Deloitte.

NEXT

Sustainable Requirements
Is your software eco-friendly?

Callum Jones & Vibhuti Kaushal

January 2024

MAKING AN IMPACT THAT MATTERS since 1845

Contents

Authors

Callum Jones

+44 7714 644527

callumjones@deloitte.co.uk

Vibhuti Kaushal

+44 7440 476756

vibhutikaushal@deloitte.co.uk

Introduction: The impending challenge

IT innovations in cloud computing coupled with widespread digitisation have led to a proliferation of internet usage. Estimates suggest that 65% of the world population has access to the internet, with somewhere between 16 – 43 billion devices connected to it.[1]

Sustainability has emerged as a pervasive challenge for societies around the globe. Our increasingly digitised economy, fuelled by our dependence on technology for routine daily tasks, brings sustainability to light in a new context: software.

Technology professionals should be ready to elevate their thinking, and champion this new dynamic of software delivery. Engineers should consider this with every requirement, every test, and every line of code written.

In a 2022 survey of 221 industry practitioners involved in software projects in Asia, across various application domains such as banking, finance, and management, it was clear that sustainability in software is ambiguous.[2]

The results show that even though sustainability is deemed important by 91% of practitioners, 92% were not able to identify sustainability requirements for software applications.[2]

The importance of requirements cannot be understated – it is suggested that 80% of the environmental impact of a product is determined in the design and requirement phase.[3]

This paper will examine how software impacts the environment, specifically looking at problems that can be solved through sustainable requirements and design.

"Our increasingly digitised economy, fuelled by our dependence on technology for routine daily tasks, brings sustainability to light in a new context: software."

Impact on the environment: Where sustainability meets software

What considerations should technology professionals be making? Consider two lenses:

Running the software

For software to be used by the public it needs to be accessible on a server. Cloud vendors such as Amazon's AWS own datacentres with hundreds of thousands of servers, and there are thousands of these facilities around the world. These servers store and exchange data constantly, computing programmatic functions round the clock.

The environmental system pressure here is twofold: firstly, datacentres emit many pollutants and harmful gas emissions which contribute to global warming – this applies to both running them and cooling them down. Despite a lack of consensus and varying estimates, between 0.6 – 3.5% of global greenhouse gas (GHG) emissions are attributed to cloud computing and datacentres.[4] For context, the global aviation industry is estimated to account for between 1.9 – 2.5% of global emissions.[5]

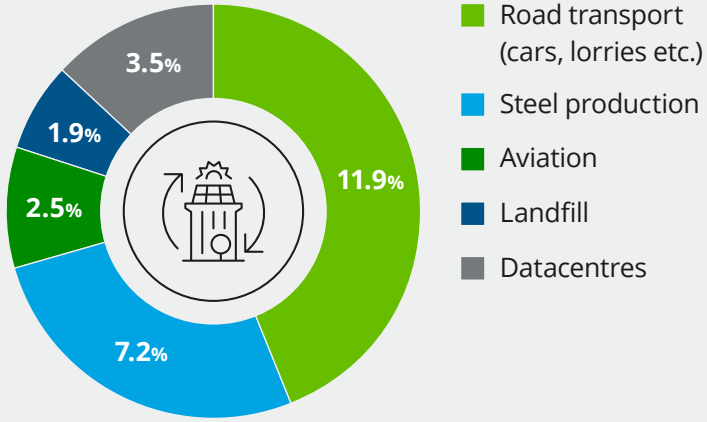


Figure 1: Percentage of global GHG emissions by industry. For illustrative purposes, upper bounds of estimated emissions have been used.[4, 5, 6]

Secondly, datacentres have significant energy requirements. It is estimated that energy consumption is set to rise from 2% to 8% of total global energy by 2030.[7]

The scale with which society uses digital products explains the high energy consumption and GHG emissions. It is estimated that at least 2.5 billion gigabytes of data is created every day, and that by 2025, there will be over 200 zettabytes, or 200 trillion gigabytes, of data in cloud storage.[8]

Many industries are reliant on datacentres to serve their users' digital needs, and indeed some industries have a larger environmental impact than others. According to the University of Cambridge, the Bitcoin network requires roughly 115 terawatt-hours of energy to maintain, equivalent to around twice the energy consumption of the entire nation of Switzerland.[7] And, as is well known, most electricity is produced through burning fossil fuels, usually coal.

