



GREEN SOFTWARE



In our current era, prioritizing sustainability and environmental consciousness has become increasingly prevalent. While there's a myriad of eco-conscious actions businesses can take to address environmental challenges, it's worth noting that software development can also be approached in a sustainable manner. Let's explore into the realm of **Green Software** technology. "Green Software" is software that is responsible for emitting fewer greenhouse gases. It refers to software solutions designed with energy efficiency, minimizing their carbon footprint and promoting sustainability.



Image Courtesy :- <https://bit.ly/498LQDd>



Image Courtesy :- <https://bit.ly/3Ty0eB>

"Let us code for a future where technology serves progress without compromising the beauty of our planet." - Bill Gates

Green Software in AI

Programmers can adopt green software strategies in AI development, particularly by trimming training data. Assessing data needs from collection to preprocessing is crucial. Studies show that reducing training data size significantly cuts energy consumption. For example, in SMS spam detection, downsizing the dataset to 20% resulted in a 75% drop in energy use with minimal accuracy loss. Selecting energy-efficient algorithms, like those assessed by tools such as CodeCarbon and ML CO2 Impact, further reduces carbon footprint. These practices optimize AI development, aligning with sustainability goals while maintaining performance.

DID YOU KNOW?

Accenture, GitHub, Microsoft, and ThoughtWorks partner with the Linux Foundation to establish the Green Software Foundation. This initiative aims to embed sustainability in software engineering. They've introduced 10 Recommendations for Green Software Development. Greentech Media praises eMeter, Sustainable Spaces, Autodesk, Intermolecular, and others for their eco-friendly software efforts..

FACT

While we often think of hardware as the main culprit, software contributes significantly to carbon emissions. Every line of code, every algorithm, and every deployment decision matters.

The environmental impact of your code can be influenced by the programming language you choose!

- C: The old but gold standard for efficiency*
- Rust: A modern language focusing on speed and safety*
- C++: Balances performance and control over system resources*
- Java: Efficient in terms of processing power, despite higher memory usage*
- Ada: Known for its strong typing, mature design, and high-level abstractions*
- Go, TypeScript, Kotlin, Julia, Dart, Scala, Haskell: These are more likely to be used in new green-field projects*
- Ruby, Python, Pearl : These are most energy inefficient*

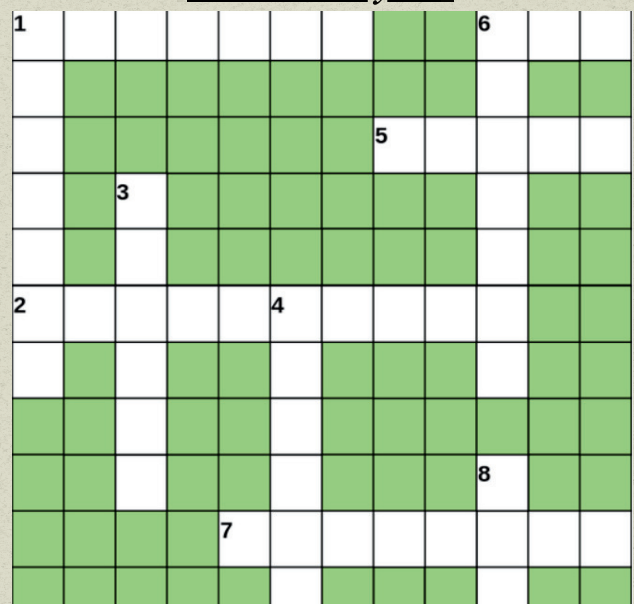
Characteristic of Green Software :

- 1.Resource Optimization :** Efficient use of system resources, including processor, memory, and storage.
- 2.Carbon Footprint :** The software should have a low carbon footprint, with a focus on reducing CO2 emissions during development and operation.
- 3.Sustainability :** Software should support sustainable practices, such as extending the lifespan of hardware through efficient performance.
- 4.Software Carbon Intensity (SCI) :** Adherence to the SCI specification, which provides a methodology for calculating the rate of carbon emissions for a software system.
- 5.Design and Coding :** Incorporate green principles in design and coding, including the choice of language and AI models that are energy-efficient.
- 6.Lifecycle Management :** Consideration of the environmental impact during the development, operation, and disposal stages of the software lifecycle.
- 7.Compliance :** Adherence to relevant green software standards and regulations.
- 8.User Control :** The software should offer controllability features to users, such as configurable settings for energy savings.
- 9.Maintainability and Portability :** Software should be easy to maintain and portable, to reduce the need for frequent updates or replacements that could have environmental impacts

GREEN SOFTWARE VS TRADITIONAL SOFTWARE :

Aspect	Green Software	Traditional Software
Energy Efficiency	Prioritizes efficient resource consumption.	Often overlooks energy efficiency.
Renewable Energy	Utilizes renewable sources to power data centers.	Relies on non-renewable energy sources.
Resource Usage	Optimizes resource utilization via virtualization.	Has less efficient server infrastructure.

