Uncovering the Environmental Impact of Software Life Cycle

ICT4S 2023

Thibault Simon, Pierre Rust, Romain Rouvoy, Joël Penhoat

ICT environmental footprint

ICT environmental footprint

Between 2.1% and 3.9% of global GHG emissions[1]

[1] C. Freitag *et al.*, "The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations", *Patterns*, vol. 2, no. 9, p. 100340, Sep. 2021.

ICT environmental footprint

84% of platinum and 70% of cobalt resources are located in high-risk contexts[1]

Credit: Dillon Marsh

[1] É. Lèbre *et al.*, "The social and environmental complexities of extracting energy transition metals", *Nature Communications*, vol. 11, no. 1, p. 4823, Sep. 2020.

ICT environmental footprint

Only 1.8 of the 9.2Mt of e-waste generated between 2014 and 2019 was officially documented as properly collected and recycled[1]

Credit: Muntaka Chasant

[1] V. Forti *et al.*, "The global e-waste monitor 2020", *United Nations University (UNU), International Telecommunication Union (ITU) & International Solid Waste Association (ISWA), Bonn/Geneva/Rotterdam,* vol. 120, 2020.

Software environmental footprint

While ICT impact comes from hardware,
responsibility lies within software

Literature on software environmental footprint mostly considers:

Literature on software environmental footprint mostly considers:

A single impact category, **climate change**[1, 2]

Literature on software environmental footprint mostly considers:

A single impact category, **climate change**[1, 2]

Only the energy consumed by resources, **not their embodied impact**[1, 2]

Literature on software environmental footprint mostly considers:

A single impact category, **climate change**[1, 2]

Only the energy consumed by resources, **not their embodied impact**[1, 2]

Software usage impact, **omitting its development**[3]

[1] J. Taina, "How Green Is Your Software?", in *Software Business*, 2010, pp. 151-162.

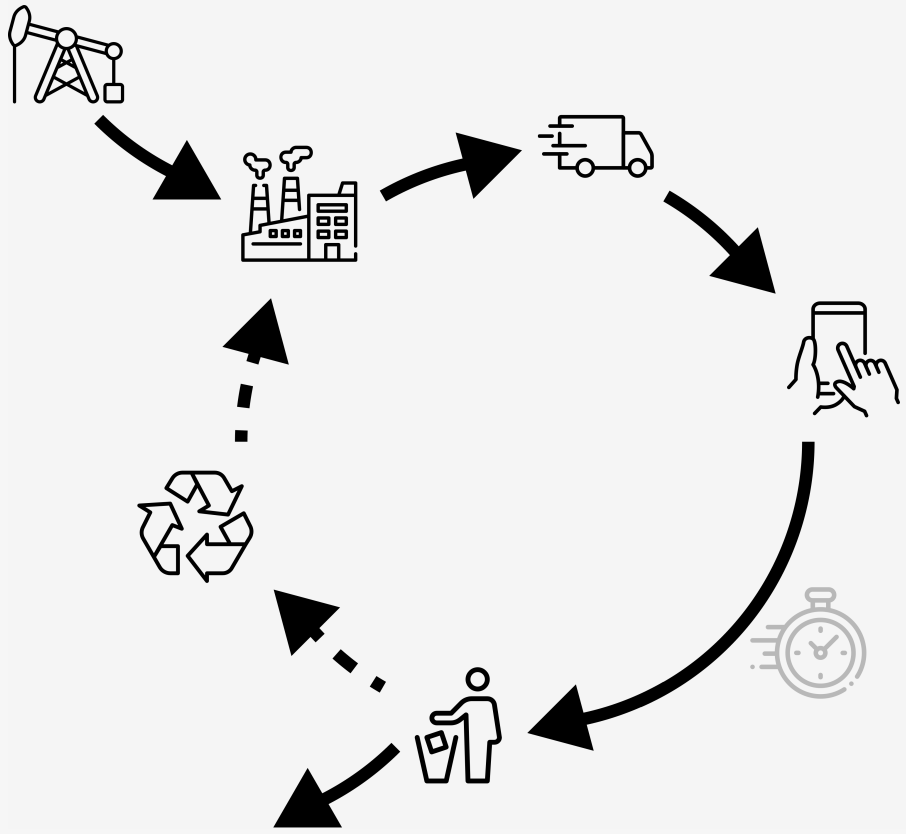
[2] E. Kern *et al.*, "Impacts of software and its engineering on the carbon footprint of ICT", *Environmental Impact Assessment Review*, vol. 52, pp. 53-61, Apr. 2015.