In online streaming and distribution, the huge size of the digital files, as well as a thirst for higher and higher definition streaming, means large amounts of data is exchanged through global datacentres. Take one of the largest global online streaming services as an example: they have over 230 million subscribers, who on average watch around 3 hours of video per day.[9] The average high-definition video uses 3 gigabytes of data per hour. That is a potential 2 billion gigabytes of data exchanged through their servers per day, which equates to high energy usage and carbon emissions.

"Take one of the largest global online streaming services as an example: they have over 230 million subscribers, who on average watch around 3 hours of video per day.

Delivery of the software

Throughout the software delivery lifecycle there is ample opportunity for the delivery team to put processes in place to minimise environmental impact.

It is not uncommon for a large digital programme to run thousands of automated tests per day. For each one it is often standard practice to store test evidence; however, is that necessary? Storage of photos and videos is not free. Furthermore, the coding language chosen to write those tests in can also influence its sustainability, and indeed the coding language of the application itself.

Not forgetting the importance of requirements and design, the easiest way to make a product more sustainable is to design it that way from the start, not retrofit sustainability later! There is consensus among environmental practitioners that sustainable design is one of the most impactful ways to address sustainability. For example, designing a product in a modular way in which modules can be reused for other products once discarded. Doing analysis up front and defining the requirements clearly based on data is possibly the most impactful way to minimise the environmental impact of a product.

What prevents companies from building sustainable software?

Even though it is known that software can be made more sustainable, there are few companies taking action.

Below are a few of the key reasons:

Lack of acceptance and support

The IT industry is lacking the drive to act sustainably, and many companies are taking a passive approach to achieving it for their software. This may be due to IT companies seeing themselves as far removed from notoriously environment-damaging industries such as oil and gas extraction, or car manufacturers. Whilst this is true to an extent, the facts already presented show a clear case for finding the drive to act and the scale of impact mentioned.

Lack of knowledge in the development team

On the ground, there is a lack of understanding of how to make software more sustainable. Whether this be through ways of working, tooling, infrastructure, functionality of the software, or all of the above, development teams need to have a new mindset, and a new skillset, in order to develop software sustainably. The current software engineering knowledge base does not endorse sustainability as it does other areas such as quality, data management and security.

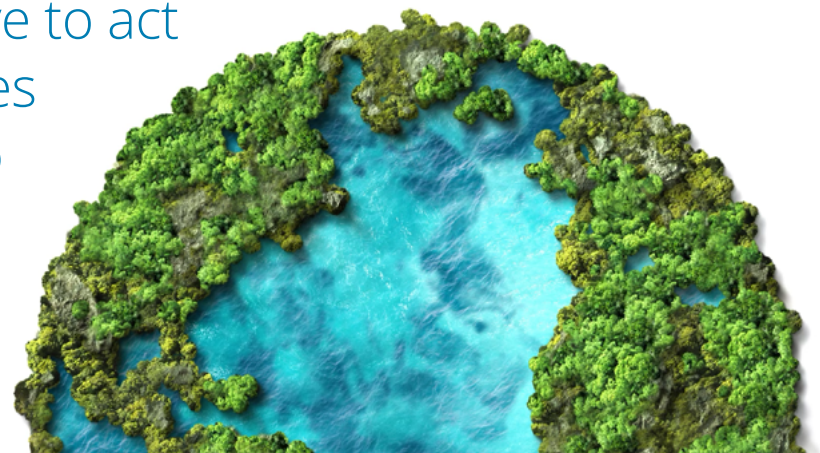
Additional costs

Senior leaders across a number of industries face the same dilemma – it costs money to act sustainably.

