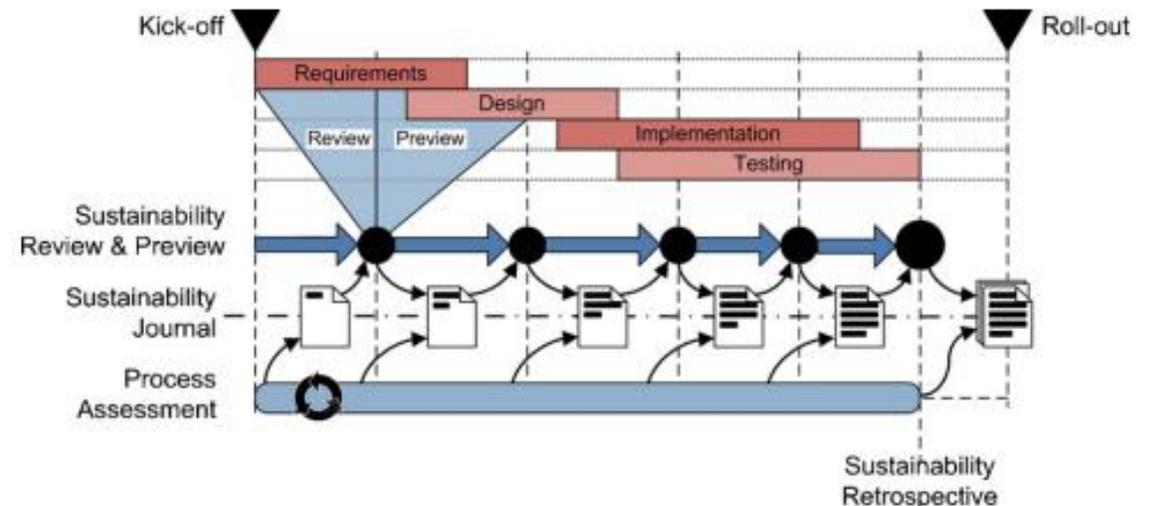
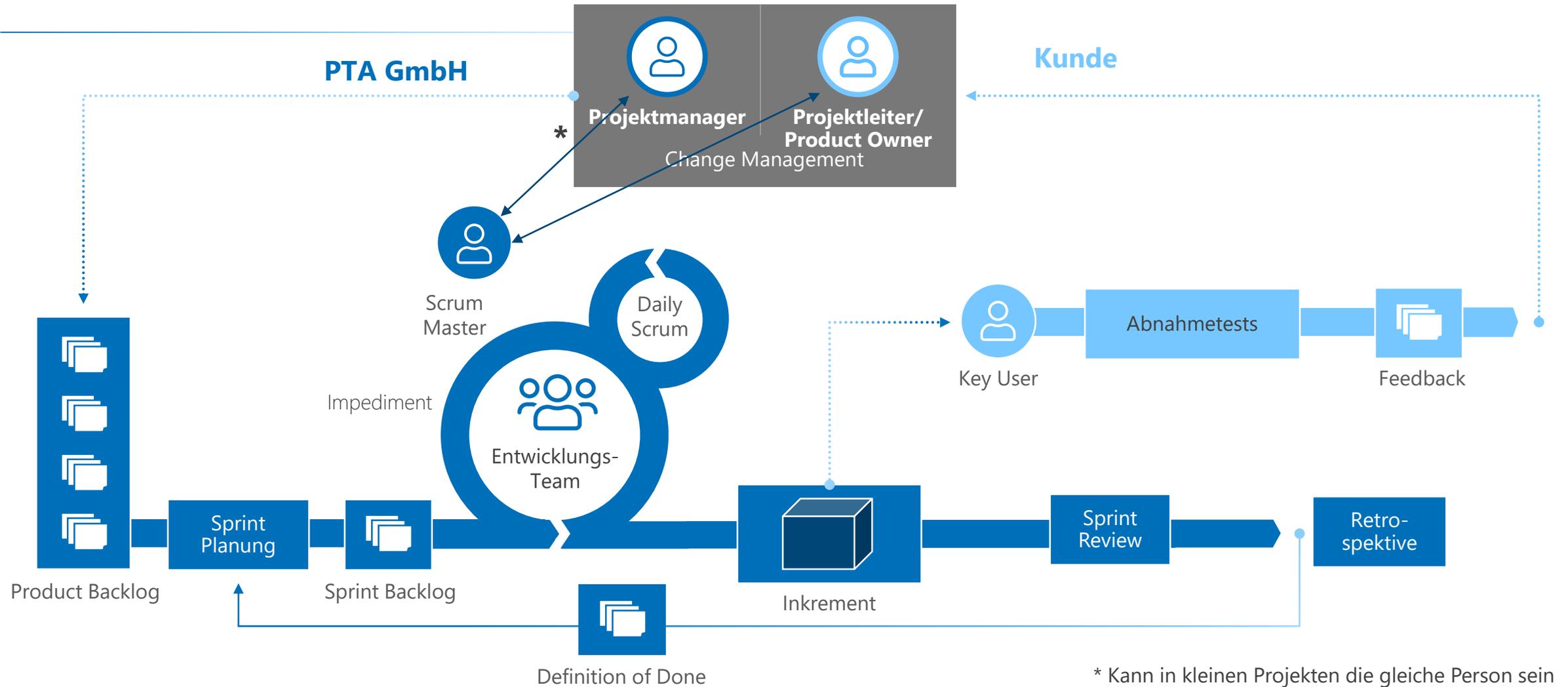
