# How to activate Sustainability in Software Engineering teams

Understanding sustainability as a term is easy. We know the planet needs our help in staying healthy and taking measures to promote sustainability in all aspects of life will help us help our planet.

With us being part of the technology industry, it may feel like digitizing the world is definitely a green step in this direction as we reduce the dependency on paper and subsequently lesser trees get cut. While this helps, we also must account for all the carbon we emit during the process of digitization.

“The carbon footprint of our gadgets, the internet and the systems supporting them account for about 3.7% of global greenhouse emissions, according to some estimates. It is similar to the amount produced by the airline industry globally, explains Mike Hazas, a researcher at Lancaster University. And these emissions are [predicted to double](https://theshiftproject.org/en/article/unsustainable-use-online-video/) by 2025.” [Source[: BBC.COM](https://www.bbc.com/future/article/20200305-why-your-internet-habits-are-not-as-clean-as-you-think)]

Now every time someone says that IT leads to carbon emissions, the first thing that comes to mind is – it’s all happening because of the data centers! While that is one of the significant contributors towards emissions, there are other aspects, for instance the supply chain that was used to manufacture the assets an engineer will use during development and potentially even while disposing these assets. The emissions from their company site while they worked. Hence calculating carbon emissions is trickier than it might seem, but luckily for us the community has come a long way and there are resources which can provide the foundational elements for us to move forward in our journey to be more planet friendly.

Understanding carbon emissions

Fundamental notions around sustainability and carbon emissions -

Why is everyone only talking about carbon emissions, what about the other greenhouse gases?

Carbon emissions aka carbon equivalent (CO2-eq) is the standard way of measuring greenhouse gas emissions. CO2-eq is a metric used to compare the emissions from various [greenhouse gases](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Greenhouse_gas_(GHG)) such as methane, nitrous oxide etc. on the basis of their [global-warming potential (GWP)](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Global-warming_potential_(GWP)). It works by converting amounts of other gases to the equivalent amount of carbon dioxide with the same global warming potential.

What does it mean when someone says 1 metric ton CO2-eq?

The unit used alongside CO2-eq is CO2e Metric tons or MTCO2e. To put this in perspective, 1 MTCO2e is approximately the amount of emissions when a gasoline powered car is driven for 2500 miles or results by burning 1100 pounds of coal.

Isn’t sustainable software engineering only about optimizing – “the more efficient code I write, the less time it needs on the hardware” and voila – there is lesser carbon emission?

While it’s true that there is a direct correlation between the efficiency of code and the amount of carbon emissions caused, there are also other ways to reduce the overall emissions. Take an example – the washing machine you have at home consumes power when run. More load and frequent runs would obviously lead to more power consumption and hence more emissions. So, optimizing your laundry load is one way to become more environmentally friendly 😊. With that said, think of a scenario where your provider tells you that the power quality is greener during a particular time in the day and you run your washing machine during that window, this could help you further bring down the carbon footprint. The same can be applied to software as well.

## Terminology used in sustainable software engineering

Now that you understand some basics, here are some terms that you will come across very frequently while working in this domain. This is just a subset, please see references for more details.

**Scope 1, 2 and 3 carbon emissions** – these refer to the types of emissions. To put it very simply –

Scope 1: emissions that directly result from business activities, such as stationary combustion of fuels for backup power generation in cloud datacenters.

Scope 2: emissions that indirectly result from purchased energy, such as exhausted from an electric power plant.

Scope 3: emissions that indirectly result from all other business activities, such as those associated with the upstream raw materials extraction, manufacturing, and delivery of cloud-based IT asset infrastructure (such as servers) from suppliers to be used in our cloud datacenters.

**Carbon Intensity** - The carbon intensity of electricity is a measure of how much carbon (CO2eq) emissions produced per kilowatt-hour of electricity consumed. Carbon intensity changes by location since some regions have an energy mix containing more clean energy sources than other regions.