On the contrary, there is an increased appreciation amongst senior leaders that building sustainability into the business will provide it with resilience against the future and even reap a financial benefit. It is acknowledged by top financial investors that “climate risk is investment risk”. For example, a magazine publisher may benefit from having a digital arm to its business to insulate against possible future consumer trends, or even legislation, against deforestation.

What has been articulated so far is that business and society may see a significantly positive environmental impact when sustainability is taken into consideration in design and development of software.

“The IT industry is lacking the drive to act sustainably, and many companies are taking a passive approach to achieving it for their software.”



Writing sustainability into requirements

Business Analysts, Requirement Engineers, Product Managers, and Developers alike can use this analysis as a starting point to think about sustainability from the outset, and not allow it to become an afterthought in the development of applications.

What advice is there to write more sustainable requirements? Sustainability can fit in to both functional and non-functional requirements as follows:

Functional requirements

As mentioned, functional requirements are driven by the industry and context. There is a clear opportunity for more research in this area, and some industry-specific application guidance for functional requirements. Here are a few examples to consider:

- The Sustainable Web Design framework suggests that there is an opportunity for sustainable software when designing any application with a front end. **The use of images and videos**, for example, increase **the size of a webpage** and therefore the amount of data that needs to be passed across the internet. Sustainable requirements should minimise the presence of these to only those with a justified purpose. The framework also suggests to implement a "page weight budget" for a website, outlining a maximum byte limit for each page, which can be set in the requirement stage.[10]

- The practice of **approximate computing** is when operation of the software allowably does not compute an accurate result, suitable for uses in contexts where it is not necessary to provide one.[11] This is done to improve efficiency by, for example, minimising data transfer. For example, a video conferencing application may not need to stream video in a high number of frames per second to meet users' requirements for the call, hence this is something that may be investigated through user research in the requirements stage.

- Building an **Eco-Mode** into software applications can empower the user to choose to make the software more sustainable. Think of a website that allows the user to switch into eco-mode, and by doing so removes all pictures and videos from the site taking it down to barebones. This is already present on streaming sites where the user can choose to watch in lower quality (admittedly this is not intended for the purpose of saving the planet, but for times of low internet bandwidth).

Businesses can further benefit from thinking about sustainable design principles such as circular economy and waste elimination, and try to build such characteristics in to the software itself. For example, a location-based software application should consider fuel saving options.

Non-functional requirements

Arguably a new branch of non-functional requirement (NFR): sustainability should be considered alongside performance, reliability, accessibility, security and other NFRs. Here are some key topics to consider:

- **Hardware utilisation:** This involves ensuring proper use of infrastructure resources, such as load balancers, to ensure not only availability but efficiency in use of infrastructure, and thus sustainability. The Green Software Foundation explains that to improve hardware efficiency, one might use as few servers as possible, with the servers running at the highest utilisation rate.[12]

  The first benefit of this is maximising energy efficiency. The proportionality between server utilisation and energy consumption is not linear: at 0% utilisation, a computer can draw 100 watts; at 50% it draws 180 watts; and at 100% it draws 200 watts.[12]