[3] L.1410 : Methodology for environmental life cycle assessments of information and communication technology goods, networks and services, 2014

The benefits of a life cycle perspective

Life Cycle Analysis (LCA):

The study of the environmental impacts contribution of a product or service across its entire life cycle[1]



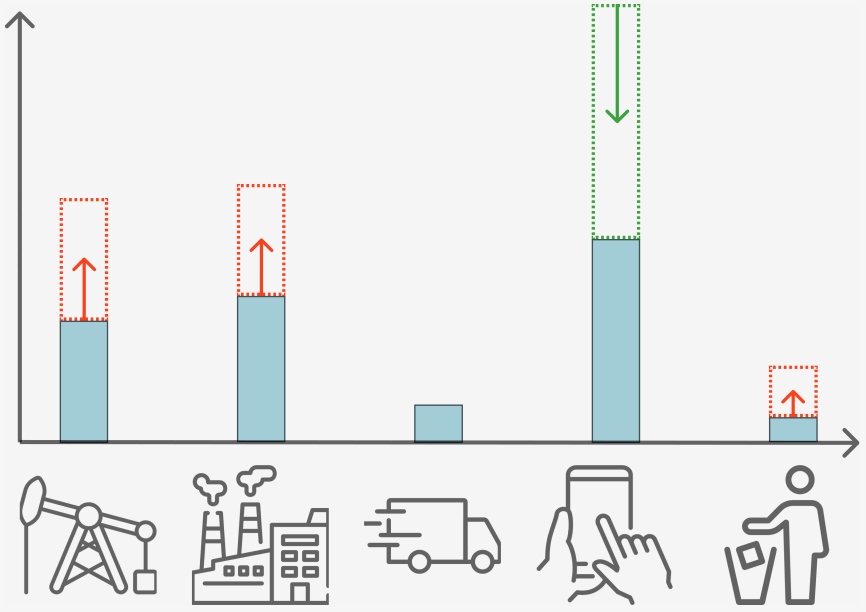
1. ISO 14040:2006: Environmental management - Life cycle assessment - Principles and framework

The benefits of a life cycle perspective

Identify the **shifting** of a potential environmental burden:

The benefits of a life cycle perspective

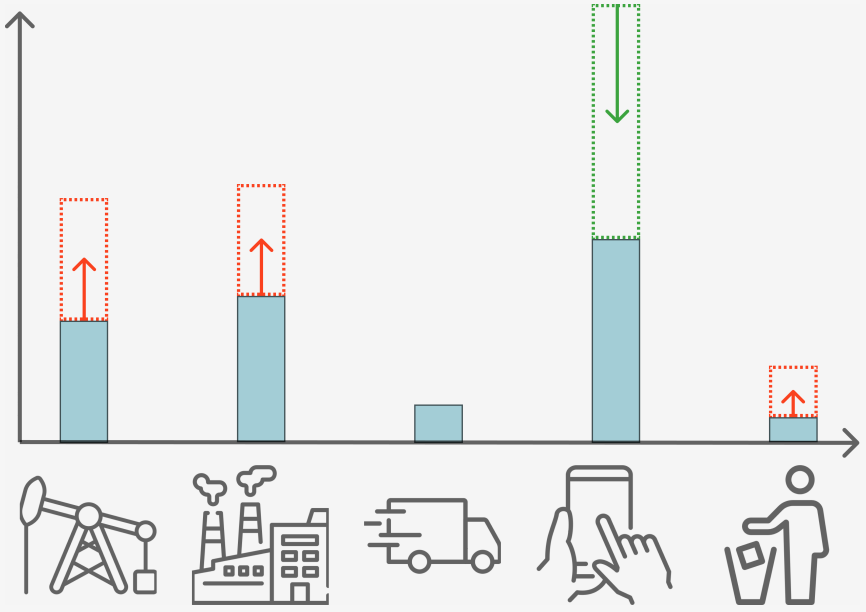
Identify the **shifting** of a potential environmental burden:



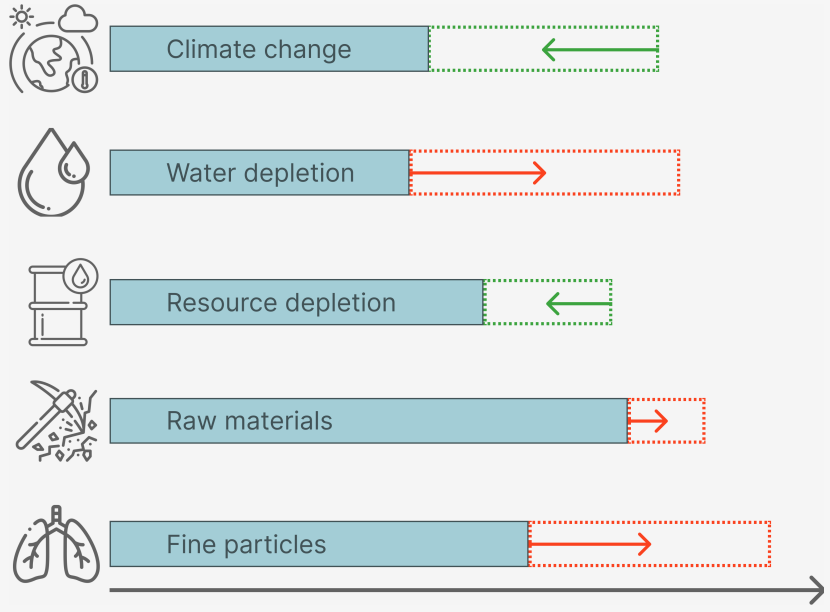
1. Between **phases**

The benefits of a life cycle perspective

Identify the **shifting** of a potential environmental burden:



1. Between **phases**

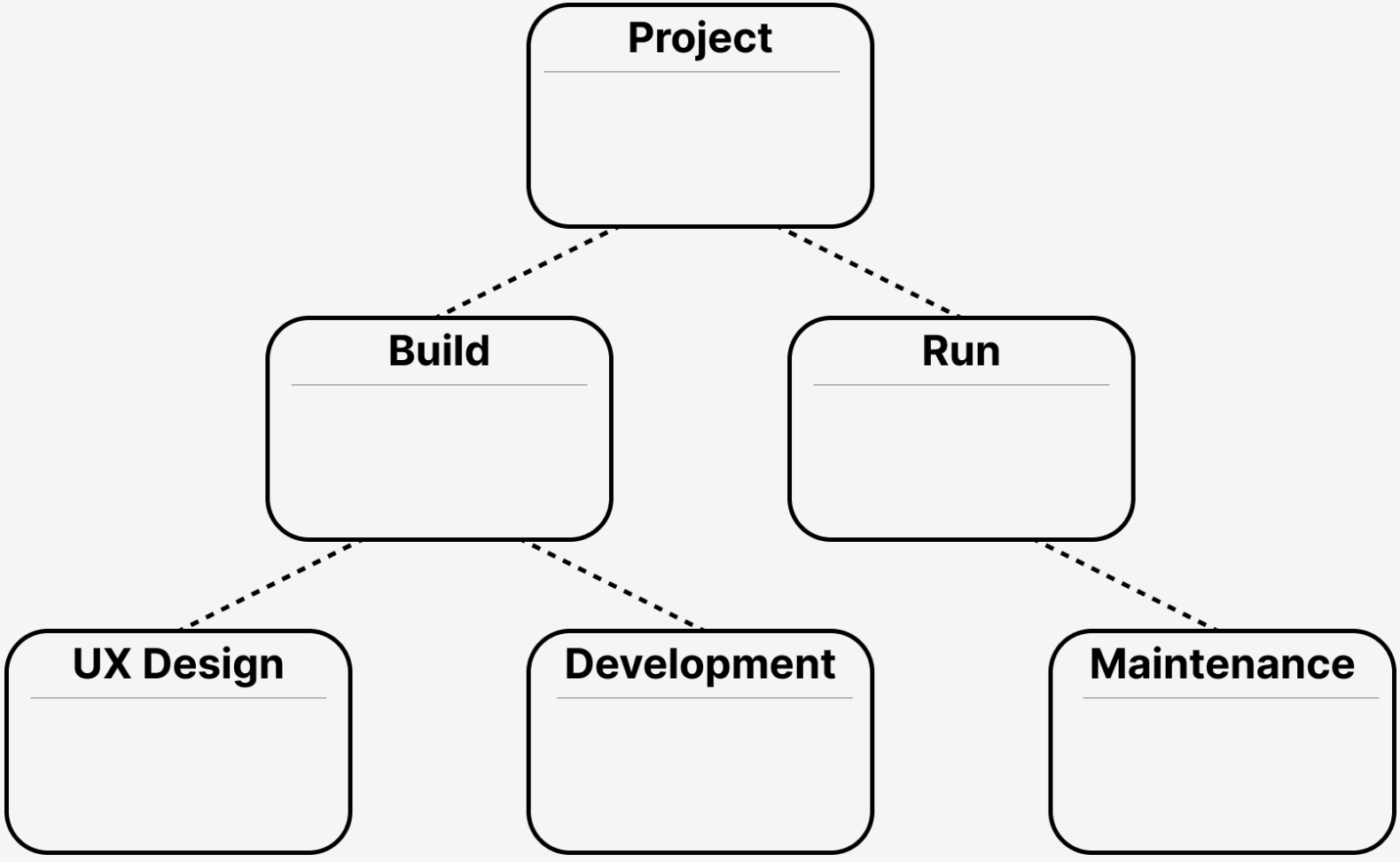


2. Between **categories**

Applying such a holistic approach to software

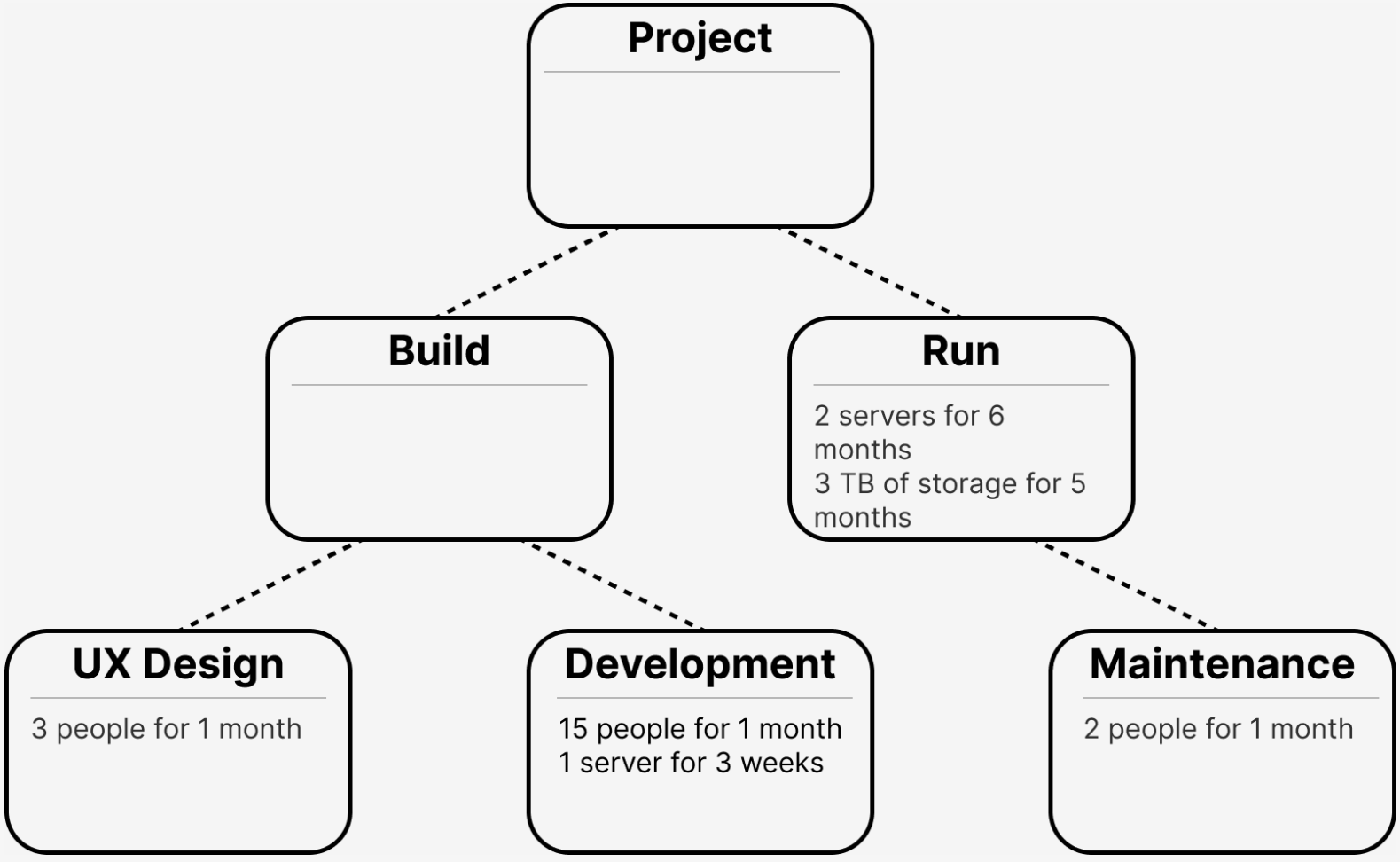
Modeling methodology

Activities



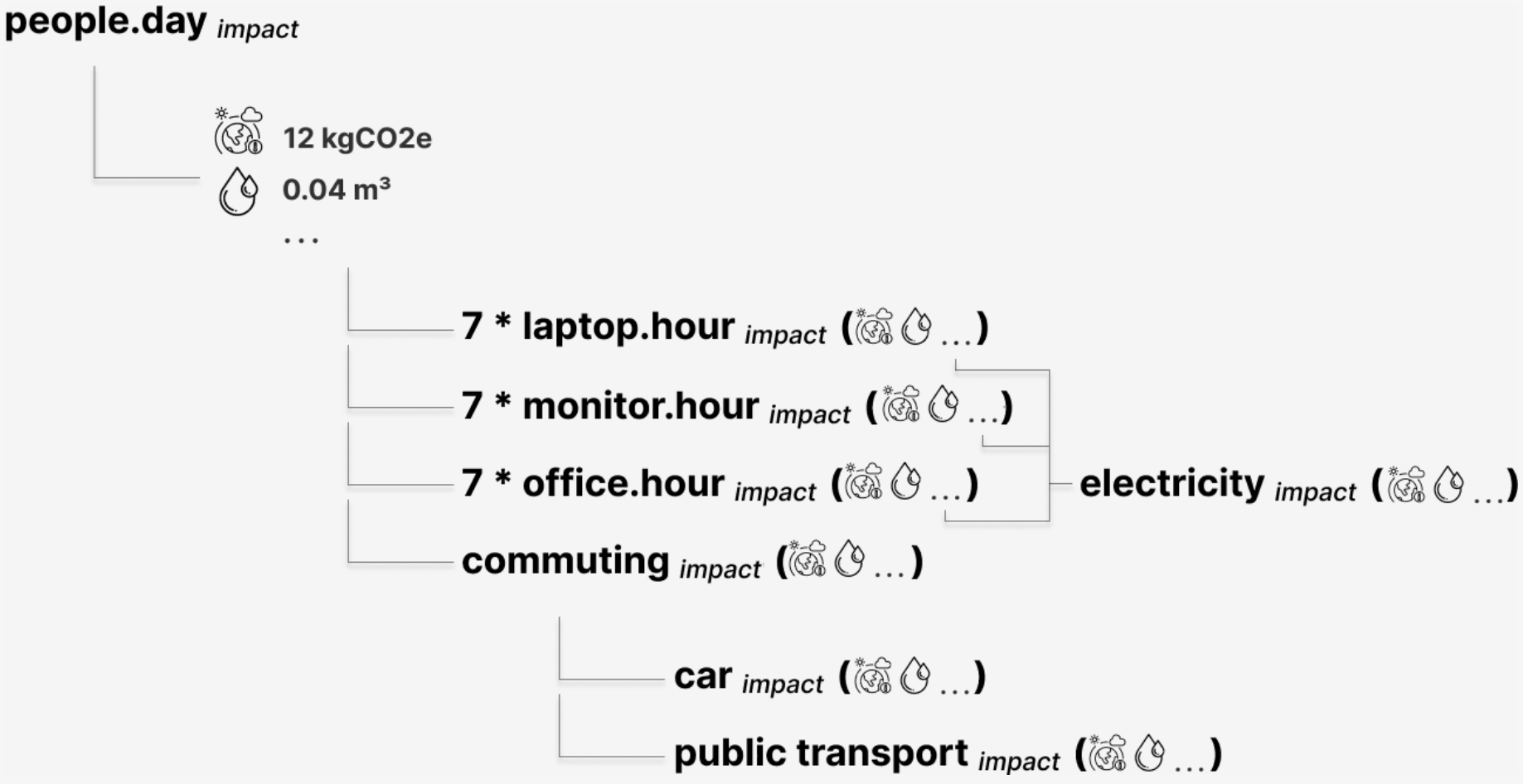
Modeling methodology

