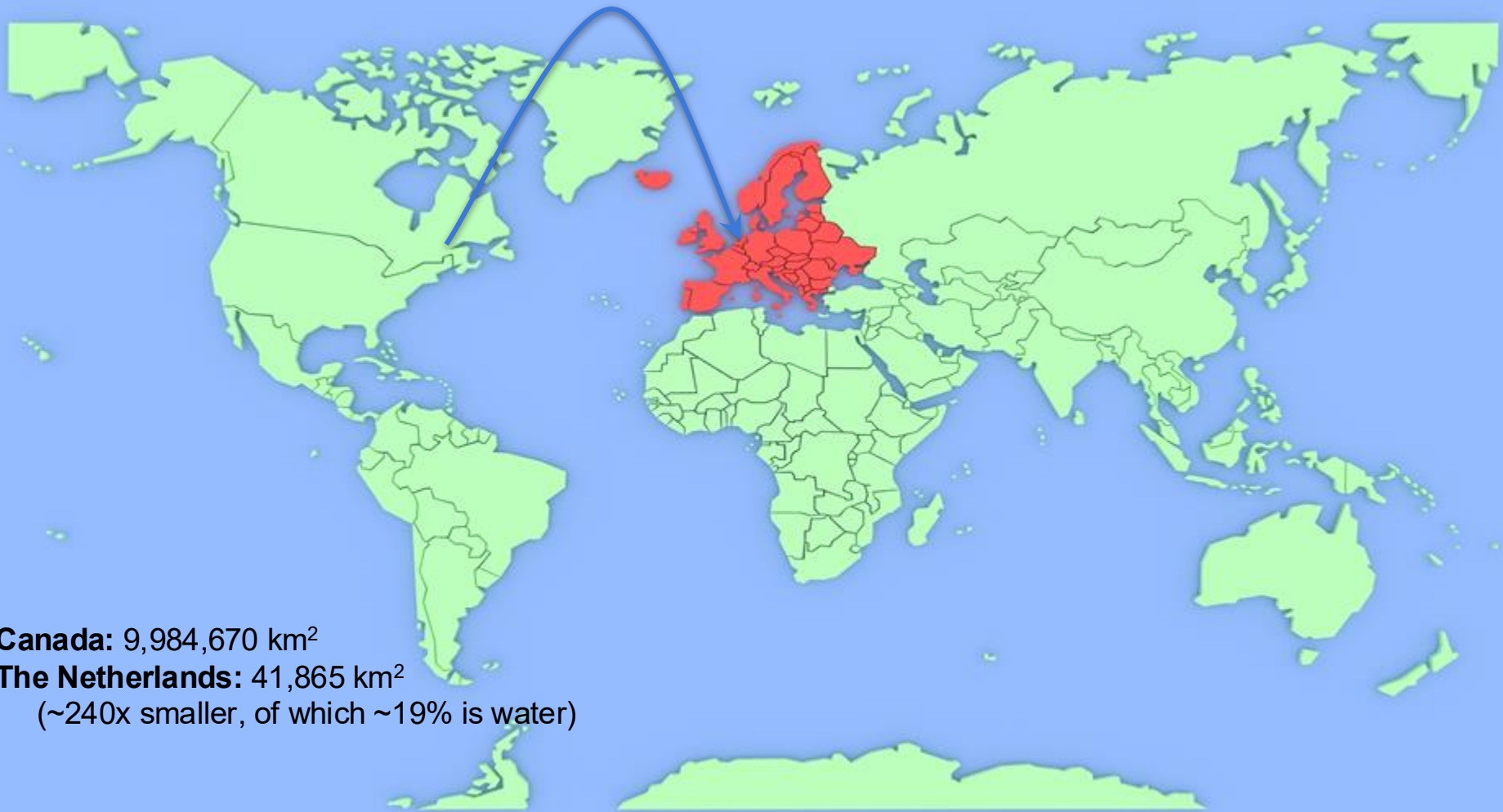


Software Sustainability and its Engineering: How far have we come?

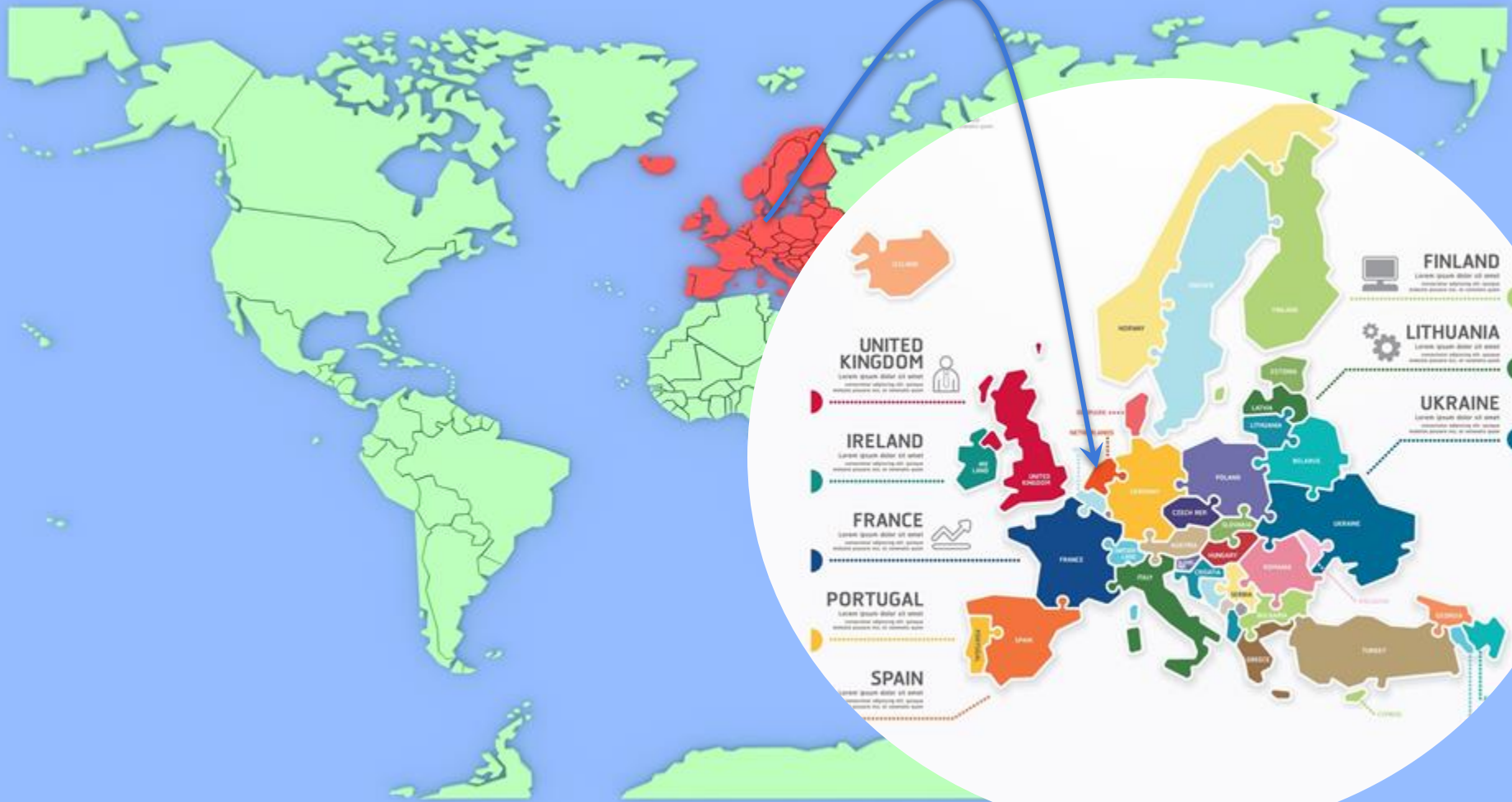
Patricia Lago
Vrije Universiteit Amsterdam

a disclaime **R**





Canada: 9,984,670 km²
The Netherlands: 41,865 km²
(~240x smaller, of which ~19% is water)



UNITED KINGDOM

IRELAND

FRANCE

PORTUGAL

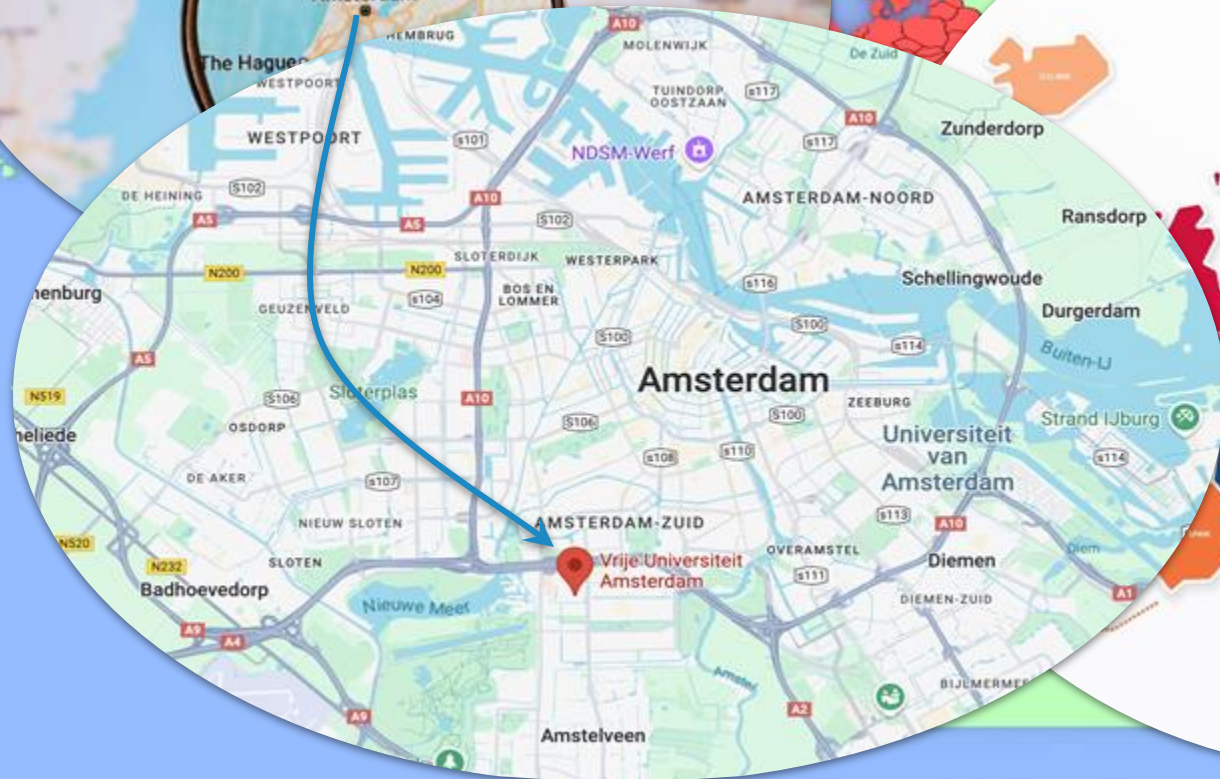
SPAIN

FINLAND

LITHUANIA

UKRAINE







Vrije Universiteit Amsterdam - VU Amsterdam - Zuidas



32,000 STUDENTS
4800 EMPLOYEES
139 BSc/MSc
PROGRAMS

7 FACULTIES
24 INTERDISCIPLINARY
INSTITUTES

[Discover our Campus](#)
[NU University Building](#)