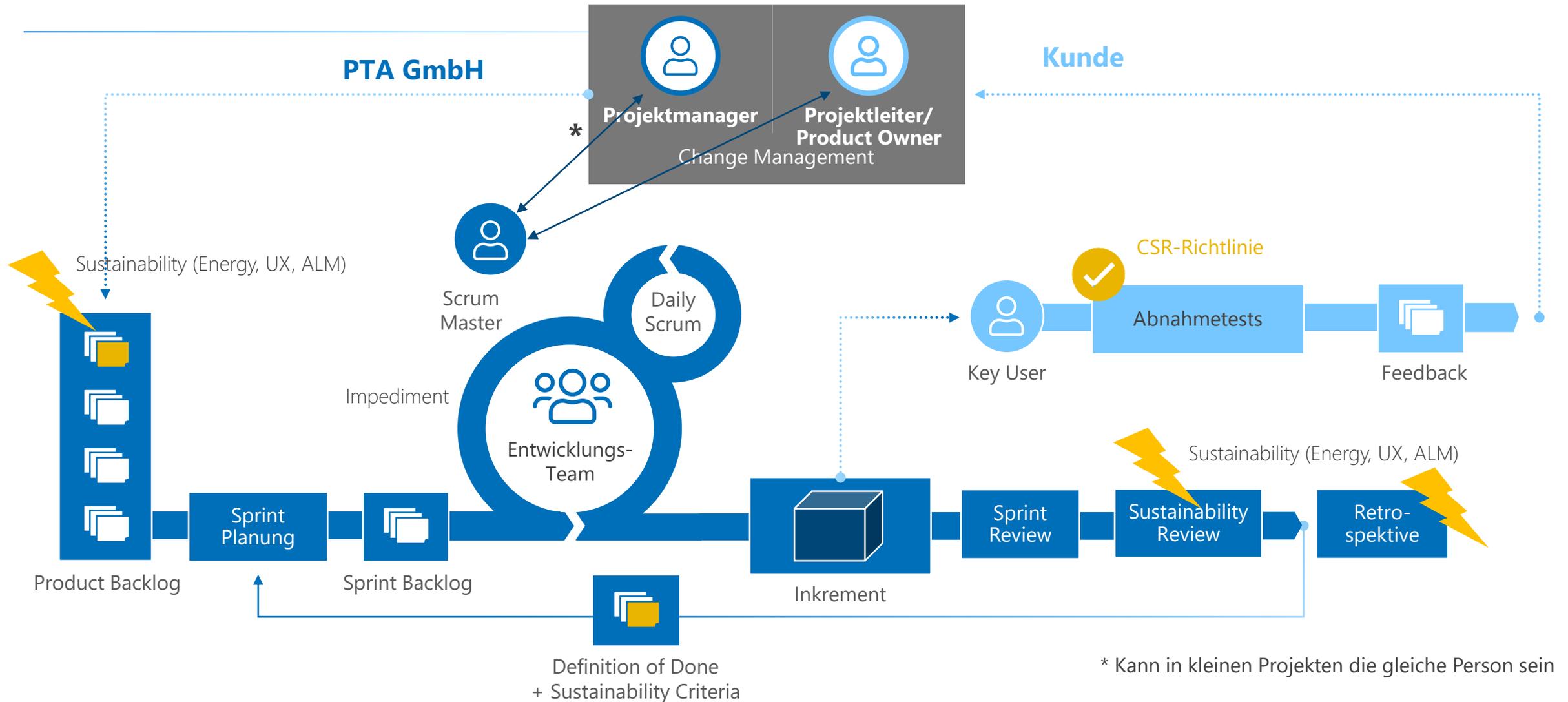