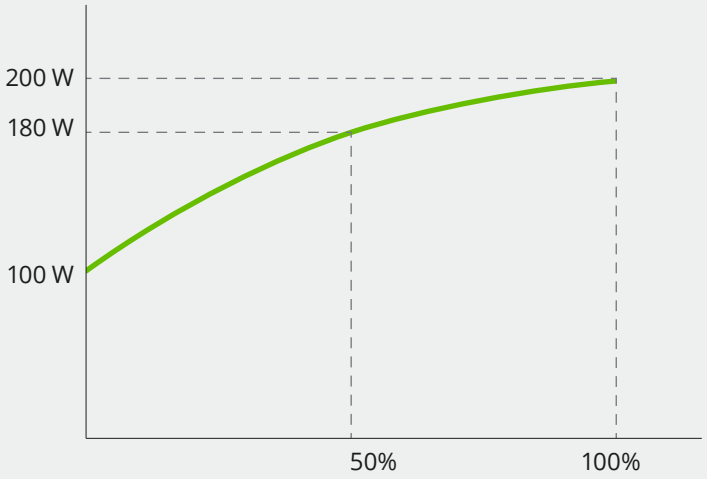


Figure 2: The non-linearity of utilisation and power.[12]

"To improve hardware efficiency, one might use as few servers as possible, with the servers running at the highest utilisation rate."

The second benefit to operating a lower number of servers relates to an important and often overlooked consideration for any hardware device: embodied carbon. This considers the carbon emitted during the manufacture of a device. For a server this is low relative to the carbon emitted during use, but for a desktop computer it is considerably more.

For servers, operating at lower utilisation means that an application's workload is spread across a larger number of servers – imagine running 4 servers at 25% utilisation rather than 1 server at 100%. Underutilisation is common in software – a survey of 100 companies found that more than half were underutilising their cloud resources, operating at CPU below 40%.[13] The issue is that the more servers used, the larger the cumulative embodied carbon attributed to running the software.

The amortization of the embodied carbon (considering the spread of the embodied emissions across the lifetime of a device) is an extension of this concept that an eco-friendly technology professional should understand. Most of the carbon emissions for end-user devices occur during its manufacture – see the Figure 3 below.[12]

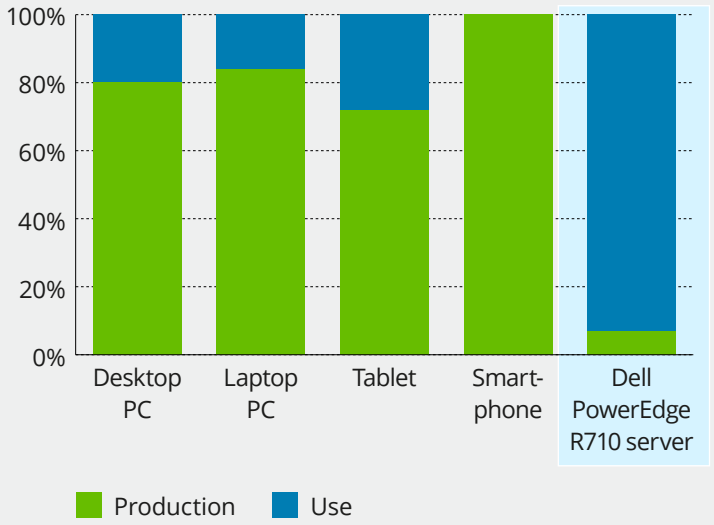


Figure 3: Annual carbon emission proportions from ICT devices. Note the server circled in red is measuring the total emissions across its average lifetime, whereas the end-user devices are measured annually.[12, 14]

Considering the high proportion of embodied carbon in these devices, it is vital to make them last for as long as possible. A software delivery team can contribute to extending the life of the device by simply allowing their software to be run on older device models and OS versions. This will spread the amortization of the device over a larger number of years, which would reduce annual emissions for that device.

Furthermore, a desktop produces 15 times greater annual emissions than a smartphone. This means that making software available on a smartphone as well as, or instead of, a desktop can also make the software more eco-friendly.

- **Energy efficiency:** Knowing how much energy the software uses to fulfill its functionality is important. In agile development this can be learned through spikes, rapid prototyping, experimentation or an equivalent.

