Where is the green in the mindset?
Improving awareness for green coding in the software development process

Verena Majuntke

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Motivation

- ICT accounts for approx. 8-10 % of the total global energy consumption ([1], 2023)
- Data centers used 1-1.3% (240-340 TWh) of global electricity (IEA, 2023)
- Further scope: Radio Access Networks, Internet, IoT Devices, Home …

Environmental influence data centers:

- Carbon Intensity Germany 2023: 354g CO2/kWh → 85 billion kg CO2
- Carbon Intensity China 2023: 492 g CO2/kWh → 118 billion kg CO2

How can we develop software such that the software and its development process use as little resources as possible?

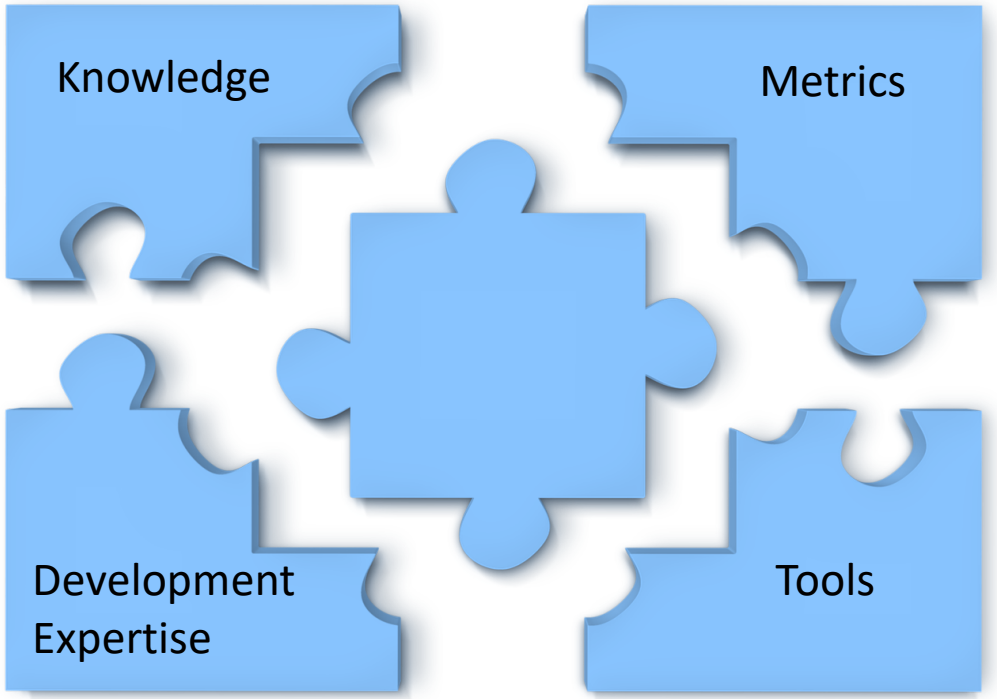
Green Coding and Sustainable Software

Definition Green Coding:

> Green Coding is the act of designing, developing, maintaining, and (re-)using software systems in a way that requires as little energy and natural resources as possible. Green Coding methods or practices thus mean any action or use of technology intended and suitable to further this. (Junger et al., 2024)

Definition Sustainable Software:

> Sustainable software is software that minimizes environmental impact and maximizes resource efficiency in its design, development, operation and use while considering economic and social factors. (Calero et al., 2021).


Knowledge
Metrics
Development Expertise
Tools

Challenges – Reducing energy consumption

1	Energy consumption is often a desire not a target / need for KPIs
2	There is a lack of awareness and information on env. sustainability
3	Lack of simple tools with easy integration in tool chain
4	Lack of knowledge on how to assess environmental sustainability
5	There is a lack of knowledge and best practices on how to reduce resource consumption
6	Communication and transparency (on company level) is required

Approach for agile development

Environmental sustainability as a binding non functional requirement (NFR)

How to approach:

- Identify which aspects need to be assessed / measured
- Specify the measurement scenario with defined workload
- Measure the identified parts
- Compare results to target value
- BUT: Target values do not exist yet, no context