Fig. 3. Example for enhancing software development processes that fits into the procedure model part "Develop" [29].

Agiles Vorgehensmodell



Agiles Vorgehensmodell





03

Fazit

Zusammenfassung / Fazit

1. Wir können mit Softwareentwicklung einen erheblichen Energiesparbeitrag leisten!
2. Wir können Green Coding berücksichtigen beim
 - Anforderungs-Engineering
 - Vorgehensmodell
 - Design und **Konzept** / Evaluierung der Software und der Software-Architektur
 - **Konfiguration** und **Coding** nach erweiterten Programmierrichtlinien und Verwendung EnergyBenchmark
 - Application Life Cycle Management
3. Wir werden die Ansätze und unseren Leitfaden kontinuierlich verbessern!



Vielen Dank – bei Fragen sind wir für Sie da....

PTA GmbH – IT Beratung aus Mannheim

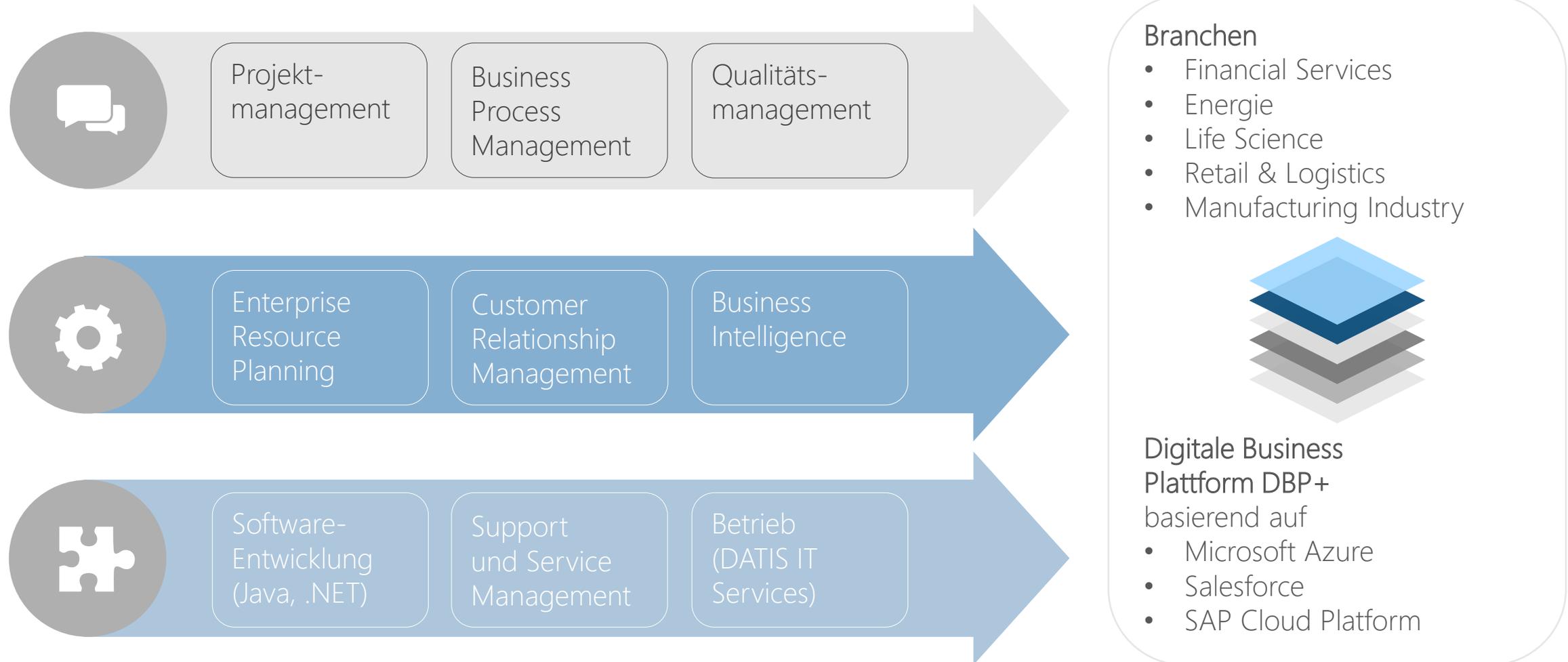
ZAHLEN & FAKTEN



- ✓ Seit 1969 als IT-Dienstleister erfolgreich im Markt
- ✓ DATIS GmbH: Seit 2005 ein Schwesterunternehmen der PTA und Service-Rechenzentrum
- ✓ Hersteller- und Finanzmarktunabhängig
- ✓ Über 400 festangestellte Berater in der PTA-Gruppe
- ✓ Zertifiziert nach ISO9001, ISO13485 & IT-Services nach ISO27001
- ✓ Stabiles Geschäftsmodell, Stammkunden, gezieltes Wachstum
- ✓ Kundennah in Ihrer Region:
kompetent, zuverlässig, reaktionsschnell, kostengünstig
- ✓ Geringe Fluktuation, hohe Bandbreite an Themen
- ✓ Über 3.000 erfasste Projektreferenzen auf www.pta.de



... mit einem umfassenden Leistungsspektrum ...