Vrije Universiteit Amsterdam - VU Amsterdam - Zuidas





The S2 group @ VU Amsterdam

s2group.cs.vu.nl



Prof. Dr. Patricia Lago
Full Professor



Dr. Ivano Malavolta
Associate Professor



Dr. Eritzá Guzmán
Assistant Professor



Dr. Ilias
Gerostathopoulos
Assistant Professor



Dr. Justus Bogner
Assistant Professor



Dr. Vincenzo Stoico
Postdoc Researcher



Dr. Klervie Toczé
Postdoc Researcher



Dr. Remco de Boer
Research Fellow - Digital
Architecture



Zubaria Inayat
Junior Lecturer



Abhishek Iyer
Junior Lecturer



Anjana M S
PhD Student



Lauren Olson
PhD Student



Markus Funke
PhD Student



Elvin Alberts
PhD Student



Iffat Fatima
PhD Student



Joran Leest
PhD Student



Anna Fischer
PhD Student



Faezeh Amou
Najafabadi
PhD Student



Tom Humbert
PhD Student



Keerthiga Rajenthiram
PhD Student



Rumbidzai Chitakurwe
PhD Student



Engelbert Fellinger
PhD Student

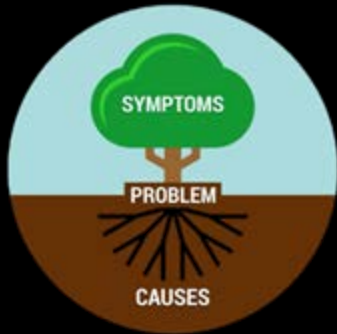


Bouazza El Moutaouakil
PhD Student



Muhammad Imran
Guest PhD Student

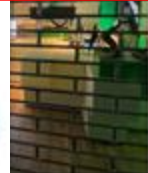
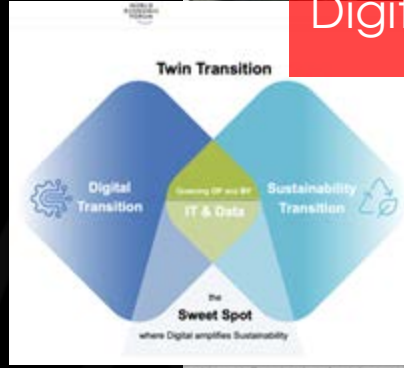
The flow



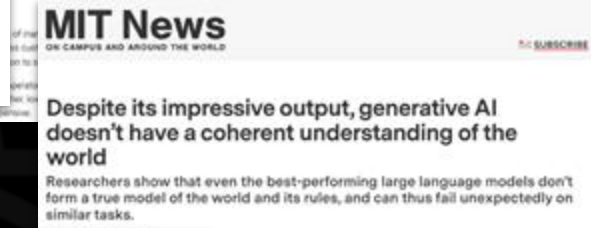


The Problem

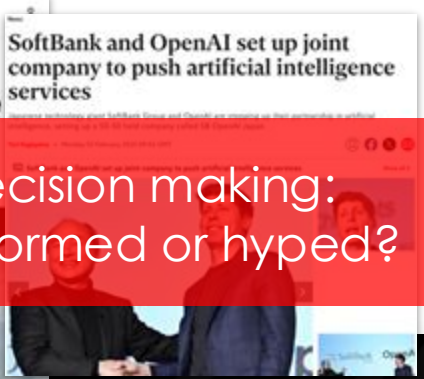
Digital comes to a cost



Digital transformation of global society: inclusive?



How digitalisation can leave the poor behind
In the pandemic, governments created digital solutions for access to social support — a risk for those on the wrong side of the digital divide



Decision making: informed or hyped?

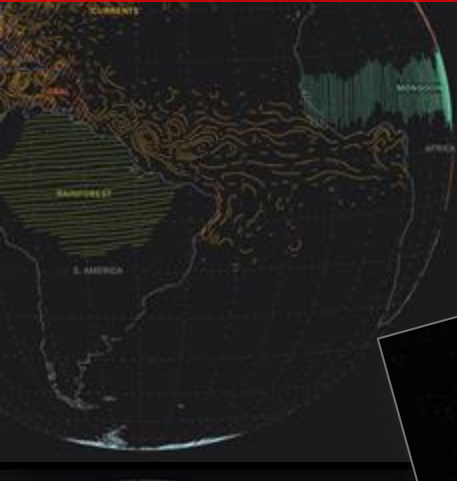
Innovation ≠ Distributive quality of life



The Symptoms

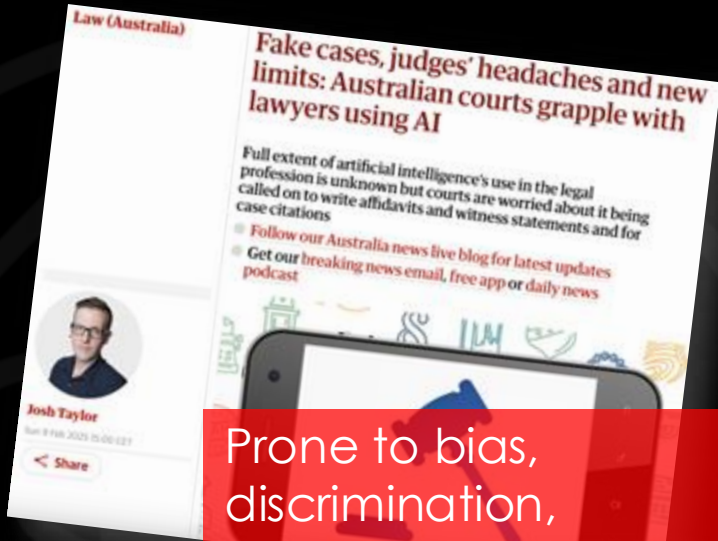
The New York Times

Planet Earth rebels



How Close Are the Planet's Climate Tipping Points?

Earth's warming could trigger sweeping changes in the natural world that would be hard, if not impossible, to reverse.



Prone to bias, discrimination, misuse



Misinformation and disinformation, Adverse outcomes of AI technologies



The Causes



"I have witnessed **large parts of society** turn their heads and deliberately **reject the truths that have been presented to them**. The rise of unreason has been the unwelcome partner to our growing scientific sophistication." [Robin McKie, Science editor, the Observer]



Denial



Feeling Powerless



5 things we can learn about leadership and leadership communication from New Amsterdam's Dr Max Goodwin



Sia Papageorgiou FRSA, FCSCE, SCMP
Strategic Communication Consultant, Trainer and Coach Helping
Communication Professionals and Business Leaders Amplify Their

May 26, 2021



We can take action

We must take action

"You know, we all feel like **the system is too big to change, but guess what? We are the system, and we need to change.**" [Dr. Goodwin, New Amsterdam, Netflix series]



The past: where do we come from?

Sustainability in general

Dimensions

FIGURE 1: THE THREE PILLARS OF SUSTAINABILITY



<https://tiinyurl.com/y3cfec4>

Order of impacts



DIRECT IMPACT
(technology)

SYSTEMIC IMPACT
(change in behavior)

ENABLING IMPACT
(supported processes)

UN Sustainable Development Goals



<https://sdgessentials.org/why-the-world-needs-the-sdgs.html>

ID 181173892 © Alain Lacroix | Dreamstime.com



https://en.wikipedia.org/wiki/Environmental,_social,_and_governance

F. Berkhout and J. Hertin, "Impacts of information and communication technologies on environmental sustainability: Speculations and evidence," Report to the OECD, Brighton, vol. 21, p. 23, 2001. (archived)

Software sustainability: a definition



- 1) ... the “capacity to **endure**” – from UN Brundtland Report: Our Common Future (1987)
- 2) ... the “**preservation of the function of a system** over an **extended period of time**” – Lago & Penzenstadler (2017). Editorial: Reality check for software engineering for sustainability—pragmatism required. *Journal of Software: Evolution and Process*, 29(2).
- 3) ... the “**balance**” among relevant dimensions - and the relationship between micro-level actions and macro-level impacts [1]

Software sustainability is the preservation of the **long-term** and **beneficial use** of software, and its appropriate evolution, in a **context that continuously changes** [2]

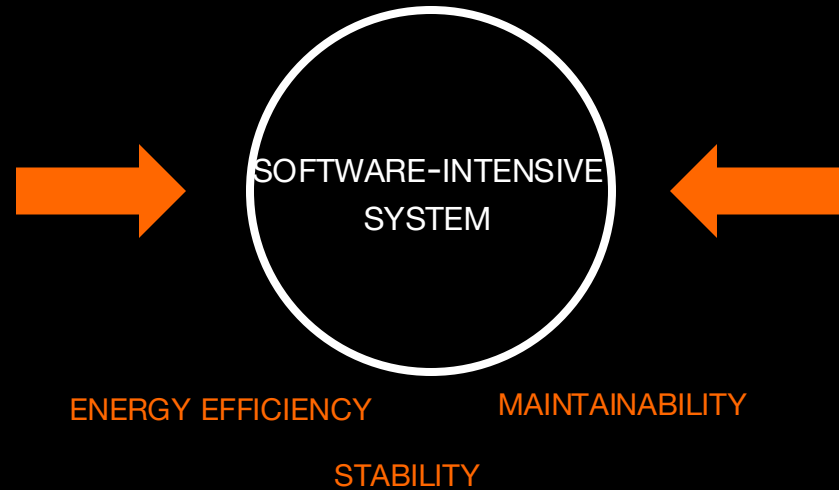
[1] L.M. Hilty and B. Aebischer (2015), “ICT for Sustainability: An Emerging Research Field,” in *ICT Innovations for Sustainability*, Springer, pp. 3–36.

[2] Lago, P. (2023). *The Digital Society Is Already Here – Pity It Is 'Unsustainable'*. In I. Vermeulen (Ed.), *Connected World: Insights from 100 Academics on How to Build Better Connections* VU University Press Amsterdam.