Approach for agile development

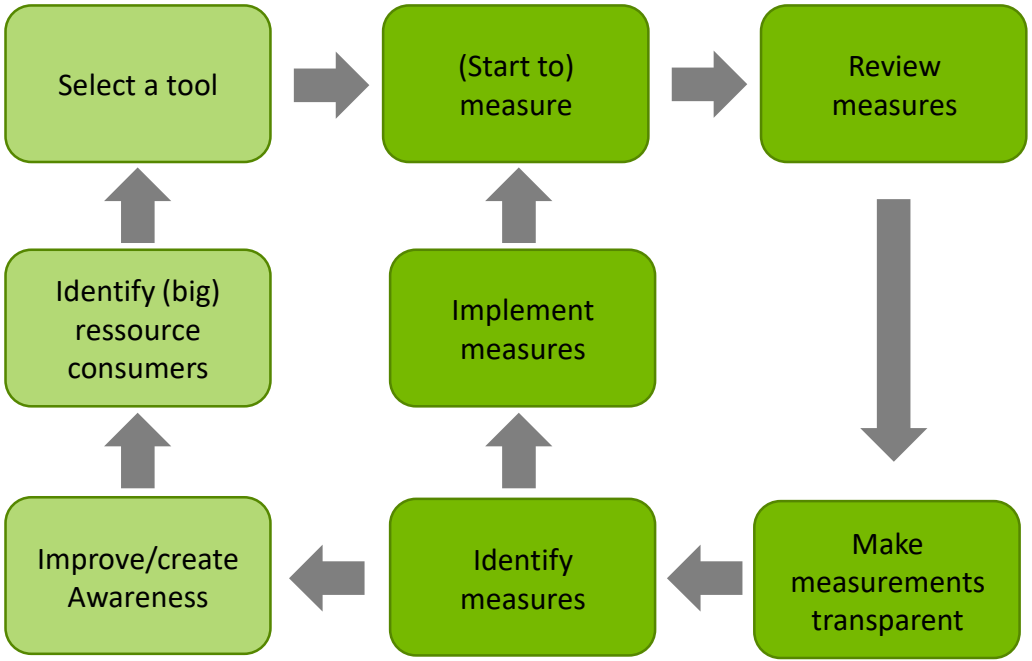
Environmental sustainability as a binding non functional requirement (NFR)

How to approach:

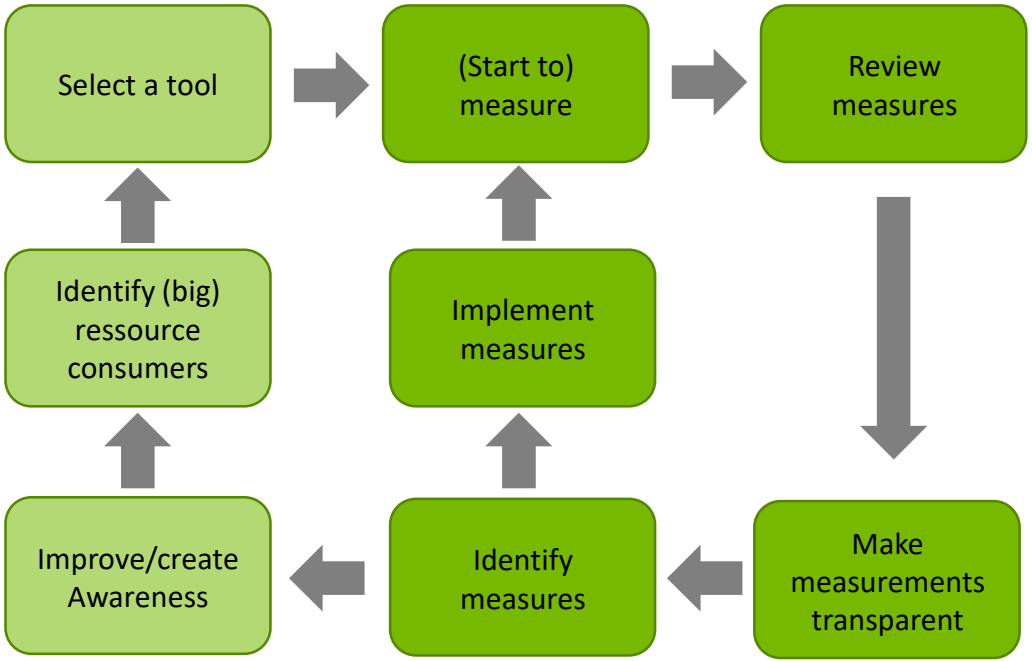
- Identify which aspects need to be assessed / measured
- Specify the measurement scenario with defined workload
- Measure the identified parts
- Compare results to target value
- BUT: Target values do not exist yet, no context

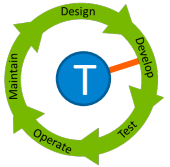
Integrate environmental sustainability continuously