  Take an example from AI application development: researchers trained an AI model to classify iris flowers. The model identified 96.17% of species with only 964 joules of energy. But they found that to achieve higher accuracy the system consumed a disproportionately higher amount of energy – to increase accuracy by 1.74% energy consumption increased around 300%. Knowing the non-functional impact allowed the researchers to know what the sustainable requirement should be set to for mass consumption of this technology – around 96% accuracy.[7]

"But they found that to achieve higher accuracy the system consumed a disproportionately higher amount of energy – to increase accuracy by 1.74% energy consumption increased around 300%."

- **Carbon efficiency:** This is sometimes an organisational prerequisite but can also be considered on a product basis if there will be dedicated infrastructure: using infrastructure at the least "carbon intensive" times and locations. Understanding the energy market is complex, however there are carbon intensive times to demand energy from a supplier (in other words: times when the supplier is using more fossil fuel sources to power its grid), and there are times where more comes from renewable energy.[12]

  Practically, this means either moving computation away from areas that have high carbon intensity (for example, the UK during winter gets very little sunlight for solar power, so shift computation to somewhere near the equator for more sunlight hours); or changing the amount of computation performed locally during carbon intensive hours (if carbon intensity is low, increase the demand, or in essence, do more in the applications). The latter of these options will only apply to permissible contexts.

  If your cloud vendor does not offer this as a service then can be complex to implement, it would require knowledge of energy suppliers across your IT infrastructure estate, and knowing times and locations of when they source energy from carbon intensive fossil fuels or carbon free sources such as renewables.

There are impactful ways to make software more sustainable before requirements or even a line of code is written – having clean and green infrastructure. Using a cloud vendor that offers ways to manage sustainable usage is an option every business should consider – many of the largest vendors do offer something.

"There are impactful ways to make software more sustainable before requirements or even a line of code is written – having clean and green infrastructure."

Conclusion

- This paper has narrowed the focus of sustainable software on to one critical aspect of the SDLC: requirements and design. There is acknowledgement that this requires a new mindset for a technology professional, a mindset that is currently scarce even in platform development teams that work with IT infrastructure every day. So, what is the next step?

- The first step is to start to incorporate sustainable thinking into conversations with delivery teams: in three amigos sessions, a design workshop, in a stand-up, or even in a coffee catchup.

- From there understand the environmental impact of products being worked on and consider the actions in this paper as a starting point for making improvements.

"Remember that even the most ambitious journeys start with a single step."

Endnotes

1 Cisco, Annual Internet Report, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf

2 Norman et al., https://www.mdpi.com/2071-1050/14/14/8596/htm

3 European Commission, Sustainable Product Policy, https://joint-research-centre.ec.europa.eu/scientific-activities-z/sustainable-product-policy_en

4 Climateiq, https://www.climatiq.io/blog/measure-greenhouse-gas-emissions-carbon-data-centres-cloud-computing

5 EESI, https://www.eesi.org/papers/view/fact-sheet-the-growth-in-greenhouse-gas-emissions-from-commercial-aviation

6 Our World in Data, https://ourworldindata.org/emissions-by-sector

7 Harvard Business Review, https://hbr.org/2020/09/how-green-is-your-software

8 Cyber Crime Magazine, https://cybersecurityventures.com/the-world-will-store-200-zettabytes-of-data-by-2025/

9 Statista, https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/

10 Sustainable Web Design, https://sustainablewebdesign.org/

11 Science Direct, Approximate Computing, https://www.sciencedirect.com/topics/computer-science/approximate-computing

12 Green Software Foundation, https://learn.greensoftware.foundation/

13 Forbes, https://www.forbes.com/sites/forbestechcouncil/2021/05/03/renewable-energy-alone-cant-address-data-centers-adverse-environmental-impact/?sh=24fdcaf45ddc

14 Dell, https://i.dell.com/sites/content/corporate/corp-comm/en/documents/dell-server-carbon-footprint-whitepaper.pdf

Deloitte.

Designed and produced by 368 at Deloitte. J31790