... und einem Avatar basierenden Beratungsassistenten



PTA GmbH – IT Beratung aus Mannheim

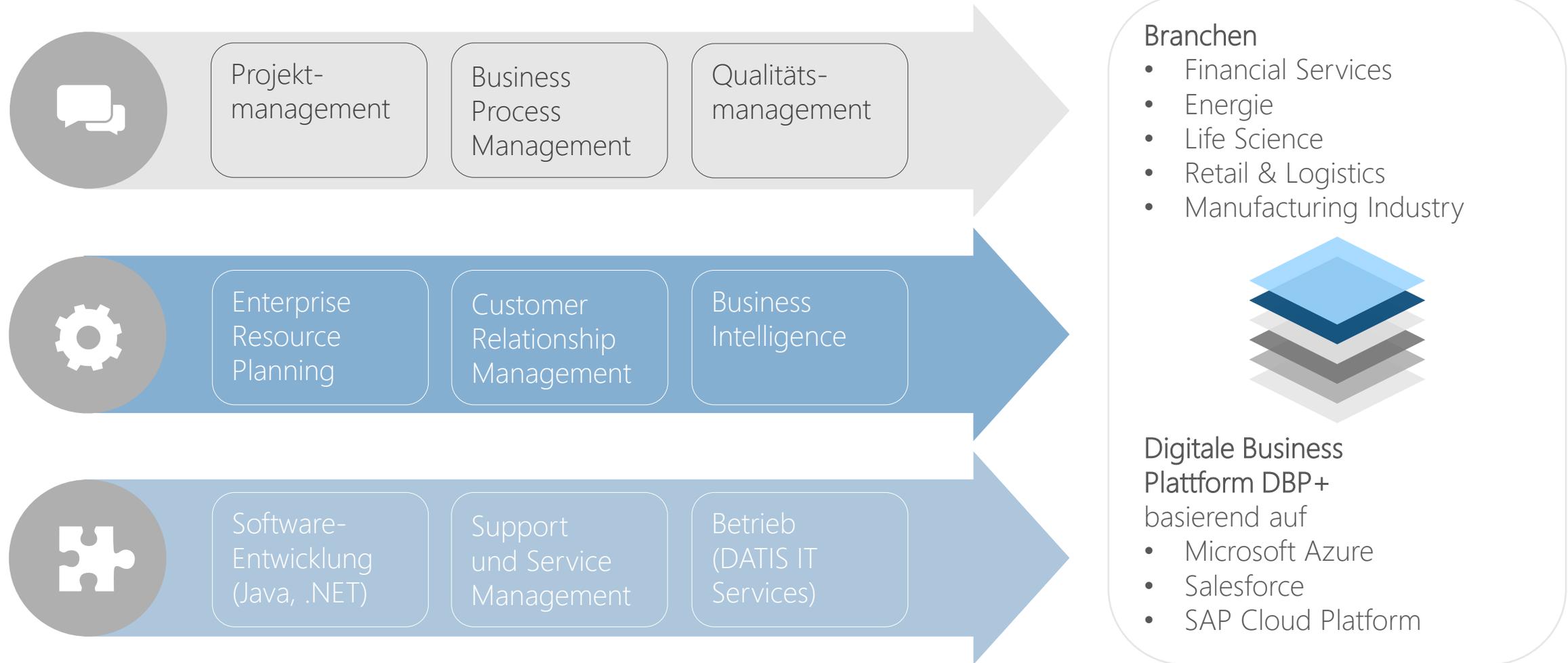
ZAHLEN & FAKTEN



- ✓ Seit 1969 als IT-Dienstleister erfolgreich im Markt
- ✓ DATIS GmbH: Seit 2005 ein Schwesterunternehmen der PTA und Service-Rechenzentrum
- ✓ Hersteller- und Finanzmarktunabhängig
- ✓ Über 400 festangestellte Berater in der PTA-Gruppe
- ✓ Zertifiziert nach ISO9001, ISO13485 & IT-Services nach ISO27001
- ✓ Stabiles Geschäftsmodell, Stammkunden, gezieltes Wachstum
- ✓ Kundennah in Ihrer Region:
kompetent, zuverlässig, reaktionsschnell, kostengünstig
- ✓ Geringe Fluktuation, hohe Bandbreite an Themen
- ✓ Über 3.000 erfasste Projektreferenzen auf www.pta.de



... mit einem umfassenden Leistungsspektrum



Back Up

Einfluss der Programmierung

- Es besteht ein direkter Zusammenhang zwischen Programmierung und dem Verbrauch von Energie. Einzelne Faktoren erscheinen erstmal in geringer Bedeutung, in Summe betrachtet und dann auch mit einer großen Anzahl von Datensätzen sieht die Bilanz ganz anders aus.
- Hierzu gibt verschiedene Merkmale, um dieses zu verdeutlichen:
 - Fehlerbehandlung
Läuft ein Programmcode auf einen Fehler ist das teuer, und das unter verschiedenen Aspekten. Unter Energetischen Aspekt ist es teuer, weil die Laufzeitumgebung versucht, den aufgetretenen Fehler zu seinem Ursprung zurückzuverfolgen. Dazu wird erheblich an Prozessorleitung aufgewendet und damit auch Energie.
 - Datentypen (Datenbank)
Datentypen sollten in Ihrer Verwendung bemessen werden. Zur Verdeutlichung: Eine Postleitzahl in Deutschland ist 5-stellig, numerisch, dann benötige ich kein String-Feld mit 50 Zeichen.