Implementing environmental sustainability

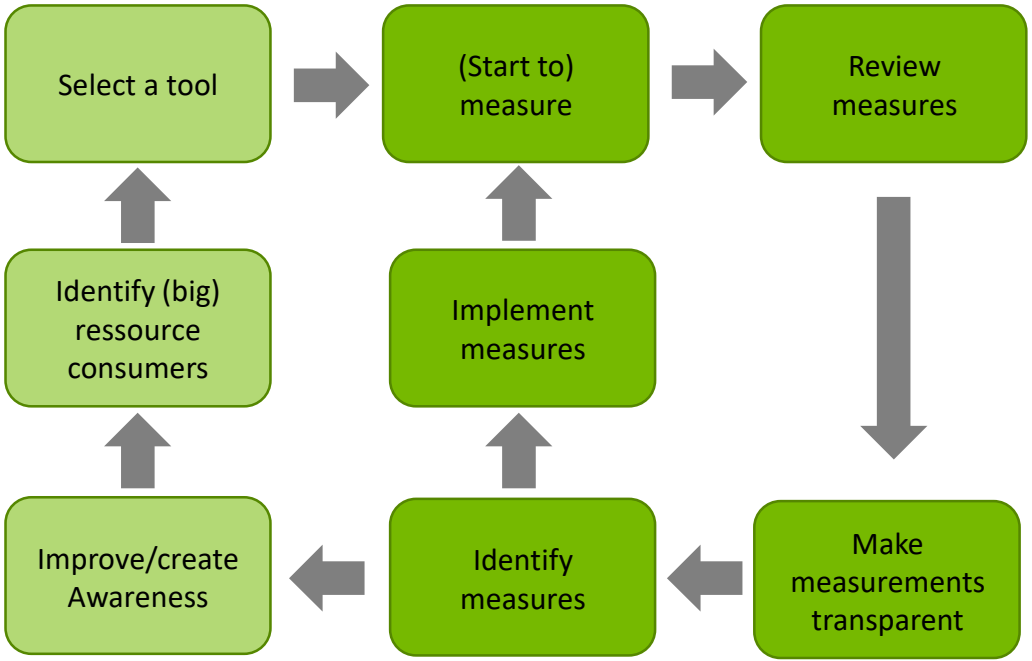


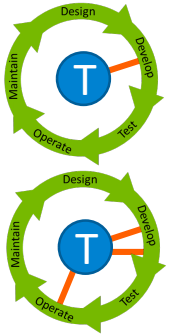
Implementing environmental sustainability



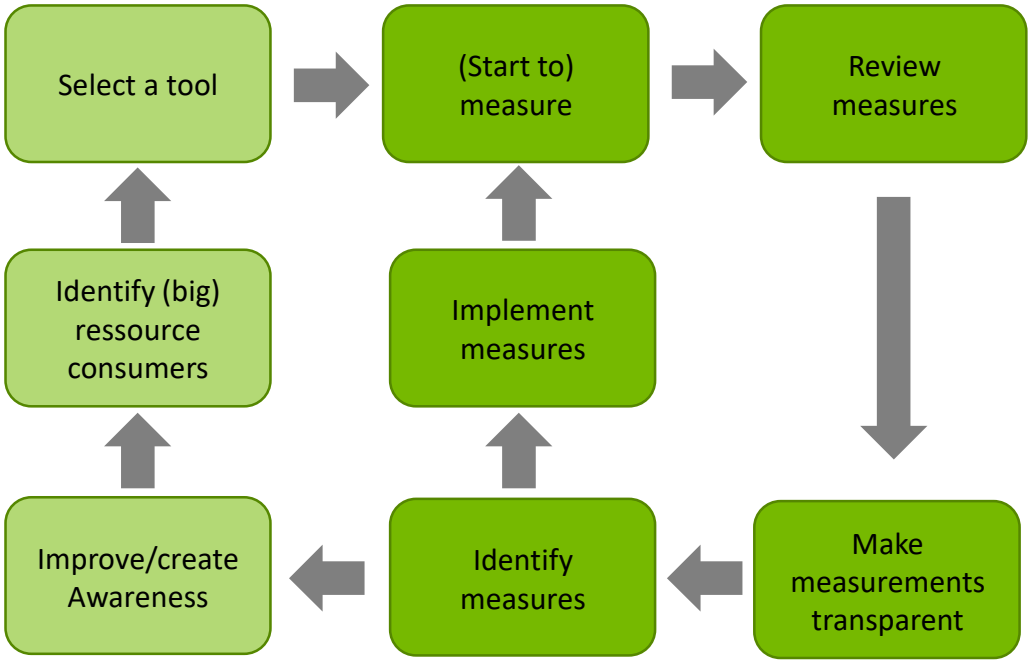


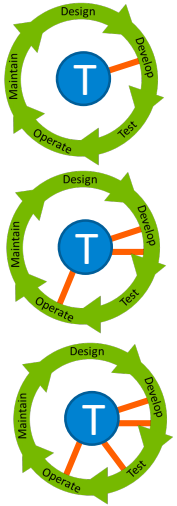
Implementing environmental sustainability



