

SUSTAINABLE SOFTWARE DEVELOPMENT



A comprehensive approach that centers around two main ideas:



Product teams should make their solutions as cost-efficient, productive, and eco-friendly as possible



Optimization should increase the quality of products



8

PRINCIPLES OF SUSTAINABLE SOFTWARE ENGINEERING

#1

CARBON

Build apps that are carbon efficient

The footprint of a spam email is 4g CO₂

#2

ENERGY

Develop software that is energy efficient

An email with a photo or large attachment accounts for 50g CO₂

#3

CARBON INTENSITY

Consume electricity that has the lowest carbon intensity

The Internet, devices, and the systems supporting them generate about **3.7%** of global greenhouse emissions

#4

EMBODIED CARBON

Build apps that are hardware efficient

The average website produces **1.76g** CO₂ per page view

#5

ENERGY PROPORTIONALITY

Build hardware that is energy efficient and reduce its idle time

2,670 data centers are located in the United States, which is appreciably more than any other country

#6

NETWORK EFFICIENCY

Reduce data size and distance of travel

Data centers use around **1.8%** of electricity in the United States

#7

DEMAND SHAPING

Develop software that is carbon-aware

Training a single AI model emits as much carbon as 5 cars in their lifetime

#8

OPTIMIZATION

Focus on step by step optimization to boost overall carbon efficiency

Using a single computer and monitor for six years instead of four helps to avoid 190kg of carbon emissions equivalent

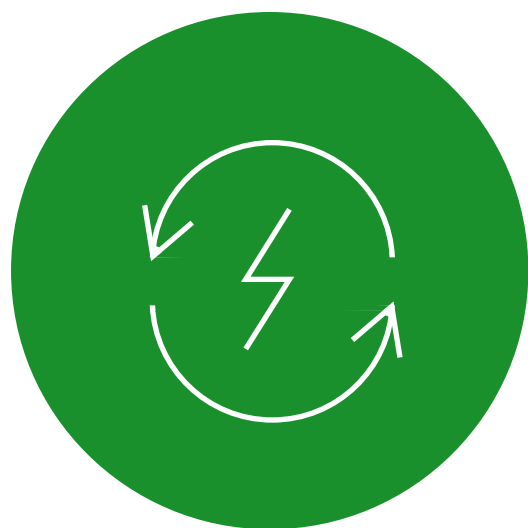
GLOBAL BRANDS LEADING THE WAY IN SUSTAINABILITY

BRANDS HAVE COMMITTED TO RENEWABLE
ENERGY, INCLUDING:



TECH INDUSTRY LEADERS ARE SUSTAINABLE SOFTWARE ENGINEERING TREND-SETTERS

APPLE



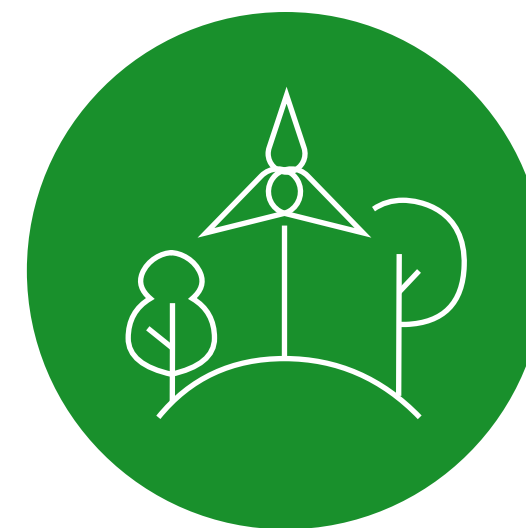
Already uses 100%
renewable energy

MICROSOFT



Will offset all historical
emissions by 2050

FACEBOOK



Committed to net zero emissions
across its value chain in 2030

TAKE 3 STEPS TOWARDS SOFTWARE SUSTAINABILITY

STEP #2

Use green coding for sustainable programming

- Optimize features with high energy consumption & common usage scenarios
- Remove unused features & loops
- Reduce data usage
- Adapt the work of the app to the power mode & operating conditions of the device

STEP #1

Rebuild your existing models and educate your team

- Implement automated testing
- Avoid overdesign
- Adopt tools to calculate the environmental impact of the software

STEP #3

Measure the impact of your product

- Monitor the energy consumption of the developed software
- Identify the modules you can optimize
- Repeat these steps throughout the entire sustainable software engineering cycle

CONTACT US

**LEARN HOW LEOBIT CAN HELP YOU
BUILD SUSTAINABLE SOFTWARE
APPLICATION TODAY**

leobit.com

Leobit