Resources



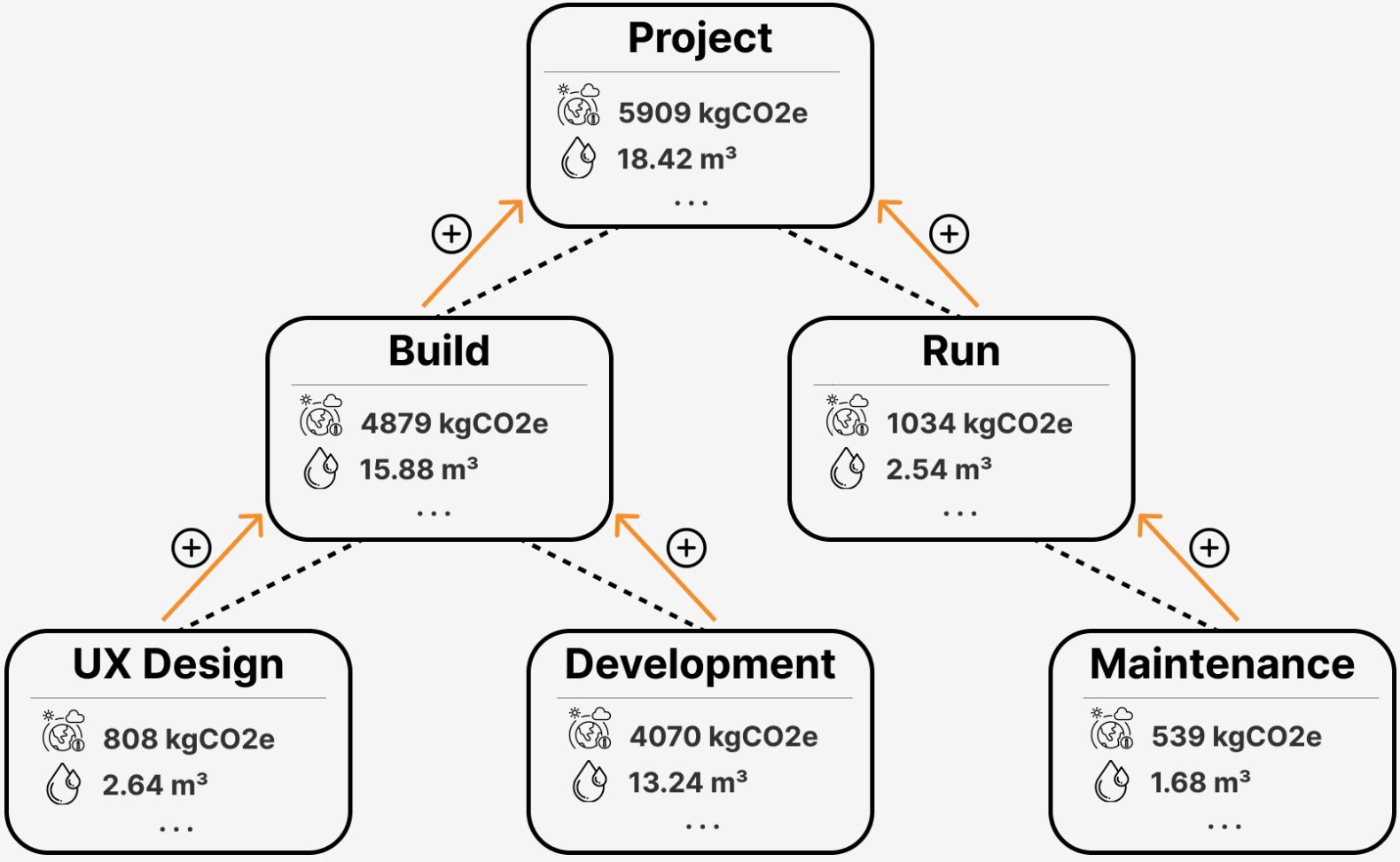
Modeling methodology

Resources environmental impact



Modeling methodology

Activities impact



To uncover impact shifts

To uncover impact shifts

Sample use case

GitLab project, as **representative** of software size and complexity

To uncover impact shifts

Sample use case

GitLab project, as **representative** of software size and complexity

A **post-mortem approach** (COCOMO) on the repository Source Line of Code to estimate the effort distribution for the ***build*** phase

