Strategies for green software (incl. AI) | v2026.02.25

Infographic | *Software Energy Lab* | Radboud University
Bernard van Gastel | *Bernard.vanGastel@ru.nl* | https://sustainablesoftware.info

Radboud Universiteit

A feedback *to people*

1. dashboard
2. effect on battery life
3. outside consequences of interactions with IT
4. share experience & data
5. general awareness
6. user behavior suggestions

B usage *by organisations*

1. flexible strategy
2. fitting software/hardware
3. reduce requirements (e.g. uptime/sla)
4. adjust usage to current conditions
5. max storage limit and usage limits
6. productivity of developers and teams

C handling *of data*

1. batching
2. ahead or on-demand
3. store less
4. transmit less
5. algorithm to data (avoid copies)
6. improve AI models (incl. smaller models)

D overhead *of software*

1. coding languages (more ahead of time)
2. coding techniques (like event driven)
3. minimise interaction between components
4. less layers and (unused) features
5. reuse results
6. improve algorithms (incl. program logic)

E utilization *of systems*

1. sleep more (e.g. dev/test envirs.)
2. use lean systems
3. update
4. concentrate
5. scale systems
6. make modular

F flexible *hosting*

1. use when/where green
2. right sizing
3. avoid vendor lock-in
4. use hardware longer
5. smaller (internet) distance
6. security (prevent unauthorized use)

Explanation

Our research (see our website) indicates that it is **difficult for stakeholders** within ICT projects to **communicate about the environmental impact** of/with software, and that **many professionals lack an overview of strategies** in their conversations. With this discussion guide, we aim to support these challenges.

These strategies can be applied during **requirements, architecture, implementation, testing, and deployment**. This infographic can be used in **conversations between different roles**, such as technicians, purchasers, and managers.

A structure for consumers that is often used: **computing, storage, transmission, and idle**. Each application and use case has **a different mix of consumers**, and therefore requires **different strategies**. Sometimes there is a **conflict between different strategies**, and a **custom trade-off** must be made.

The perspective of **process, technology, and product** is often used for producing quality products such as sustainable software.

Definitions

Sustainable digitalization is deciding which processes in a chain to support with software, and which not to. This takes into account economic, social, and environmental impact.

Sustainable software is created with **sustainability-by-design** (which includes green-by-design): being aware of impact throughout the entire software development process. This is in contrast to impact analysis and mitigations done afterward, which can lead to missed opportunities.

Green software is **green-by-design**:
- software with **minimal environmental impact** (from IT itself), or
- software with **a green effect** (using IT to achieve a positive environmental outcome).

Electronics have an environmental impact during **production, use, and end-of-life**. This impact can be **electricity** (energy) and the associated **CO_2 emissions**, but also **water** or other resources.