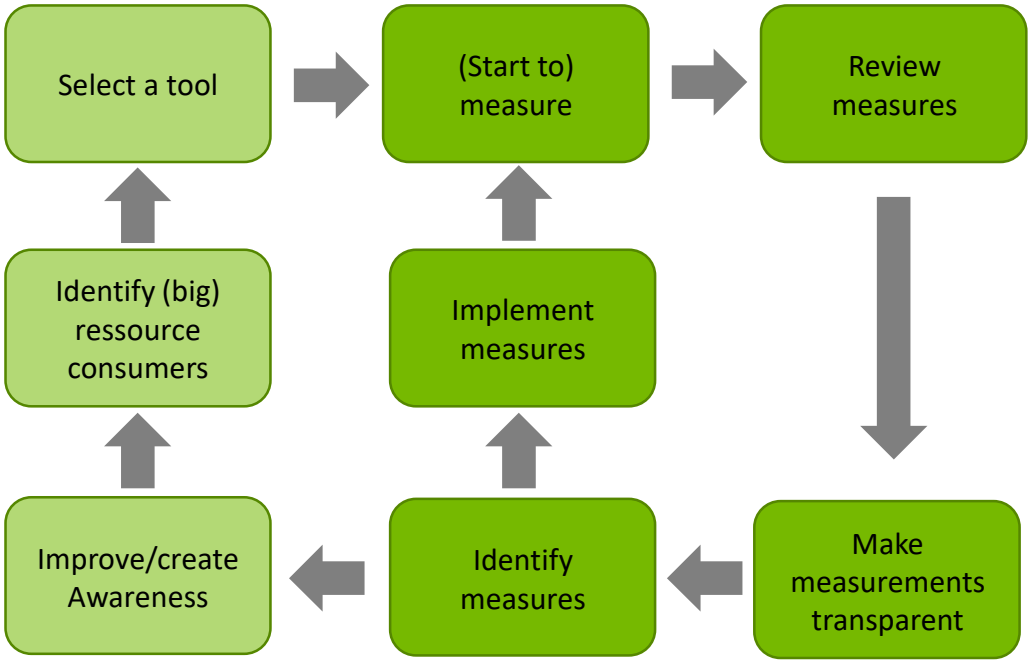


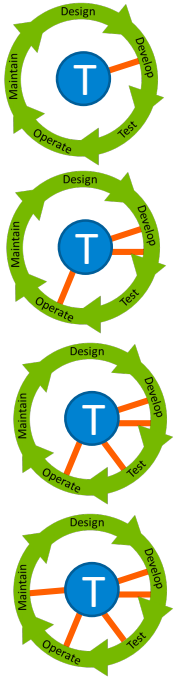
Implementing environmental sustainability