To uncover impact shifts

Sample use case

GitLab project, as **representative** of software size and complexity

A **post-mortem approach** (COCOMO) on the repository Source Line of Code to estimate the effort distribution for the ***build*** phase

Using the **hardware requirements** for a 50 000 users instance for the ***usage*** phase

To uncover impact shifts

Sample use case

GitLab project, as **representative** of software size and complexity

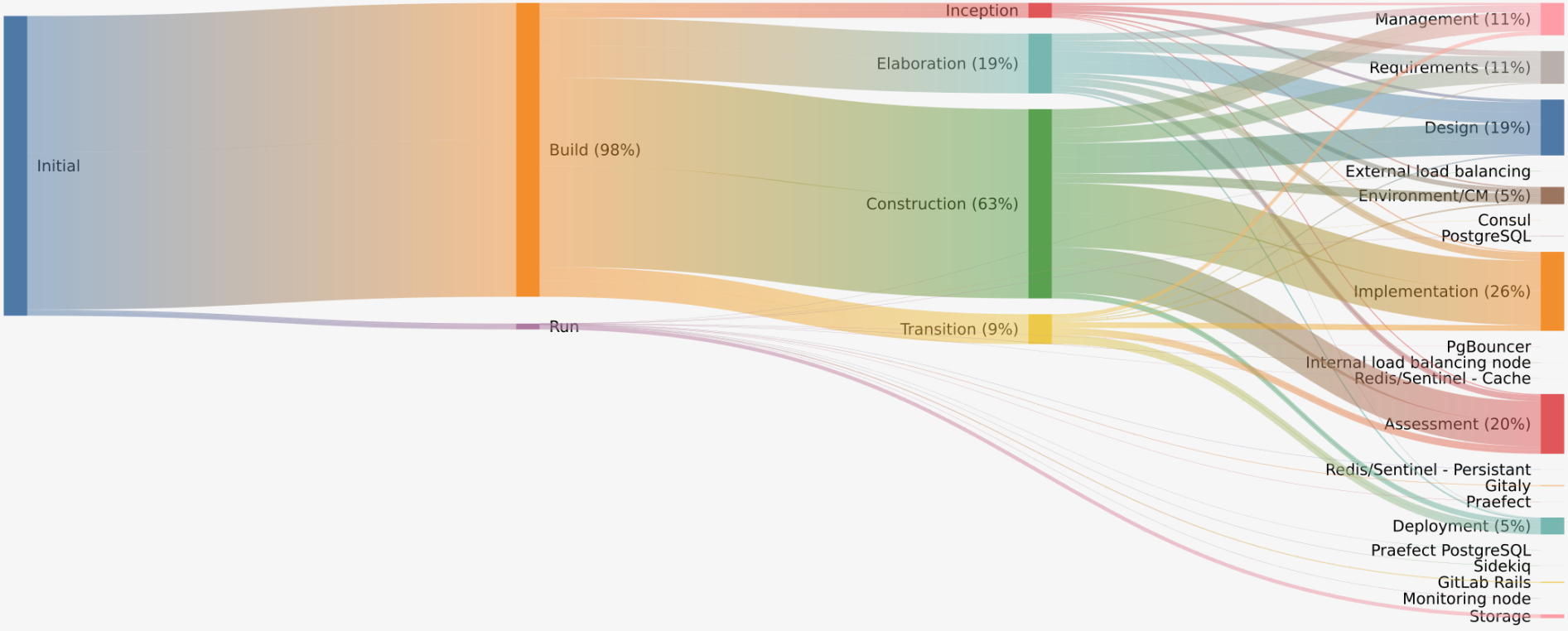
A **post-mortem approach** (COCOMO) on the repository Source Line of Code to estimate the effort distribution for the ***build*** phase

Using the **hardware requirements** for a 50 000 users instance for the ***usage*** phase