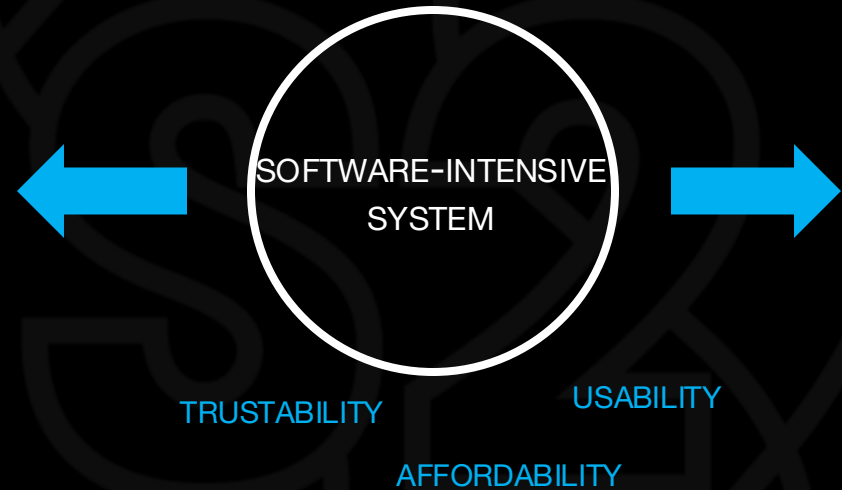
The role of software in sustainability: two perspectives



Inward looking



Outward looking



Direct impact
(sustainable software)

Indirect impact
(software for sustainability)

The role of software in sustainability: dimensions of **focus**, dimensions of **time**



SUSTAINABILITY

Energy-efficient
drone



Medical
emergencies



tech



env



econ



soc



focus

time

1

DIRECT

2

INDIRECT

3

SYSTEMIC



The role of software in sustainability:

dimensions of **focus**, dimensions of **time**, and **rebound effects**

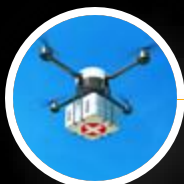


SUSTAINABILITY

Energy-efficient drone



Medical emergencies



- tech
- env
- econ
- soc

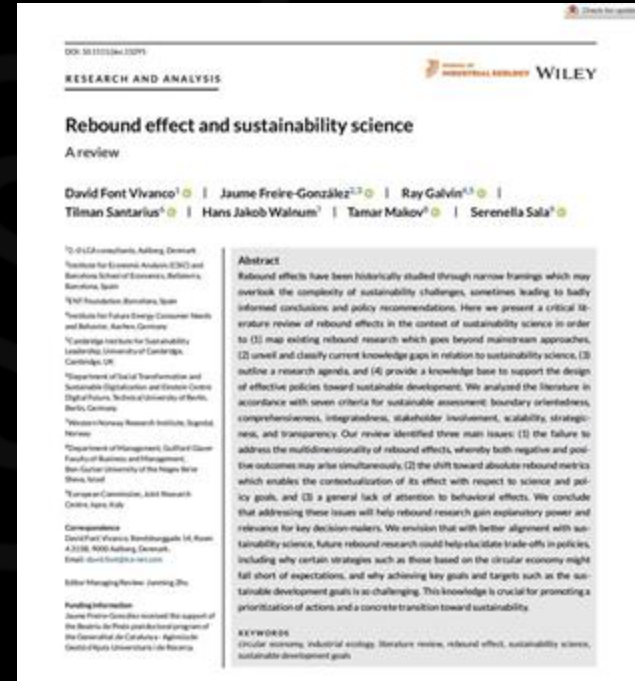


focus time

- 1 DIRECT
- 2 INDIRECT
- 3 SYSTEMIC

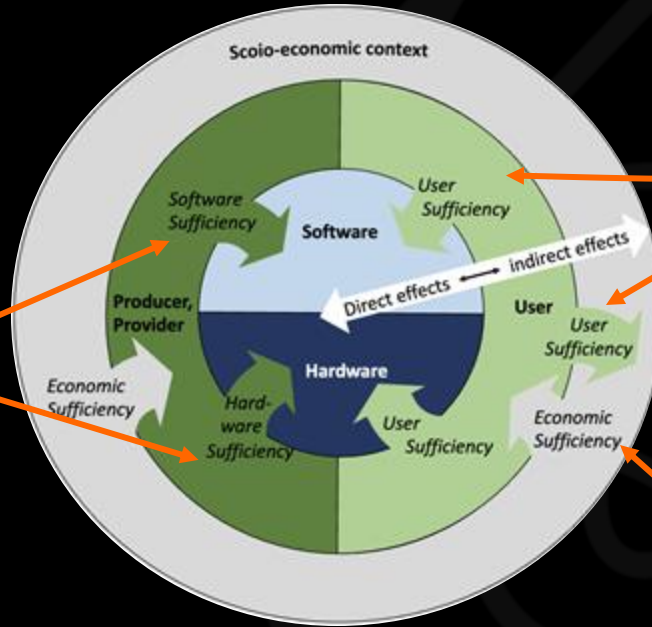


REBOUND EFFECTS (or JEVONS PARADOX)
(negate intended impacts)



The role of software in sustainability:

Digital sufficiency



Policies empowering the **user** in how ICT is used (e.g., focus) and for what purpose (e.g., collaboration)

Policies reducing the need for **HW** and **SW** (e.g., the right to repair)

Policies fostering an **economy** that addresses societal and individual needs (rather than growth) (e.g., reducing avg working hours)

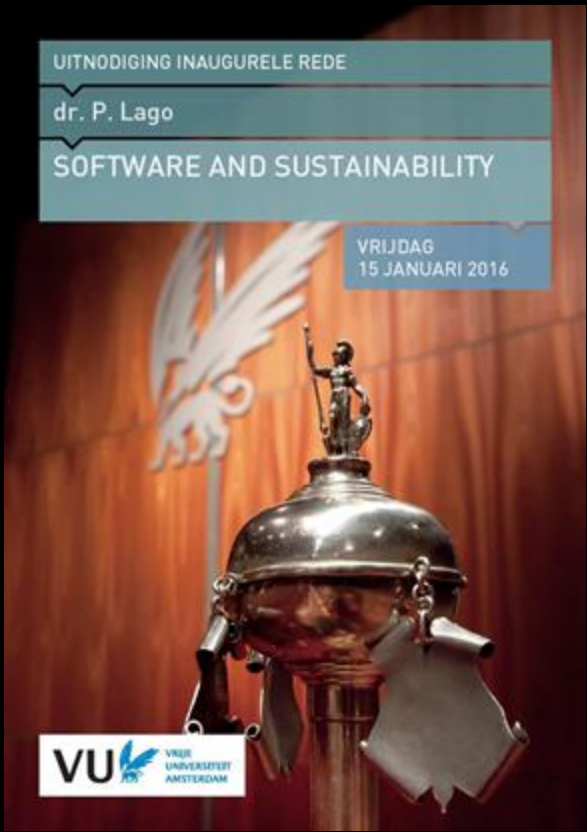
*Fig. 1: Interplay of the four dimensions of digital sufficiency

- (*) T. Santarius et al., "Digital sufficiency: conceptual considerations for ICTs on a finite planet," Ann. Telecommun., vol. 78, no. 5–6, pp. 277–295, 2023.
- L. Hilty, "Computing Efficiency, Sufficiency, and Self-sufficiency: A Model for Sustainability?," in Computing with Limits, Irvine, CA, USA, 2015.

The present: how far did we come?



UITNODIGING INAUGURELE REDE
dr. P. Lago
SOFTWARE AND SUSTAINABILITY
VRIJDAG
15 JANUARI 2016



The present: how far did we come?



The good



The bad



The mystery

Notion of sustainability	DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
Sustainability-aware sw/SE	TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
Sustainability-aware sw architecting/SA	DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY

The present: how far did we come?



The good



The bad



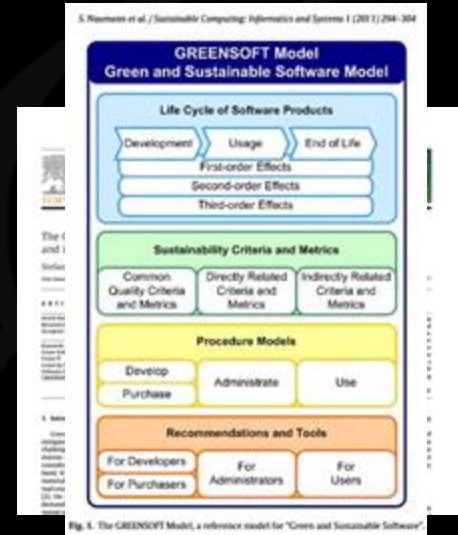
The mystery

Notion of sustainability

DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY

Sustainability-aware sw/SE

Sustainability-aware sw architecting/SA



GREENSOFT reference model (2011)

contributed articles

Framing Sustainability as a Property of Software Quality

BY PATRICIA LAJO, SEVEY AKINLI BICAK, ANICA CRNOVIC, AND DRISIT PENZENTSTABLER

This framework addresses the environmental, dimension of software performance, as applied here by a paper mill and a car-sharing service.

capital and value. The social dimension is concerned with increasing communities, the environmental dimension seeks to improve human well-being by promoting natural resources, and the technical dimension is concerned with supporting business and well-being of software systems. Sustainability is understood only when accounting for all dimensions, including the environmental dimension. It is possible to see an environmental perspective and environmental dimension in the context of software systems. "Considerations among the four dimensions, namely technical, design, social and sustainability." Practical software engineering research only needs software developers must understand the relationship among goals of the four dimensions.

The abstracting of system with more engineering research will lead to sustainability in that the technical and economic dimensions are taken into account while the environmental and social dimensions are not. The question we address here is how these concepts relate to software and how to break down the engineering research into software-specific requirements.

4D sw sustainability (at ICSE GREENS 2013, Published in 2015)

The present: how far did we come?



The good



The bad



The mystery

Sustainability Awareness Framework (SusAF)

Lessons Learned from Developing a Sustainability Awareness Framework for Software Engineering Using Design Science

STEFANIE BETZ, Fortwangen University and LUT University, Fortwangen, Germany
 BIRGIT PENZENSTADLER, Chalmers University of Technology and LUT University, Gothenburg, Sweden
 LETICIA DUBOC, La Salle, University Ramon Llull, Barcelona, Spain
 RUZANNA CHITCHYAN, University of Bristol, Bristol, UK
 SEDEF AKINLI KOCAK, Vector Institute for Artificial Intelligence, Toronto, Canada
 IAN BROOKS, University of the West of England, Bristol, UK
 SHOLA OYEDEJI, LUT University, Lappeenranta, Finland
 JARI PORRAS, LUT University, Lappeenranta, Finland
 NORBERT SEYFF, University of Applied Sciences and Arts Northwestern Switzerland, Windisch, Switzerland
 COLIN C. VENTERS, University of Huddersfield, Huddersfield, UK

Sustainability Assessment Framework (SAF) Toolkit

Software and Systems Modeling
<https://doi.org/10.1007/978-3-031-20270-024-01230-9>

THEME SECTION PAPER

The sustainability assessment framework toolkit: a decade of modeling experience

Patricia Lago¹ · Nelly Condori Fernandez² · Ifat Fatima³ · Markus Funke¹ · Ivano Malavolta¹

Received: 29 March 2024 / Revised: 28 August 2024 / Accepted: 17 October 2024
 © The Author(s), under exclusive license to Springer Verlag GmbH Germany, part of Springer Nature 2024

Abstract
 Software intensive systems play a crucial role in most, if not all, aspects of modern society. As such, both their sustainability

Notion of sustainability

Notion of sustainability	DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
	TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
	DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY

Sustainability-aware sw/SE

Sustainability-aware sw architecting/SA

The present: how far did we come?