Let's Play!!!



The Software Carbon Intensity (SCI)

The SCI score is a rate of carbon emissions, not a total.
The equation is a simple and elegant solution to the extremely complex problem behind it:

Carbon emitted per kWh of energy, gCO2/kWh

Carbon emitted through the hardware that the software is running on

$$SCI = ((E * I) + M) \text{ per } R$$

Energy consumed by software in kWh

Functional Unit: this is how software scales, for example per user or per device

The "per R" is what makes the SCI into a tool that works for every software domain, every use case, and every person.

Green Software Foundation
greensoftware.foundation

- DOWN :** [Image Courtesy :-https://bit.ly/3TPvKKw](https://bit.ly/3TPvKKw)
- ... are complete piece of physical hardware that is used to compute or support computer.
 - images are graphical representations of mathematical objects such as lines, curves, polygons
 - Indentation is a special type of error in programming language.
 - The..... refers to parts of a computer application or a program's code that allow it to operate and that cannot be accessed by a user.
 - is the computer network that connects computer/devices within the range of an individual person.

- ACROSS:**
- The number system consists of at most 10 digits.
 - In Symmetric Key Cryptography, same key can be used by the sender and receiver for of the message.
 - memory is the fastest system memory.
 - are unexpected problem with software or hardware.
 - An operating system is a system that acts as an intermediary between computer user and computer hardware.

Answer Key:

DOWN- 1. DEVICES 3. VECTOR 4. PYTHON 6. BACKEND 8. PAN

ACROSS- 1. DECIMAL 2. ENCRYPTION 5. CACHE 6. BUG 7. SOFTWARE

The Decisive Guide to Sustainable Software: Eight Essential Principles

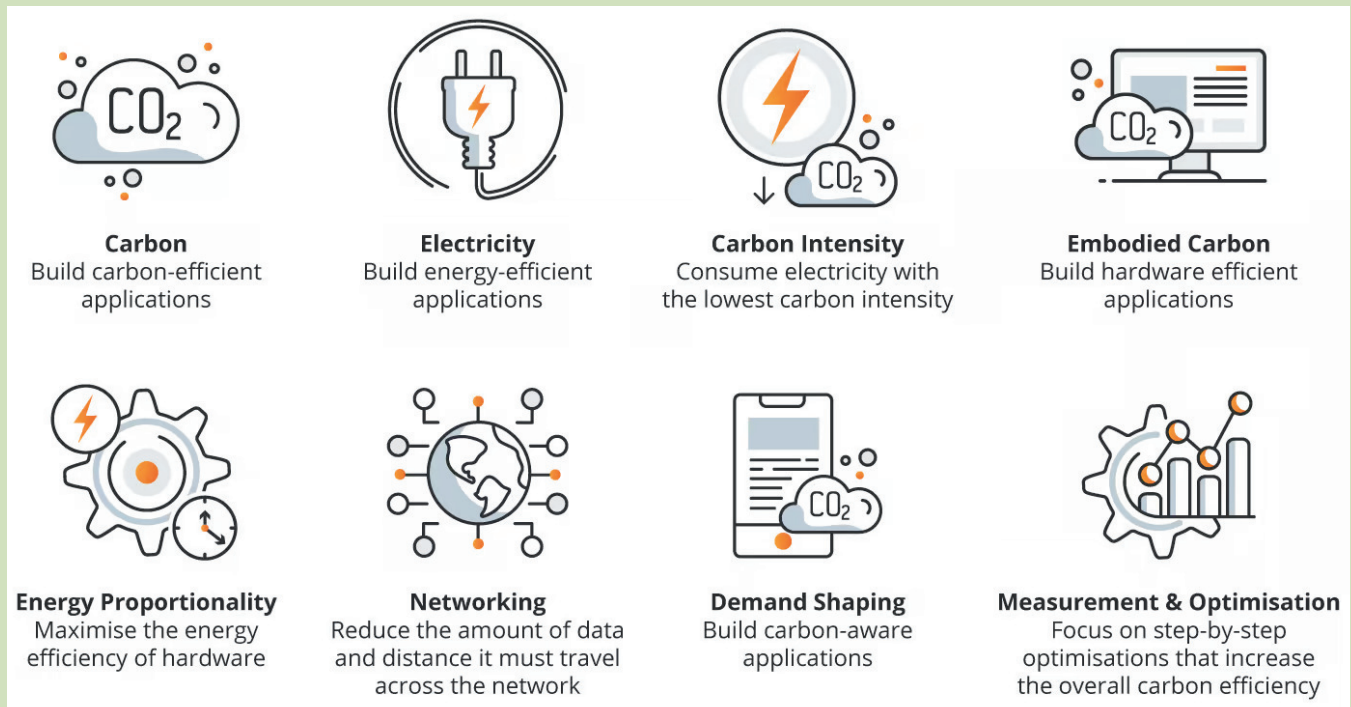


Fig.: Eight Essential Principles (Image Courtesy: www.future-processing.com)

Did you know?

1. The adoption of green software aligns with broader global initiatives and goals, such as the United Nations' Sustainable Development Goals (SDGs). Specifically, it contributes to responsible consumption and production, climate action, and sustainable cities and communities.
2. According to researchers, software is expected to account for 14% of world's carbon footprint by 2040.

A QUICK OVERVIEW OF WHAT YOU NEED



Recent Developments:

- **Renewable Energy Integration:** Green software leverages algorithms to optimize energy use, promoting reliance on renewable sources.
- **Smart Grid Technologies:** Green software supports intelligent energy distribution, enhancing efficiency and sustainability in power grids.
- **Transformative Technologies:** AI, ML and Blockchain are being leveraged to advance green tech in sectors like energy management, waste disposal, smart cities, sustainable farming, and green transportation.
- **Edge Computing:** To reduce network traffic and associated carbon footprint, there's a shift from cloud to edge Computing, minimizing interactions with central servers.
- **Reusable Components:** Incorporating reusable Components in software development can reduce the application development time and, consequently, the environmental impact.
- **Decarbonization Measures:** Organizations are enhancing their decarbonization efforts to achieve net-zero emissions. This includes measuring the energy impact of software across its lifecycle.

Image Courtesy :- <https://www.technoidentity.com/insights/sustainable-software-is-the-next-normal-scaling-to-make-the-difference/>

Manish Shukla

Batch - 2015

Manish Shukla is a Lead Gen AI Data scientist at Verizon renowned for his expertise in leveraging data-driven insights to solve complex problems across various industries. Join us as we embark on a journey, guided by the insightful interview conducted by Riddhi Garudkar, to explore his significant contributions to the ever-evolving realm of Data Science.



1.What inspired you to pursue a career in Data Science and what are some of the projects that you have worked on or are currently working on in this domain?

Right from childhood, I used to like the numbers a lot. When I came to VCET, I did well in mathematical subjects and at the same time I used to analyze past years and predict the questions from exams. When I joined my first company, I enjoyed playing with data. All this helped me to identify my interest and pursue my career in Data Science. Talking about my recent project, I am trying to build a Generative AI system to classify clauses from documents using Natural Language concepts and Large Language models, which will avert manual efforts and save millions of dollars.

2.How do you handle data collection, preprocessing, and augmentation for Generative AI models and what are some of the best practices that you follow?

I will say it totally depends on the use case you are trying to solve. In order to achieve the goal I follow some steps such as understanding business problems, designing data pipelines for collecting data and aggregating, Imputing statistical values and Feature Engineering for preprocessing, Interacting with User to take feedback on product and lastly deploying and scaling for product release.

3.What are some of the skills that you think are essential for a successful Data Scientist and how do you develop or improve them?

Personally I will say problem solving techniques, analytical skills and effectively communicating your thought process, it helps me to deliver innovative solutions and solve problems effectively. It develops by learning the new algorithms, developing projects by using your learnings and participating in data hackathons.

4.What are some of the challenges that you face while working with Generative AI models and how do you overcome them?

I always face the challenge of understanding business problems. Whenever I interact with users, requirements change and it leads to ambiguity in deliverables. I try to document every requirement and accordingly design a solution, before jumping into code.

5.What were some of the most memorable or impactful experiences that you had during your time at the college?

I had many memorable experiences from my time at VCET and it helped me to learn many life lessons. One of them was in DTSP class, when Prof. Ashish Vanmali used to give challenging problems to earn bonus marks. It helped me to think critically, beyond my limit to solve those problems and know something apart from the entire class.

HOD EXTC : Dr. Amrita Ruperee

Staff In-charge: Ms. Ashwini Katkar

Secretary: Miss. Riddhi Garudkar

Disclaimer: The contents of this are referred from online resources and do not necessarily represent the views of ETA committee members, VCET or any of the staff members.