Carbon intensity also changes over time due to the variable nature of renewable energy. For example, when it's cloudy or the wind isn't blowing, carbon intensity increases since more of the electricity in your mix comes from sources that emit carbon.

**Energy Proportionality** - Measure of the relationship between power consumed in a computer system and the rate at which useful work is done (its utilization). If the overall power consumption is proportional to the computer's utilization, then it is said to be energy proportional.

**Demand Shifting** - the strategy of moving compute to regions or times when the carbon intensity is less, or to put it another way when the supply of renewable electricity is high.

## Activating Sustainability

For someone interested in the domain of sustainable software development, the immediate thought is where and how do I get started. Here is a simple framework that I am leveraging to structure our team’s work towards sustainability.

Measure | Assess | Optimize

**Measure** – You can't improve what you don't measure. The first step is towards identifying the amount of carbon emissions due to our systems. You could be managing full stack applications, services, data stores, backend processes – everything contributes to emissions and knowing how much we emit has to come first before we can start making a positive change in this direction.

**Assess** – Once you measure, what’s next? It’s critical that we assess our systems on an ongoing basis to understand their impact on emissions. Assessing would require formulating a set of rules/standards against which to evaluate our systems (they will need to vary for different kinds of systems – a web-based application might have standards on load time, page size etc. while infrastructure gets evaluated on their hardware efficiency). We will incorporate a sustainability assessment for all types of systems in the same way as we do today for accessibility, privacy, security etc.

**Optimize** – Knowing where we stand today and how good or bad our systems with regards to their carbon emissions, will help us take the right steps to optimize and build improved and more sustainable systems. Optimization techniques should be made reusable whether they be in the form of practices, guidelines or automations.

## Interesting Facts

Bitcoin consumes more energy than Switzerland, according to new estimate

| Source: Verge – Tech Publication

*Global IT sector electricity demand ranks behind only two countries in the world – China and US*

| Source: [www.climatecare.org](http://www.climatecare.org)

*If every adult in the UK sent one less “thank you” email, it could save*[*16,433 tonnes of carbon a year*](https://www.ovoenergy.com/ovo-newsroom/press-releases/2019/november/think-before-you-thank-if-every-brit-sent-one-less-thank-you-email-a-day-we-would-save-16433-tonnes-of-carbon-a-year-the-same-as-81152-flights-to-madrid.html)*– the equivalent to taking 3,334 diesel cars off the road, according to energy company, OVO.*

*| Source:* [*BBC.COM*](https://www.bbc.com/future/article/20200305-why-your-internet-habits-are-not-as-clean-as-you-think)

Five billion plays clocked up by just one music video – **the hit 2017 song Despacito** – consumed as much electricity as Chad, Guinea-Bissau, Somalia, Sierra Leone and the Central African Republic put together in a single year. The total emissions for streaming that song could be over 250,000 tonnes of carbon dioxide.

*| Source:* [*BBC.COM*](https://www.bbc.com/future/article/20200305-why-your-internet-habits-are-not-as-clean-as-you-think)

## References and Credits

[Sustainability guide - Microsoft sustainability](https://www.microsoft.com/en-us/sustainability/sustainability-guide)

[Putting a CO2 figure on a piece of computation | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/6128960)

[The Principles of Sustainable Software Engineering - Learn | Microsoft Docs](https://docs.microsoft.com/en-us/learn/modules/sustainable-software-engineering-overview/)

[How Green Is Your Software? (hbr.org)](https://hbr.org/2020/09/how-green-is-your-software)

[Sustainable Software (microsoft.com)](https://devblogs.microsoft.com/sustainable-software/)

[#GreenConf - Sustainable Software Engineering Conference - YouTube](https://www.youtube.com/watch?v=D-spTjqAswA&t=3324s)

[Home - Sustainable Web Design](https://sustainablewebdesign.org/)

[Green Software Foundation | GSF](https://greensoftware.foundation/)

*This document is an attempt to bring together my learning over the last few months on the topic of sustainable software engineering with the hope that it helps someone kick start their journey.*