The good

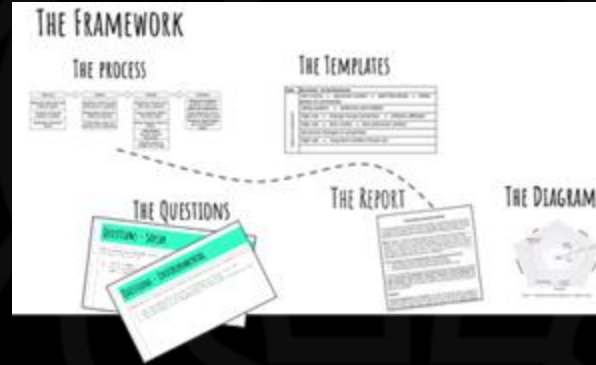


The bad



The mystery

Sustainability Awareness Framework (SusAF)



Sustainability Assessment Framework (SAF) Toolkit

Sustainability-aware sw/SE	TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
	DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY

MAIN INSTRUMENTS (OUTCOMES)	 The Decision Map (DM) for design decision making.	 The Sustainability-capability (SQ) Model to define concerns and measures.	 The KPI template as a crafting tool to define sound sustainability indicators.	
	GUIDING INSTRUMENTS	 A Checklist to help define the elements of a DM.	 The SQ Dependency Matrix as reusable catalog of SQ concerns.	 The Decision Graph to help assign the right impact timescale to DM elements.

The present: how far did we come?



The good



The bad



The mystery



Three time horizons: more an intuition than a solution

72/123 50/123
1/123 0/123

Human factors as an established QA Not really about sustainability, but rather societal implications

Notion of sustainability

DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY

Sustainability-aware sw/SE

Sustainability-aware sw architecting/SA

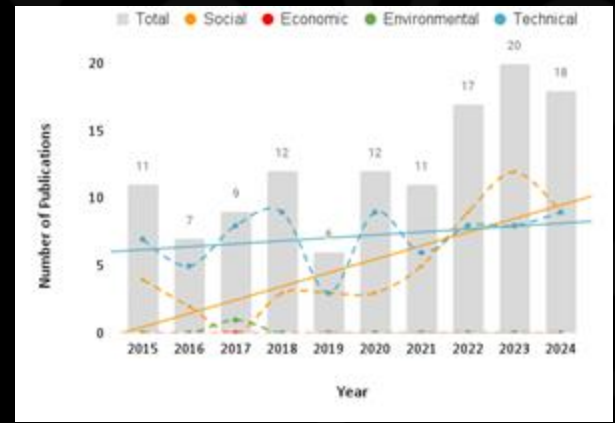


Fig.4. Direct focus of SEIS publications over the years (*)

(*) I. Fatima and P. Lago (2025), "Ten Years of Software Engineering in Society," in International Conference on ICT for Sustainability (ICT4S), IEEE. To appear.

The present: how far did we come?



The good



The bad



The mystery

Notion of sustainability	DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
Sustainability-aware sw/SE	TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
Sustainability-aware sw architecting/SA	DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY

- *Experiments: well designed and executed?*
- *Results: well described (and not over-stated)?*
- *Lessons learned: synthesized and made available?*

The present: how far did we come?



The good

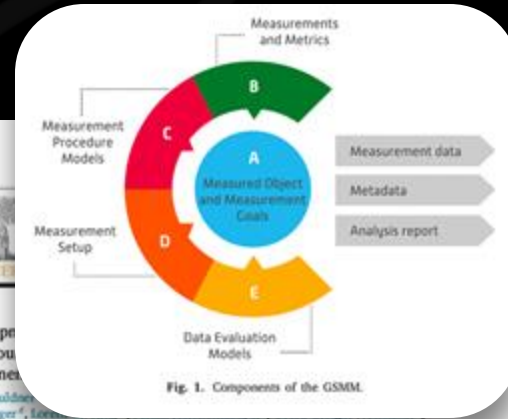


The bad



The mystery

Notion of sustainability	DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
	TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
Sustainability-aware sw/SE			
Sustainability-aware sw architecting/SA	DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY



Develop the resource components

Achim Gülden, Jens Gröger, Dennis Junger, Tom Kennes, Sandro Kreten, Patricia Lago, Franziska Mai, Ivano Malavolta, Julien Muraich, Kira Obergöcker, Benno Schmidt, Arne Tarasa, Joseph P. De Veough-Geiss, Sebastian Weber, Max Westing, Volker Wollgemuth, Stefan Naumann

Ten Years of Teaching Empirical Software Engineering in the Context of Energy-Efficient Software

Ivano Malavolta, Vincenzo Stoico, and Patricia Lago

Carbon-Efficient Software Design and Development: A Systematic Literature Review

Highlights

- H₁ Software developers are by far the most frequent stakeholders for the existing solutions to support the design and development of environmentally sustainable software.
- H₂ More support should be provided to software architects and sustainability engineers for the design of environmentally sustainable software.
- H₃ The design and development of environmentally sustainable software involves multiple different stakeholders, whose roles, contributions, and needed support should be further investigated.

Green Software Measurement Model (GSMM)

Sound experiments

Beyond sw developers

The present: how far did we come?



The good



The bad



The mystery

AS RESEARCHERS, WHAT SOCIETAL IMPACT DO WE WANT TO CREATE?



OPEN ARCHIVE (REUSABLE TACTICS)

add scientific rigour; share good practices

Notion of sustainability

Sustainability-aware sw/SE

Sustainability-aware sw architecting/SA

DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY



Software Carbon Intensity (SCI)



Resource-efficient software

- "Improve" over "use"
- Sound metrics, comparable measures



Benchmarks

The present: how far did we come?



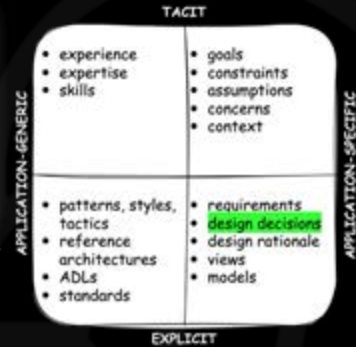
The good



The bad



The mystery



Notion of sustainability

DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
---------------------	-------------------------------------	-----------------

Standards work very well (thus far)^[1]

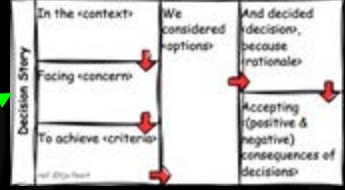
Sustainability-aware sw/SE

TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
------------	---------------------------------	------------------------

Simple Beneficial Useable^[2]

Sustainability-aware sw architecting/SA

DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY
-------------------------------	--------------------	---------------------------------



@CGI



@ABN AMRO

- [1] ISO/IEC/IEEE 42010:2022 including ethical and sustainability concerns
- Architectural Decision Records <https://adr.github.io/>
- [2] J. Maeda, The Laws of Simplicity. MIT Press, 2006.

The present: how far did we come?



The good



The bad



The mystery

Notion of sustainability

DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
---------------------	-------------------------------------	-----------------

Sustainability-aware sw/SE

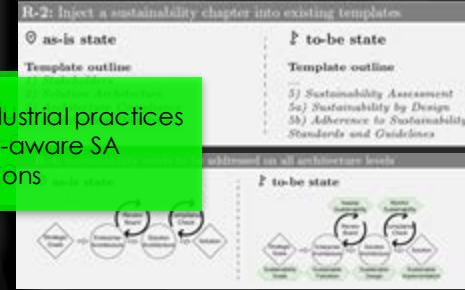
TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
------------	---------------------------------	------------------------

Sustainability-aware sw architecting/SA

DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY
------------------------	--------------------	---------------------------------



From SoTA+ industrial practices to sustainability-aware SA recommendations



- M. Funke and P. Lago, "Carving Sustainability into Architecture Knowledge Practice," in European Conference on Software Architecture (ECSA), LNCS, vol. 14212. Springer, 2023.
- Sustainable IT Lab VU-ABNAMRO: <https://digitalsustainabilitycenter.nl/pages/vu-abn-lab>



From industry context to sustainability-aware design decisions

- Precision and recall
- Resource-efficiency
- 4,911 ADRs, autom. metrics, human evals
- SLMs = small devices, in-house

- R. Dhar, K. Vaidhyanathan, and V. Varma, "Can LLMs generate architectural Design Decisions? An exploratory empirical study," in ICSE 2024, IEEE.
- -, (follow-up study upcoming)
- ArchBench: LLMs for Software Architecture Tasks. <https://www.sabench.com>

The present: how far did we come?



The good



The bad



The mystery

Sustainability can be represented by quality attributes (QAs) that have impacts across multiple dimensions^[1]



SA evaluation blueprint^[2]

- Multiple dimensions of focus
- Multiple QAs in Sust. Indicators

Sustainability Impact Score (SIS)

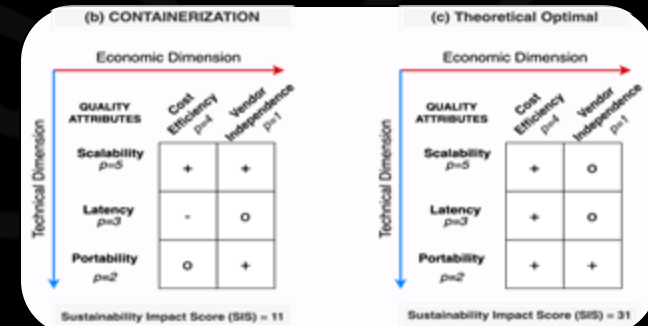
Quantified trade-offs between Sust. Dimensions^[3]

Notion of sustainability

Sustainability-aware sw/SE

Sustainability-aware sw architecting/SA

DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY



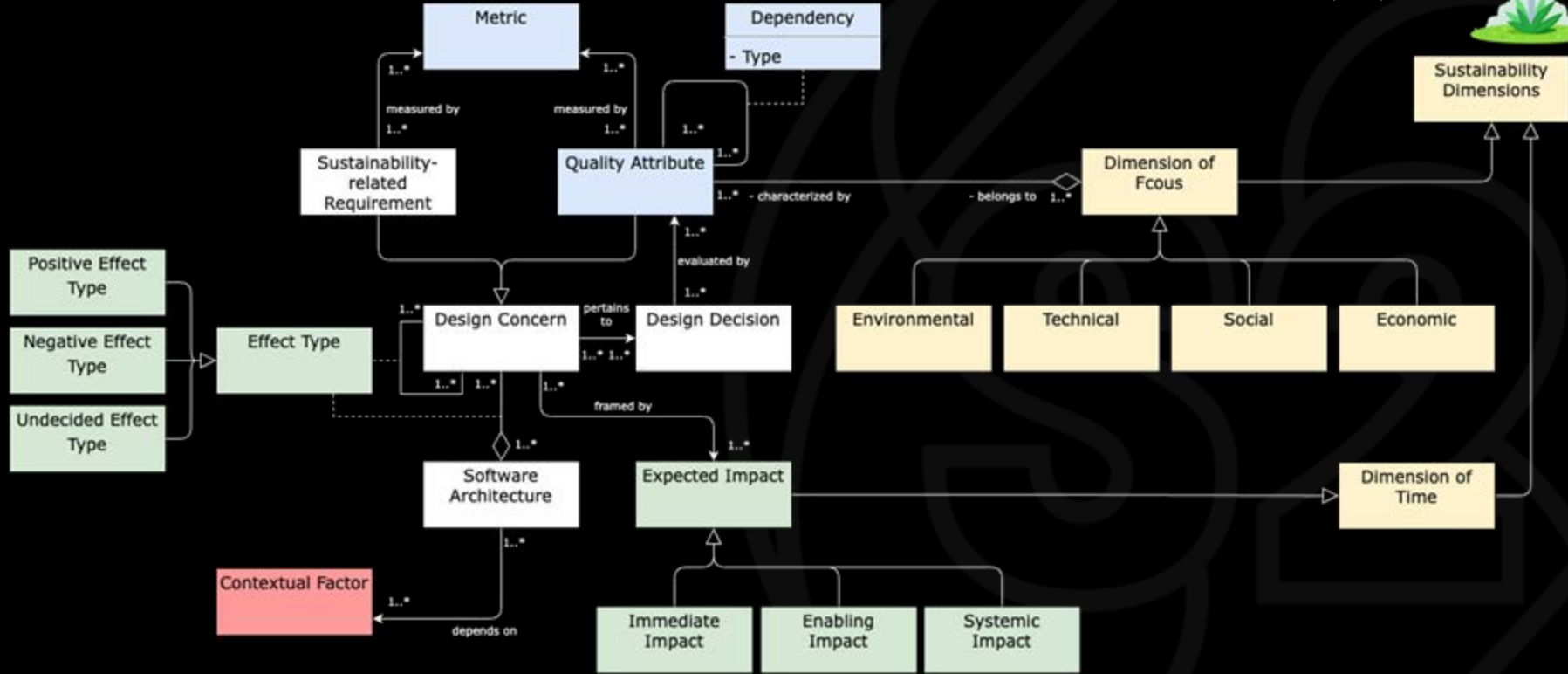
[1] N. Condori-Fernandez, et.al, "An Action Research for Improving the Sustainability Assessment Framework Instruments. In Sustainability, 12 (4), 2020.
 [2] I. Fatima and P. Lago, "Towards a sustainability-aware software architecture evaluation for cloud-based software services," in ECSA 2024 D. Symp, Springer.
 [3] I. Fatima, P. Lago, V. Andrikopoulos, B. v.d. Waaij, "Using sustainability impact scores for software architecture evaluation," in ICSCA 2025, IEEE.