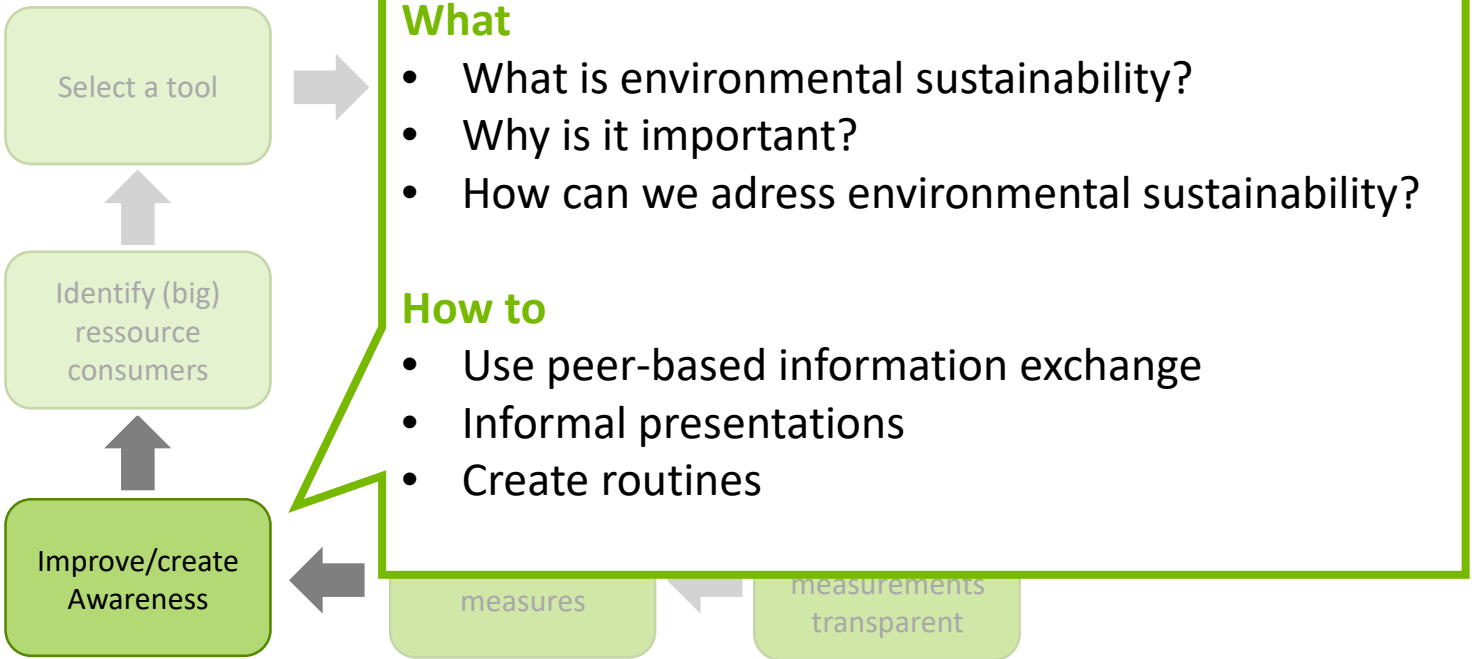


Implementing environmental sustainability

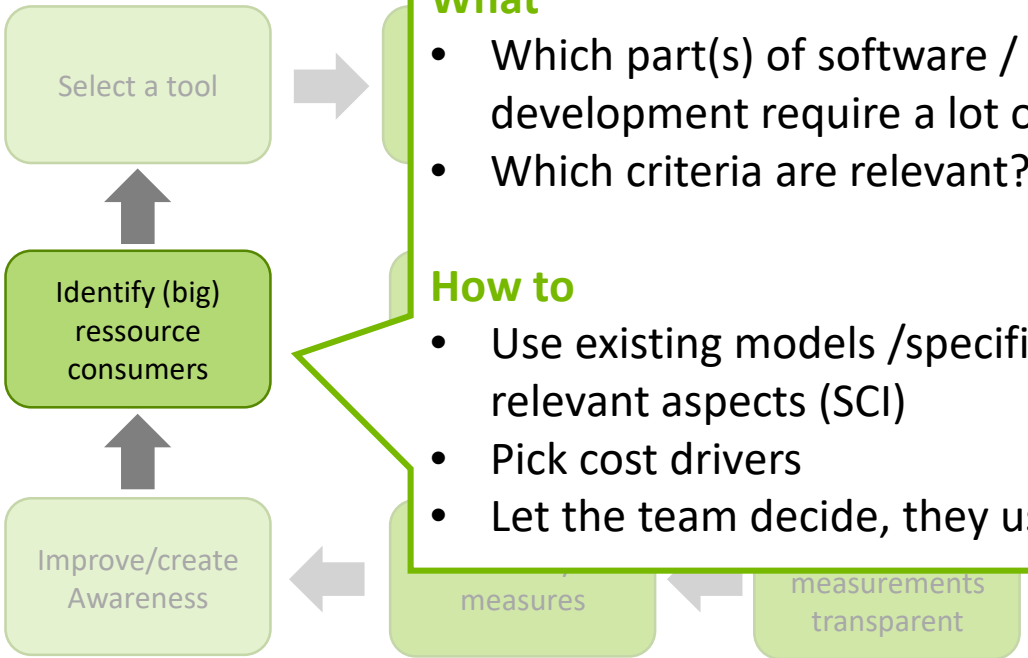




Implementing environmental sustainability



Implementing environmental sustainability



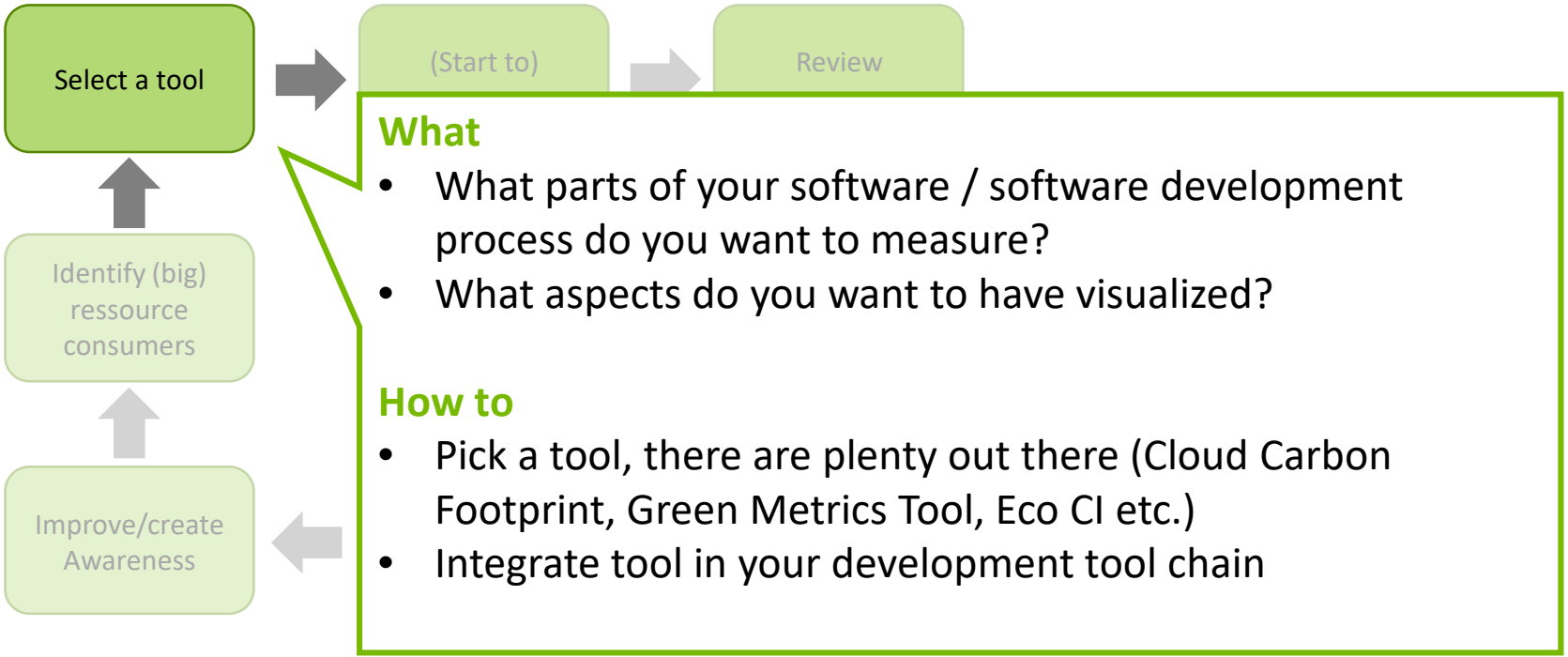
What

- Which part(s) of software / software development require a lot of ressources?
- Which criteria are relevant?