 - Type Konvertierung (Primitive Typen, Komplexe Typen)
 - Verwendung von Collections

Einfluss der Programmierung

- Type Konvertierung (Primitive Typen, Komplexe Typen)
Die Konvertierung von Datentypen (z.B. Zahl, Text, Datum) ist aufwendig, auch Energetisch. Daher ist von Anfang an in der Programmierung darauf zu achten, dass für einen Programmteil die richtigen Datentypen ausgewählt werden.
- Verwendung von Collections
In der modernen Programmierung gibt es eine Vielzahl von Collection-Typen. Je nach Aufgabe für einen bestimmten Programmteil ist es notwendig sich für einen passenden Typ zu entscheiden, mit dem die gestellte Aufgabe umgesetzt werden kann.

Einfluss der Programmierung

- Um Software zu erstellen die sich als "Energieeffizient" darstellen soll, ist keine besondere Programmierung notwendig. Aber es geht im Grundsatz darum, bestehende Qualitätsrichtlinien zur Softwareentwicklung konsequent anzuwenden und zu prüfen.
- Die Softwarearchitektur ist der entscheidende Baustein für die "Energieeffizienz" einer zu erstellenden neuen Applikation, da hier der Einfluss auf die Oberfläche, Applikation Logik und Datenbank geplant und gesteuert wird.
- Ein weiterer Einfluss kann über dem Komfort einer Applikation gesteuert werden.
- Am Ende wird aber entscheidend sein, wie die gestellten Anforderungen, Energieeffizienz, Planung und Kosten zusammen passen.
- Das Erstellen von energieeffizienter Software ist stark von Planung und Auswahl geprägt, um für die verschiedenen Programmteile die passende Programmierung zu finden und zu erstellen.