The (desirable) future: where should we go?

Systemic design



The mystery

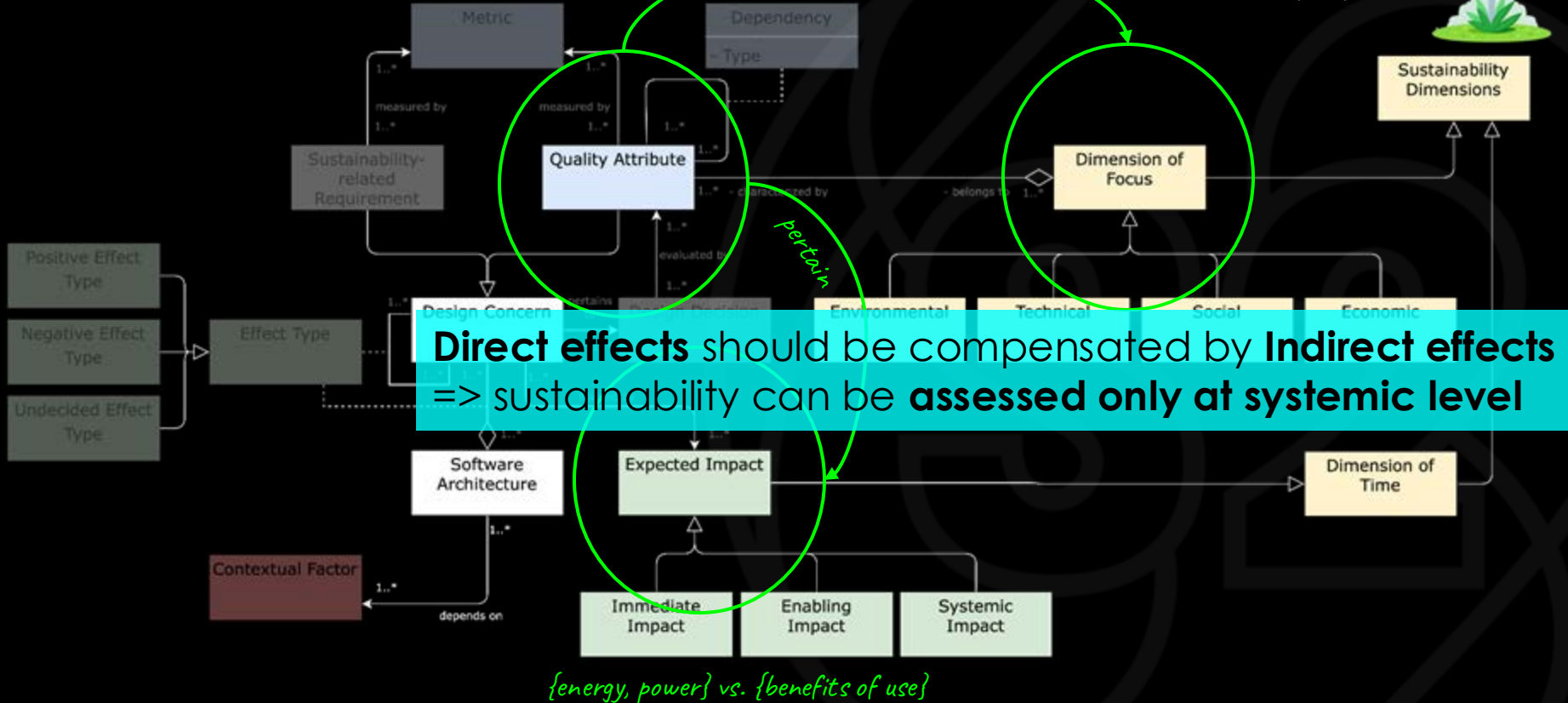


The (desirable) future: where should we go?

Systemic design



The mystery



Direct effects should be compensated by **Indirect effects**
=> sustainability can be **assessed only at systemic level**

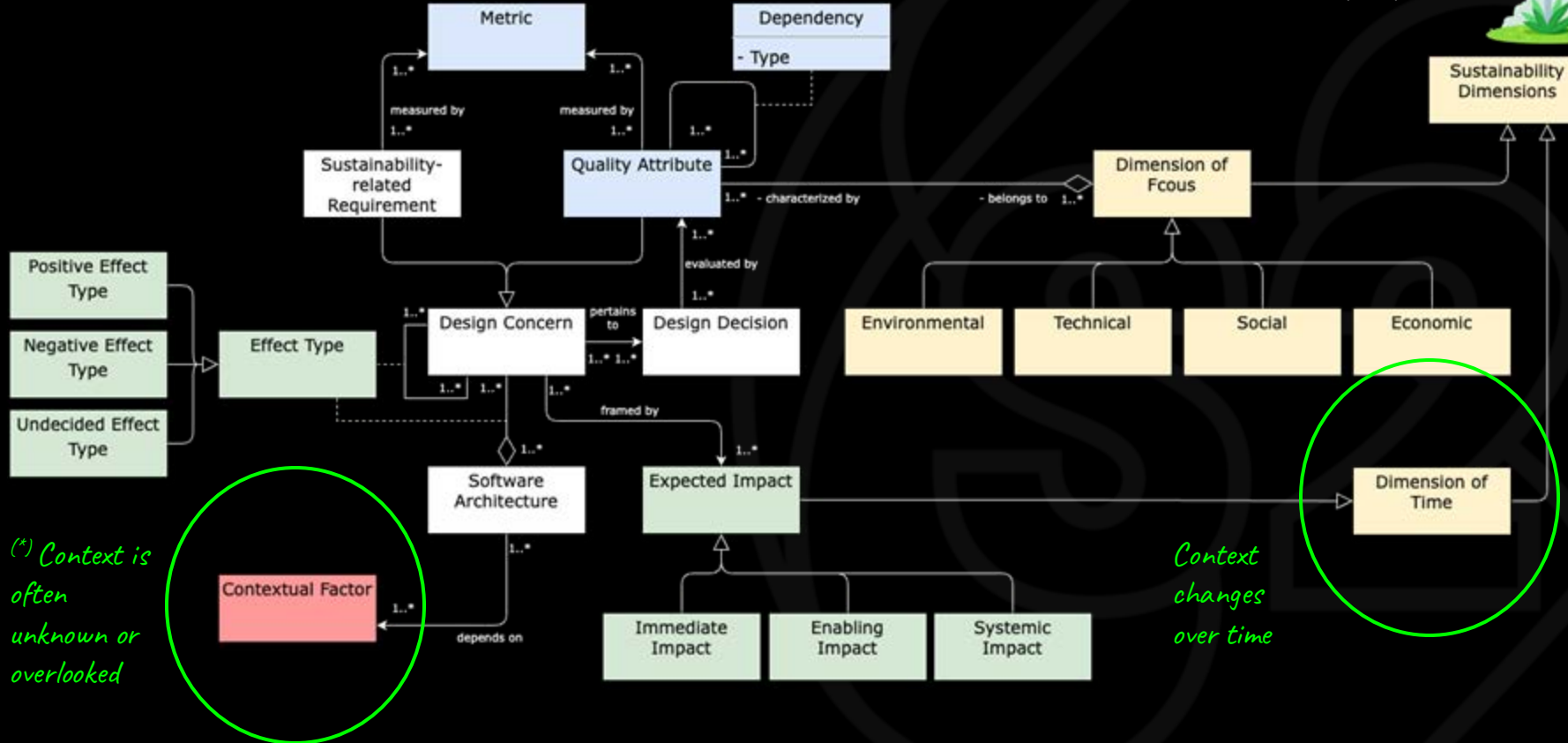
{energy, power} vs. {benefits of use}

The (desirable) future: where should we go?