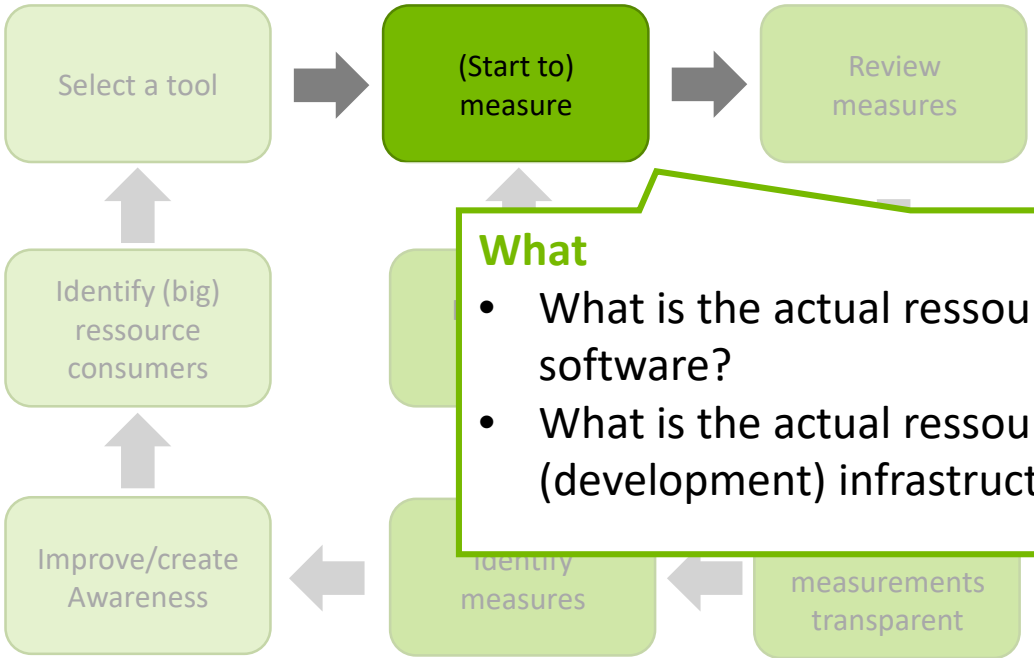
How to

- Use existing models /specifications to identify relevant aspects (SCI)
- Pick cost drivers
- Let the team decide, they usually know it best

Implementing environmental sustainability



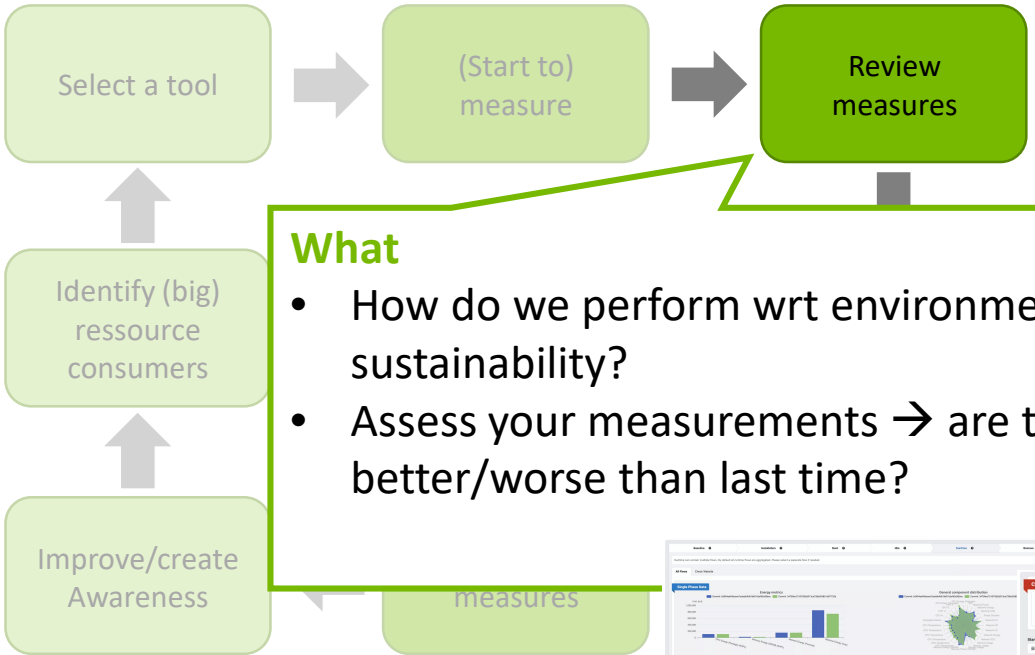
Implementing environmental sustainability



What

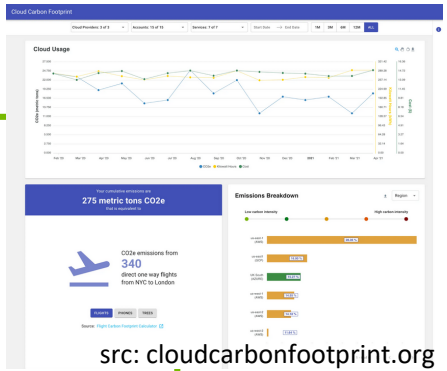
- What is the actual ressource usage of the software?
- What is the actual ressource usage of the (development) infrastructure?