Developped and hosted in **France**

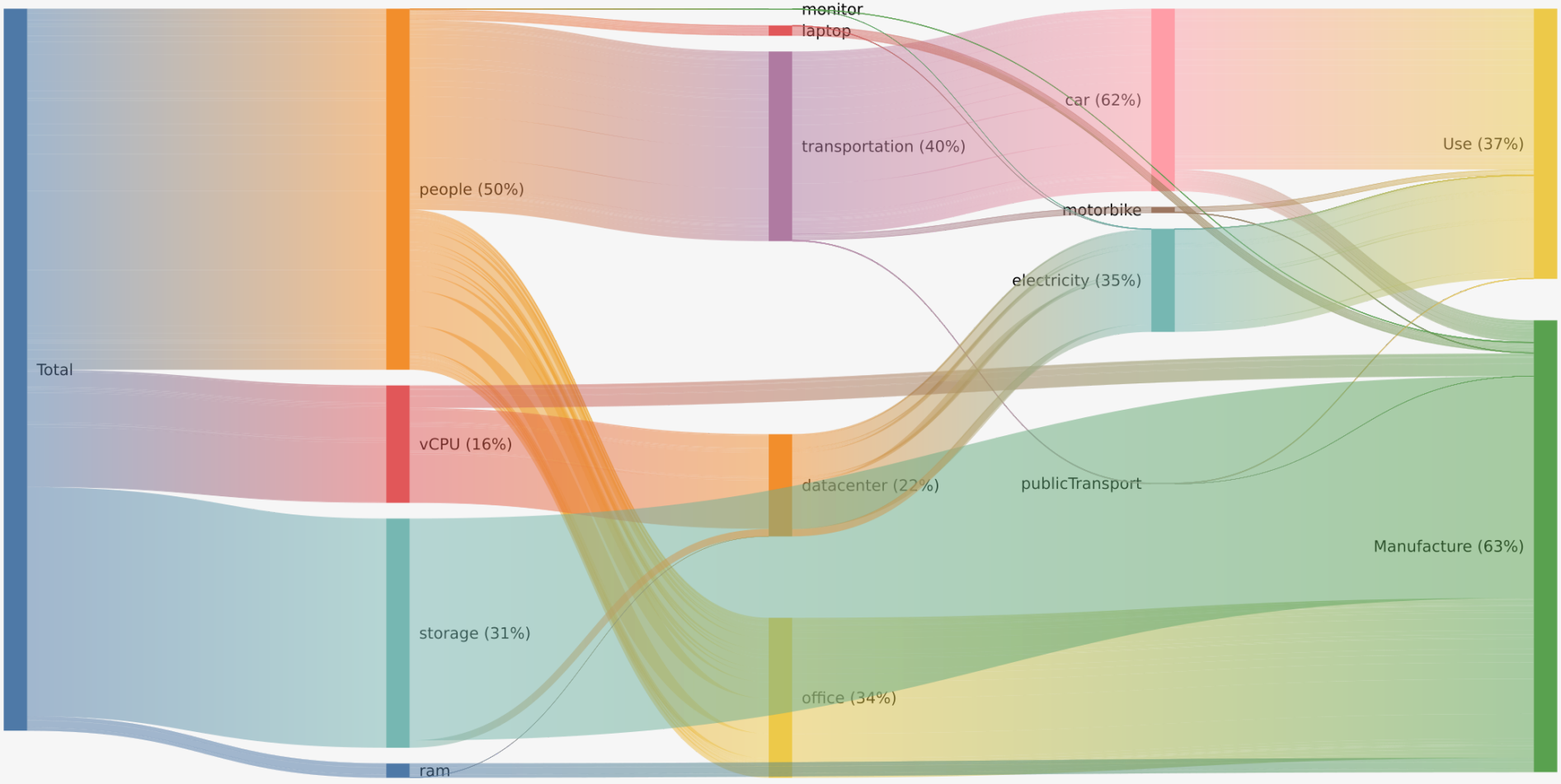
To uncover impact shifts

1. Between phases



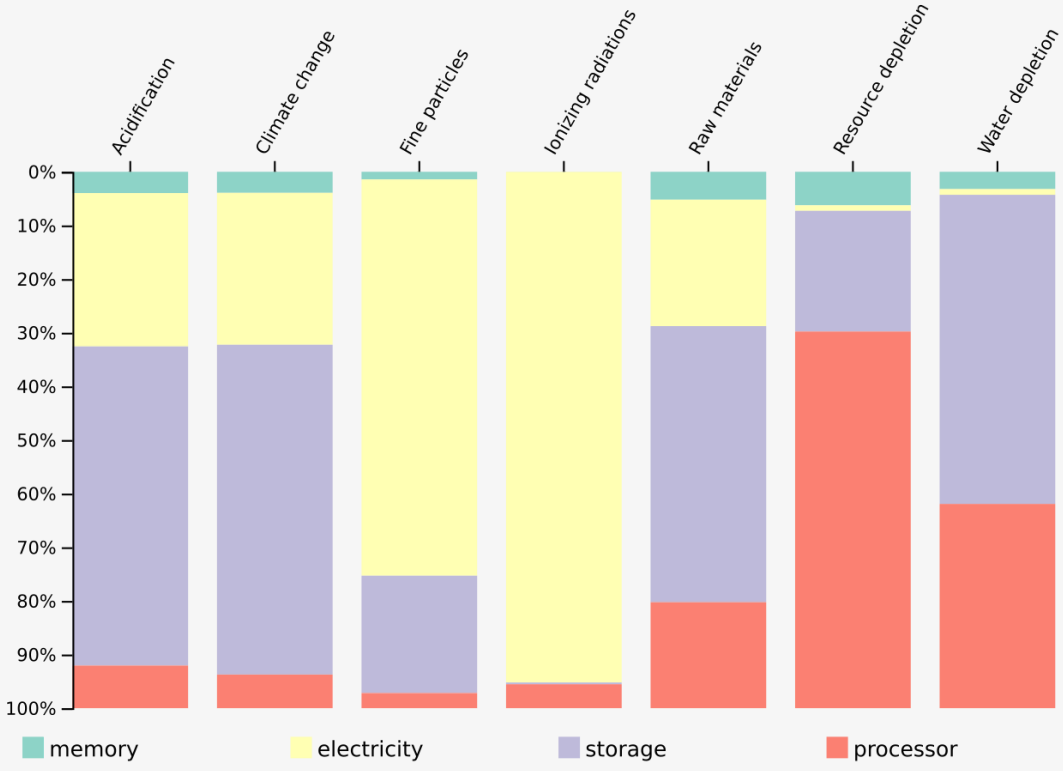
To uncover impact shifts

2. Between resources

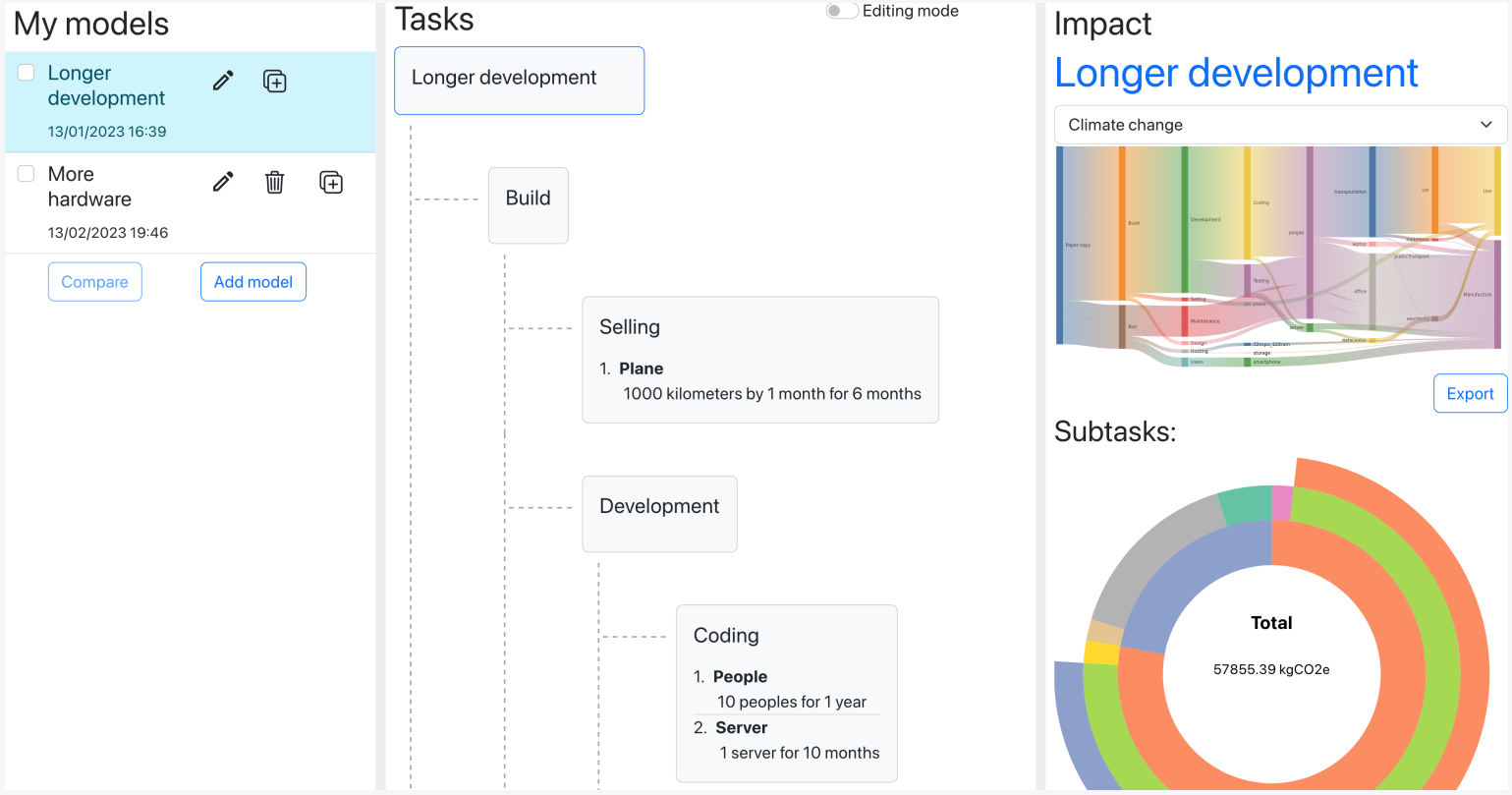


To uncover impact shifts

3. Between impact categories



Web application



Soon available under https://github.com/Orange-OpenSource

Conclusion

A **methodology** and the associated tool to model **software life cycle environmental impact**

To identify **shifts** and identify **hotspots** to take sourced decisions to **reduce software environmental impact**

The ecosystem **lacks data on the environmental footprint** of the resources it uses

Thank you