Role of context



The mystery

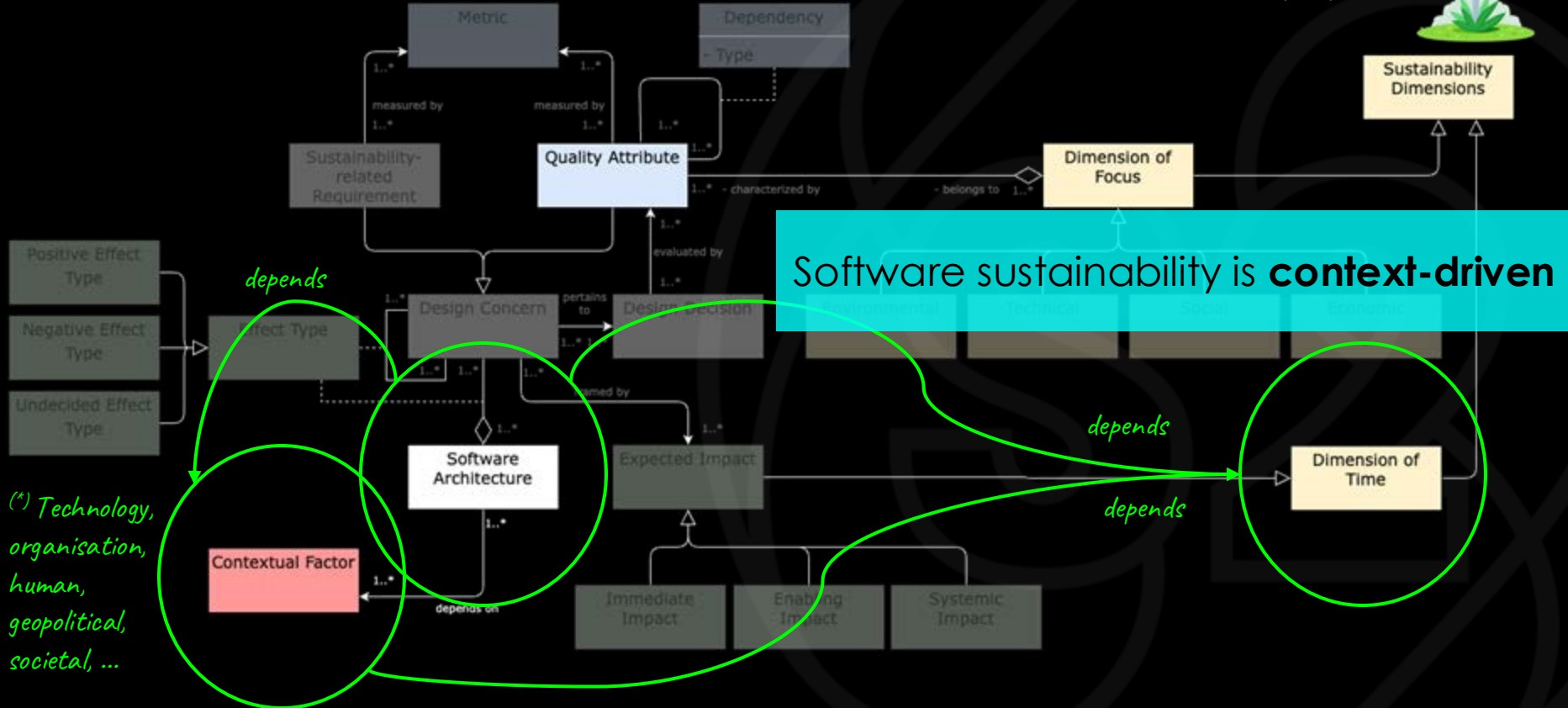
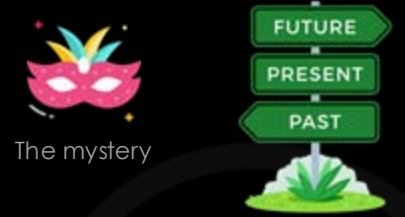


(*) Context is often unknown or overlooked

Context changes over time

The (desirable) future: where should we go?

Role of context

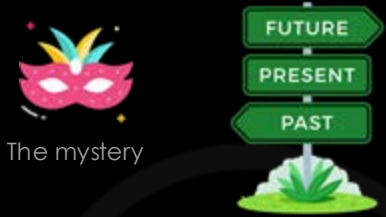


(*) Technology, organisation, human, geopolitical, societal, ...

(*) L. Briand, D. Bianculli, S. Nejati, F. Pastore, M. Sabetzadeh, "The Case for Context-Driven Software Engineering Research: Generalizability Is Overrated," IEEE Softw., vol. 34(5), 2017.

The (desirable) future: where should we go?

Time and Context



The Digital Society Is Already Here – Pity It Is ‘Unsustainable’

Patricia Lago*

Today, digitalisation and society are intertwined to an extent that they have become wholly inseparable. The point where large parts of society would be function if digital systems were to fail. Think of envisioning, traffic control, precision agriculture, or that citizens experience first-hand like access to health records, banking, retail shopping; all these are heavily based on digital solutions and would not be any more without them. The good news, however, thanks to digitalisation, such services are accessed easily and more affordably, hence potentially serving a chunk of the global population.

At the same time, digital systems operate globally involving multiple parties often crossing international and conflicting stakes, a fact that makes them challenging to manage. The global trend of extreme automation (via AI, robotics, and self-driving vehicles) both provides a solution to such challenges (making digitalisation even more autonomous), and further issues with regard to how uncontrollable

* Patricia Lago is a professor in software engineering. She focuses on the intersection between software and sustainability.

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 43, NO. 3, MARCH 2018

Engineering Within Boundaries When Software Has None

Bashar Nuseibeh*, Member, IEEE

Abstract—I served as Editor-in-Chief of IEEE Transactions on Software Engineering for four years between 2013 and 2015. Then and now I continued to be a fan of software engineering to incorporate a range of multi-disciplinary inputs and to address human and social concerns. In this retrospective editorial, I reflect on the remarkable progress that has been made in the intervening years and make the case for an even more radical softening of the software engineering discipline. Such a softening will require new cross-disciplinary methodologies for research and practice, and new organizations of the field: experiences and values of software engineers I suggest two conceptual frameworks for what I advocate: Living Labs L&L and Context L&L. Your reader, please forgive me for indulging.

Index Terms—Software engineering, requirements engineering, architectural systems, reconfigurability, context, values, software without boundaries, human-computer interaction.

1. INTRODUCTION

THE theme of the 2007 International Conference on Software Engineering (ICSE) was “Software Everywhere”. Since then, it has become quite a cliché to use the ubiquity of software in society to motivate the importance of software engineering. When I became Editor-in-Chief (EIC) of IEEE Transactions on Software Engineering (TSE) in 2010, I argued in my first and subsequent editorials [1] for broadening the scope of TSE to encompass software engineering’s human and social concerns. My goal was to attract a wider range of submissions to the journal that reflected more inter-disciplinary work, which in turn reflected the rich and varied multi-disciplinary research on modern software engineering research and practice. My 4-year tenure as EIC was too short to witness a significant change in the journal’s profile, but reflecting on the 15 years since, there is now widespread acceptance in the community that effective “software engineering in society” is a legitimate and key research concern. Software engineering journals and conferences now routinely publish papers that reflect the broader socio-technical context of software development and deployment. There are now many examples of research that considers human and social attributes and behaviours of software developers and teams, much of which is reflected by the social

Manuscript received 10 February 2018; revised 2 February 2018; final version of manuscript received 15 March 2018; accepted for publication 15 March 2018; date of publication 15 March 2018; this work was supported by EPSRC grant EP/M013166/1.

* Bashar Nuseibeh, Member, IEEE, is with the School of Computing and Communications, The Open University, Milton Keynes, U.K. (e-mail: bashar.nuseibeh@open.ac.uk).

Digital Object Identifier 10.1109/TSE.2018.2841200

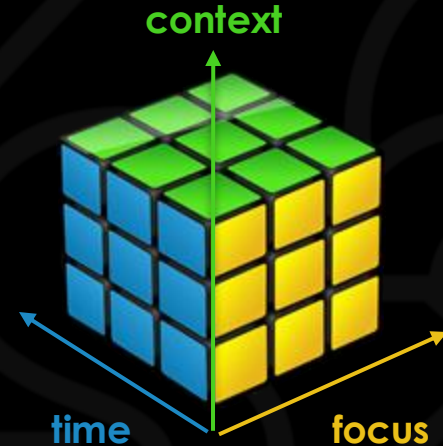
sciences, is empirically well founded, and is shaped by observable engineering practices in the wild.

Problem solved? Not quite. The last 15 years have also witnessed more significant changes, one evolutionary and the other revolutionary. They are closely related:

- 1) **Evolutionary Changes:** There has been an increase in embedding of software in society in ways that dramatically affect human behaviour and the physical environment. A growing body of empirical work has demonstrated that software, and its underpinning engineering practices, significantly affect (in both positive and negative ways) the way we live our lives. This has motivated many in the software engineering community to consider human and societal values that underlie software systems, and to explore how to make such values explicit in engineering processes [2]. For example, software engineering for sustainability [3] has emerged as a research area that is vibrant and value-driven. Other examples include software engineering for health and social care, community policing, and citizen participation and engagement.
- 2) **Revolutionary Changes:** There have been remarkable advances in the fields of artificial intelligence (AI) and machine learning (ML). These have enabled novel work in the traditional research paradigms for the community and opened new research horizons for the community.

While making values explicit in the engineering of an artefact is a key step towards the engineering of an artefact that is socially and ethically responsible, it is not enough. Instead, it is the engineering of an artefact that is socially and ethically responsible that is the key challenge [4]. Indeed, much of the work has been on engineering AI for more effective software engineering practices, but without radically changing the essence of these practices.

The software engineering community has clearly recognized the changing nature of the discipline, but it drives by the disruption and reorganisation of industrial scale organisations that are afforded by AI [5] (or by the deeper engagement of software engineering with societal concerns [5]). However, the community, in my view, is less equipped to society’s changing views and attitudes to software itself, less able to engage with software, or

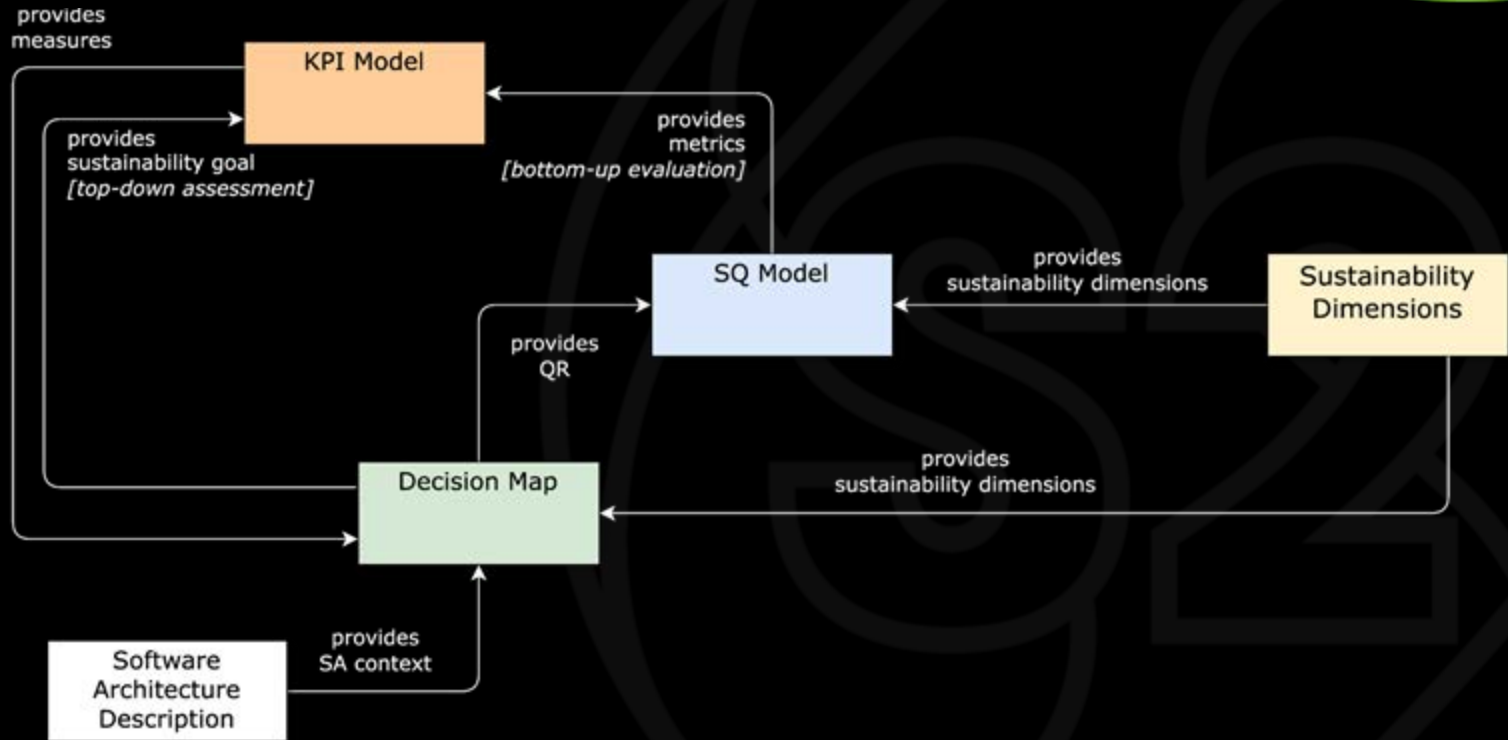
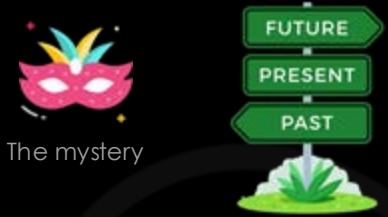