Implementing environmental sustainability

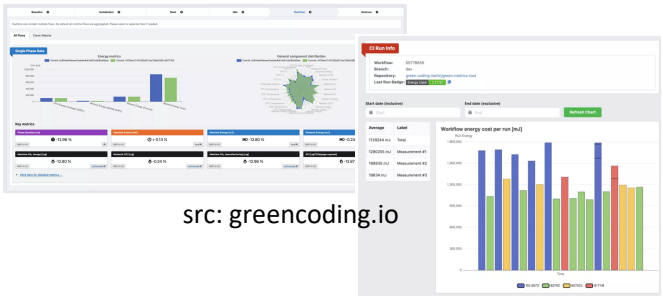


What

- How do we perform wrt environmental sustainability?
- Assess your measurements → are they better/worse than last time?



src: cloudcarbonfootprint.org



src: greencoding.io

Implementing environmental sustainability

What

- Who should know about your performance and improvement?

How to

- Use your tool(s) to make it transparent
- Communicate

Implementing environmental sustainability

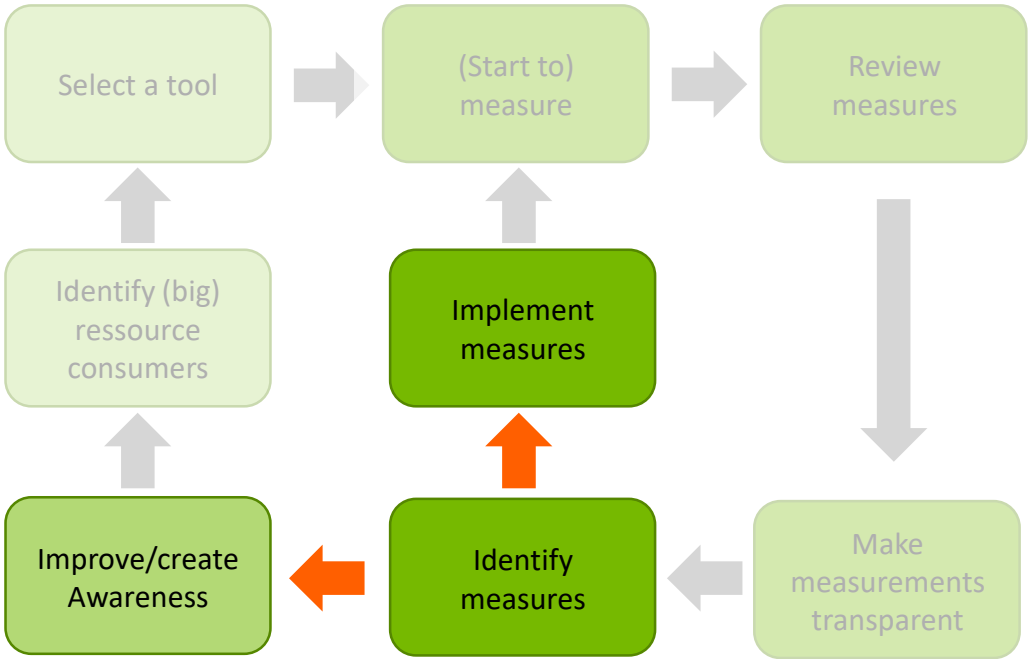
What

- How can we improve our software / process?
- Do we have knowledge/ideas how to get better?
- Do we need more information/ training for improvement?

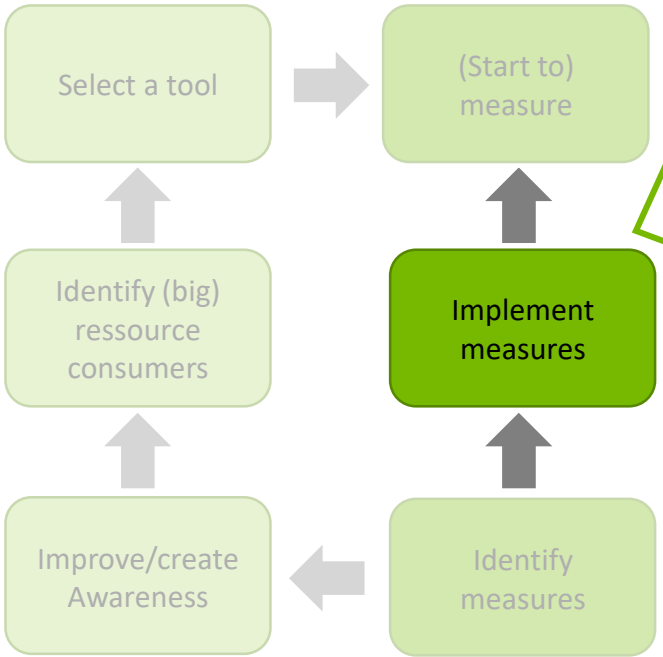
How to

- Literature (Books, Reports)
- Best practices (e.g. Green Software Foundation)