There is no sustainability without **time** and **context**

“The responsible *software engineer* of the future must be the requirements engineer and the software architect of *the lived experience*.” (Bashar Nuseibeh)

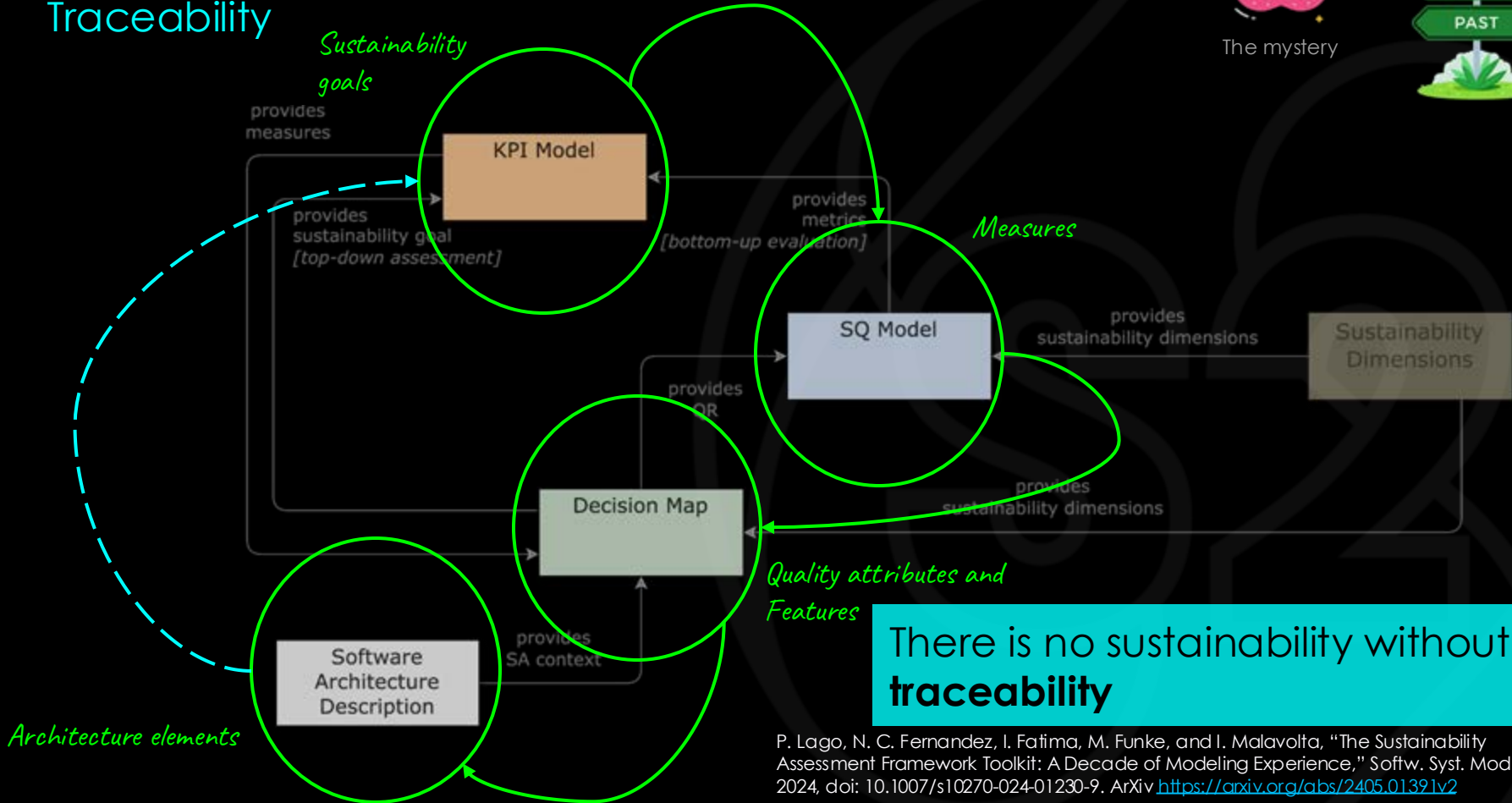
The (desirable) future: where should we go?

Traceability

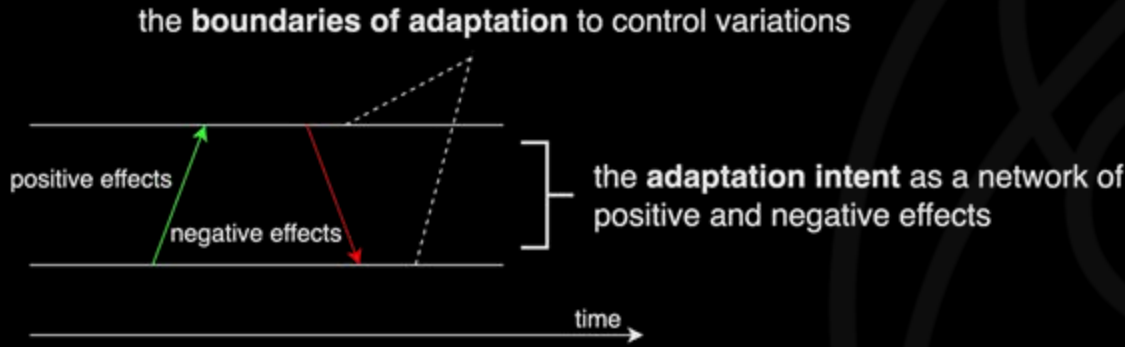


The (desirable) future: where should we go?

Traceability



The (desirable) future: where should we go? “Balance”



There is no sustainability without **balance**

2022 IEEE/ACM 44th International Conference on Software Engineering: New Ideas and Emerging Trends

Expressing the Adaptation Intent as a Sustainability Goal

Iliss Gerostathopoulos
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
i.g.gerostathopoulos@vu.nl

Claudia Rauboldt
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
c.rauboldt@vu.nl

Patricia Lago
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
p.lago@vu.nl

ABSTRACT
Adaptation and sustainability are two key challenges leading the development of software systems nowadays. Adaptation denotes the capacity of a system to cope with variations and uncertainties at runtime in order to continue providing its functionalities with certain quality levels, notwithstanding changes. But how can adaptation and its intent be expressed at design time so that to analyze its possible impact at runtime over a long period of time? To answer this question we look at adaptation from the sustainability point of view. Sustainability denotes the capacity of a system to both maintain and preserve its function over time. We propose an approach which uses decision maps to make sustainability-driven decisions for adaptation in a systematic way. The proposed approach is illustrated through two self-adaptive examples as illustrative cases.

KEYWORDS
Self-adaptive systems, adaptation intent, sustainability goal.

ACM Reference Format:
Iliss Gerostathopoulos, Claudia Rauboldt, and Patricia Lago. 2022. Expressing the Adaptation Intent as a Sustainability Goal. In *New Ideas and Emerging Trends* (IASEE '22), May 13–19, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3504933.3505178>.

1 INTRODUCTION

With the current pace of digitalization, the role of software-intensive systems is becoming increasingly profound. Modern software-intensive systems such as robotic assistants, self-driving cars, and industrial automation systems need to operate in a variety of operational contexts and are expected to be able to deal with emerging uncertainties in these contexts [16]. For instance, they are expected to continue operating when sensors fail, faults occur, and their usage increases rapidly. Even more, they are expected to balance off properties such as operation cost and user-perceived performance to optimize themselves.

To deal with runtime uncertainties, such software systems are typically designed as self-adaptive, i.e., systems that are able to modify their structure or behavior of runtime to respond to external or internal stimuli [10]. For instance, a system with autonomic capabilities is able to change its deployment architecture by adding or removing resources (typically servers) in response to changes in its usage patterns (e.g., increase in the number of user requests).

Interestingly, both the concepts of sustainability goal and sustainability relevant quality concerns (in the technical, economic, environmental, and social dimensions) [1] of sustainability can be used in framing the maintenance and long-term success of an adaptation solution. This allows us to use methods and models originally

Figure 1: A self-adaptive system view (adapted from [22]).

A canonical view of a self-adaptive system distinguishes between a managed system (the part of the system that can be changed at runtime) and a managing system (the part of the system responsible for performing the runtime changes to the managed part) [10]. As depicted in Figure 1, the managing system needs to continuously monitor both the managed system and its environment (e.g., number of users) to decide on the actions to undertake.

In practice, the functionality of continuously monitoring a system and its environment, reasoning over the monitored data, deciding on an adaptation action (e.g., deploying a new server), and performing it at runtime is often hard to design and develop, and even harder to analyze and test. To make matters worse, a managing system not properly itself can result in oscillations in performance, unstable user experience, extra operational costs or even failures—on top of the cost of the extra complexity added to the system.

For self-adaptive systems to be truly successful they need to accommodate changes with a certain degree of autonomy. This way, they are expected to not incur potentially unnecessary changes while preserving the original intent of the managed system over time [5]. In other words, the adaptation intent should correspond to the one of the managed system. To this end, there is a need to (i) define the intent behind the adaptive functionality, (ii) develop the functionality to meet its intent, and (iii) assess the impact of the functionality based on the level its intent is actually met over time.

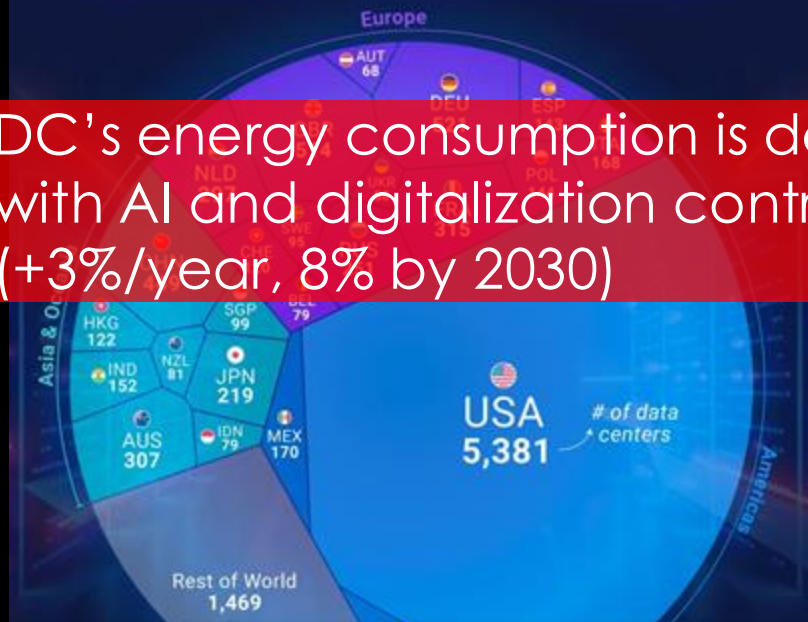
The novel idea we put forward in this paper is to express the adaptation intent of a software system as a sustainability goal it needs to satisfy, i.e., the network of quality concerns balanced over time [13]. So far, sustainability goals have been used in software projects to describe qualities that need to be measured, maintained, or kept balanced to make a project sustainable or assure that a project contributes to sustainability.