Implementing environmental sustainability



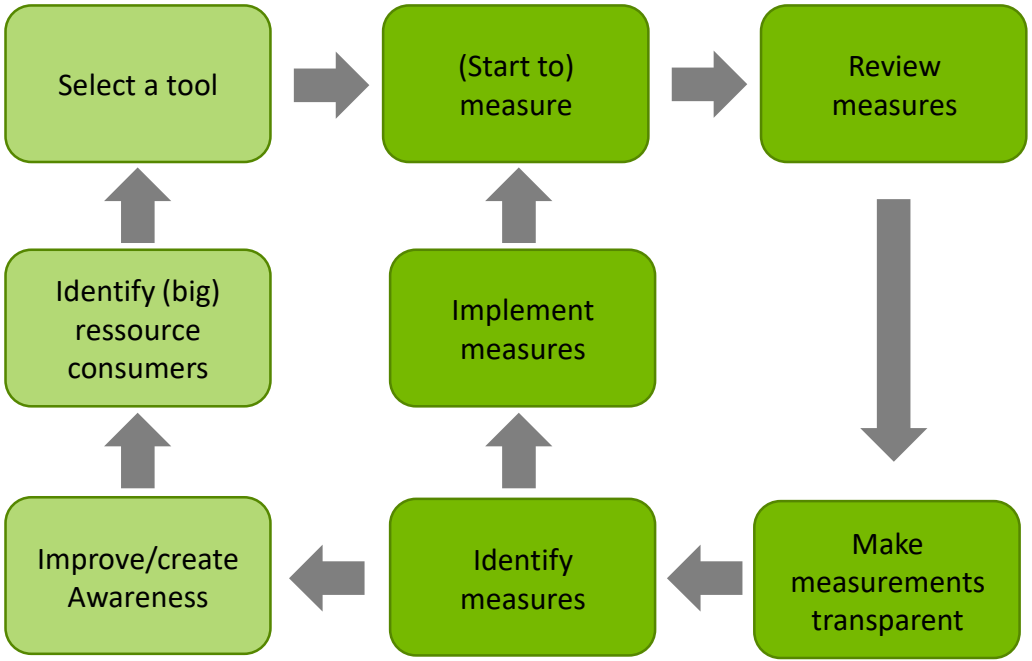
Implementing environmental sustainability



What

- Switch to a less carbon intensive cloud provider
- Shift load test in times with low carbon intensity
- Reduce the number of tests run for pushes
- Exchange energy intensive packages
- Minimize data exchange
- …

Implementing environmental sustainability



Considerations – Project start

The NFR environmental sustainability has an impact on

- design decisions (Design Patterns, Architecture, Edge Computing etc.)
- Infrastructure and tools
- Choice of frameworks and programming languages

- Developer time as natural ressource with carbon emission
- Tradeoff between measures and developer time

Benefits

- Ressource efficiency improves quality of software and software development
- Ressource efficiency also implies aspects like maintainability, modifiability, downward compatibility, avoidance of technical debts …
- Better assessment and anticipation of ressource usage
- Make your efficiency transparent and communicate it as part of your expertise

Summary

- Integrating green coding practices and methods isn‘t rocket science
- The theory and tools are already out there, you just need to start using them
- Make use of continuous improvement to steadily approach environmental sustainability
- While there is a lot of theory out there, practical insight is still rare

Questions?

www.htw-berlin.de

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Challenges - Measuring energy consumption

Study	Findings	Context
1) Fang et al.	• Developers lack knowledge and best practices on how to reduce energy consumption • 18% of study participants consider energy consumption in their decisions	Study with 100 participants on software energy consumption
2) Pinto et al.	• Developers are aware of energy consumption issues but do not know how to deal with it	Empirical study, evaluation of 300 questions and 550 answers from 800 developers on STACKOVERFLOW
3) Manotas et al.	• Mobile developers are concerned about energy consumption • Energy concerns are often neglected during maintenance • Energy requirements are often desires not targets	Quantitative (464) and qualitative (18) study
4) Ournani et al.	• Need for more awareness and information • Need for KPIs • Desire simple tool with simple output, integration in existing tools • Informal presentations, peer-based information exchange • Communication on company level is required	Interview with 10 software developers with more than 15 years of experience

Übersicht: Techniken und Praktiken

I Design

- Wahl von Design Patterns
- Microservices Architecture
- Minimierte Netzwerklast
  - Edge Computing
  - Library und Plugin Imports
  - Caching

II Entwicklung

- Ressourceneffizientes Programmieren
- Hohe (Abwärts-Kompatibilität)
- Wartbarkeit
- Auswahl von energieeffizienten Modi
- Transparenz
- Auswahl von Datenstrukturen und Algorithmen
- Open source
- Source Code Analyse
- Refactoring
- Wahl der Programmiersprache
- Programmierpraktiken

III Betrieb

- Hosting
- Taktiken für public clouds
- Demand shaping

IV Nutzung

- Offline-Nutzung
- Sustainable pre-settings
- Visibility