Introducing both the concepts of sustainability goal and sustainability relevant quality concerns (in the technical, economic, environmental, and social dimensions) [1] of sustainability can be used in framing the maintenance and long-term success of an adaptation solution. This allows us to use methods and models originally

This work is licensed under a Creative Commons Attribution International 4.0 License.
© 2022 IEEE, May 13–19, 2022, Pittsburgh, PA, USA
© 2022 Copyright held by the author(s).
ACM ISBN 978-1-60959-222-0/22/0000
<https://doi.org/10.1145/3504933.3505178>

36

THE WORLD HAS 11,800 DATA CENTERS



Data Centers and Their Increasing Energy Appetite

Estimated electricity consumption of data centers* compared to selected countries



Clouds Over the Netherlands:
Preserving Public Interest
Internet Governance in the
Era of Hyperscaler Clouds

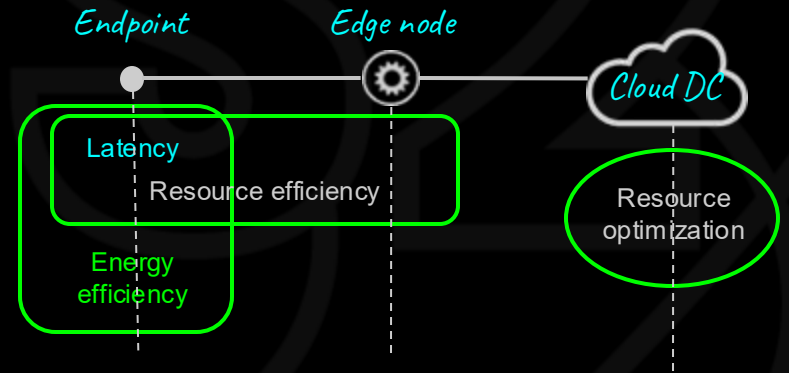
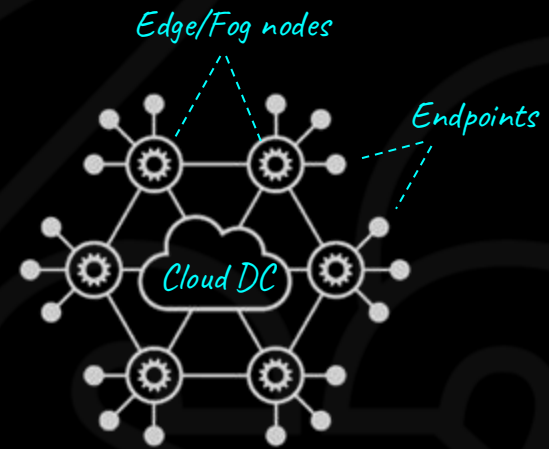
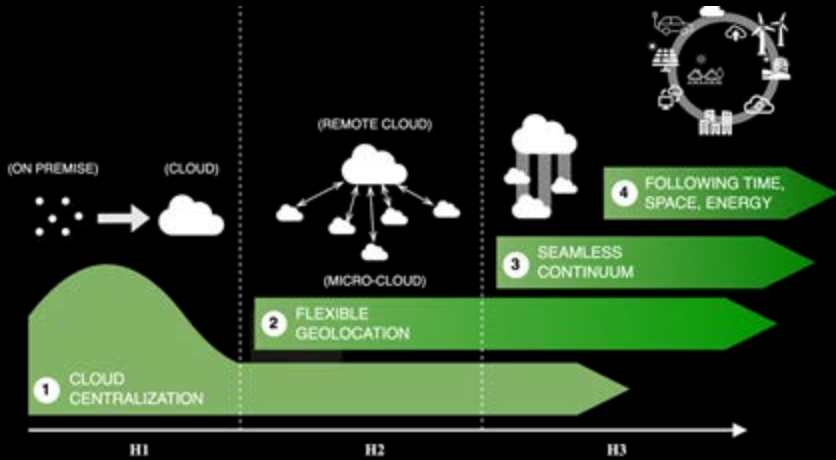
Corinne Cath

DC's energy consumption is doubling every 4 years with AI and digitalization contributing to this growth (+3%/year, 8% by 2030)

Should we invest in a **sufficient cloud**? How do **sufficient SAs** look like?

The (desirable) future: where should we go?

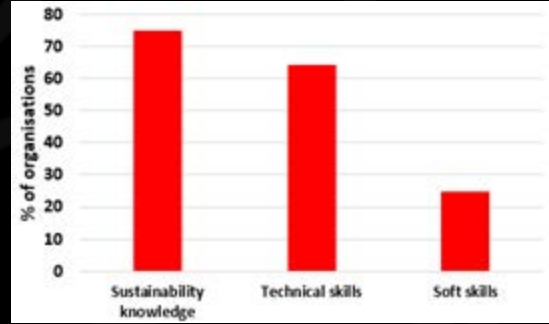
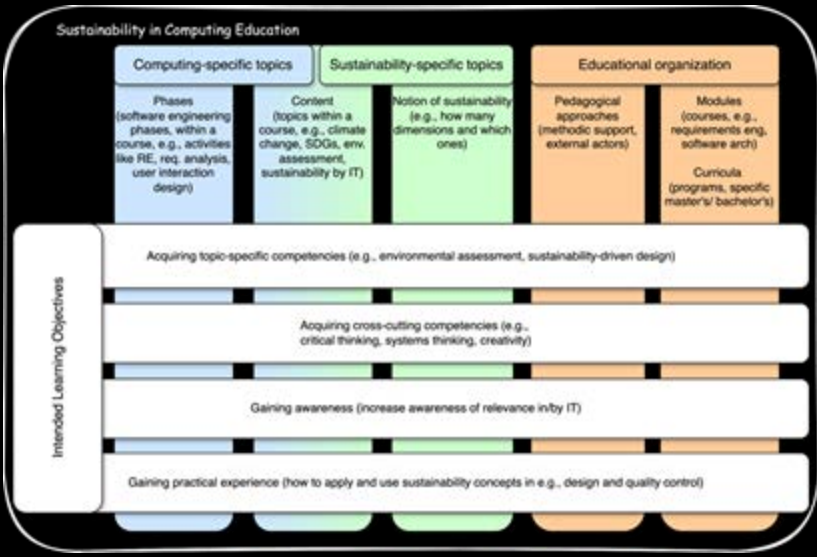
The end of cloud as we know it?



There is no sustainability without **cloud-flexible reference architectures**

The (desirable) future: where should we go?

Education and Training



There is no sustainability without **sound skills and competencies**

academia

ACM Transactions on Computing Education > Vol. 24, No. 1 > Sustainability in Computing Education: A Systematic Literature Review

RESEARCH ARTICLE | OPEN ACCESS | ©

Sustainability in Computing Education: A Systematic Literature Review

Authors: Anne-Katrin Peters, Rafael Capilla, Vlad Constantin Coroama, Rogardt Heidal, Patricia Lago, Ola Leifler, Ana Moreira, João Paulo Fernandes, Birgit Penzenstadler, Jari Porras, Colin C. Venters

ACM Transactions on Computing Education, Volume 24, Issue 1 • Article No. 13, Pages 1 - 53
<https://doi.org/10.1145/3633060>

The Journal of Systems and Software 211 (2024) 111976

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Sustainability competencies and skills in software engineering: An industry perspective^a

Rogardt Heidal^{a,*}, Ngoc-Thanh Nguyen^a, Ana Moreira^a, Patricia Lago^a, Leticia Duboc^a, Stefanie Betz^a, Vlad C. Coroama^{a,b}, Birgit Penzenstadler^{a,c}, Jari Porras^{d,e}, Rafael Capilla^f, Ian Brooks^g, Shola Oyedele^h, Colin C. Venters^{i,m}

In summary...



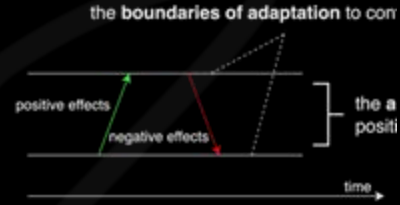
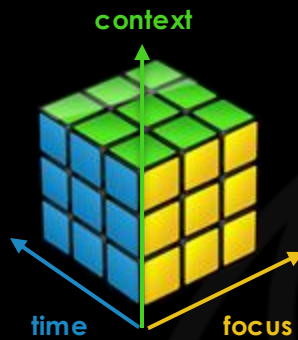
The good



The bad



The mystery



Concepts

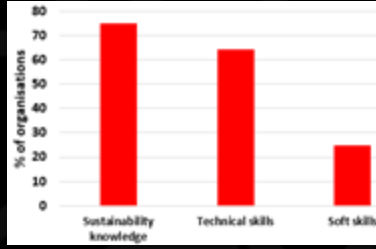
DIMENSIONS OF FOCUS	DIMENSIONS OF TIME HUMAN FACTORS	ROLE OF CONTEXT
---------------------	--	------------------------

There is no sustainability without **time** and **context**

There is no sustainability without **balance**

sw/SE

TECHNOLOGY	KNOWLEDGE SYNTHESIS MEASURES	EDUCATION AND TRAINING
------------	---	-------------------------------



There is no sustainability without sound **skills** and **competencies**

SA

DESIGN DECISION MAKING	QUALITY ASSESSMENT	SYSTEMIC DESIGN TRACEABILITY
------------------------	---------------------------	---

There is no sustainability without **traceability**

There is no sustainability without **cloud-flexible reference architectures**

Thinking of the future...

Def: **Propositions** are statements that accompany PhD dissertations in (some) Dutch universities



Reflections I care to share

1. Multidisciplinary research is hard, interdisciplinary research is *harder*. Sustainability research needs both.
2. Most research questions in software sustainability are old problems that were never solved.
3. Humans drive software engineering: neglecting the first impacts the relevance and quality of the second.
4. We are done with SLRs; what happened to study extensions?
5. "You are what you read." [P. Kruchten, private conversation]
6. "Knowledge has legs and walks home every day" [adapted from Rus and Lindvall (2002), Knowledge management in software engineering. IEEE Software, vol. 19(3)].
7. Think: isn't it ironic to use genAI for presenting a sustainability-related research?
8. Thinking time over publication frenzy.
9. Humans change per extremes: from too little societal impact to too much hype.
10. "Volere è potere" aka "Where there's a will, there's a way." [A. Einstein]

Thank you

Credits: slides, ideas and results are a collective effort with my bright and energetic colleagues in the S2 Group
@Vrije Universiteit Amsterdam s2group.cs.vu.nl

Icons: flaticon.com
Images: dreamstime



lagopatr

www.patricialago.nl



SOFTWARE AND
SUSTAINABILITY
S2 RESEARCH GROUP
VRIJE UNIVERSITEIT AMSTERDAM