Sustainability is Stratified – Toward a Better Theory of Sustainable Software Engineering

Naomod Weekly Seminar, 14/01/2024

Erwan Bousse

Nantes Université

About the paper

Authors and venue

- Paper written by researchers from Dalhousie University (Canada):
  - Sean McGuire
  - Erin Schultz
  - Bimpe Ayoola
  - Paul Ralph
- Presented at the ICSE 2023 Technical Track

Definitions

> « *Sustainable Software Engineering (SSE), means creating software in a way that meets present needs without undermining our collective capacity to meet our future needs.*

> « *[...] most Software Engineering (SE) researchers consider sustainability in terms of resource consumption and waste reduction [4], Becker et al. argue that software sustainability [...] encompasses five interdependent dimensions: environmental, social, economic, individual and technical [5].*

Findings of previous systematic reviews (selection)

- SE has less sustainability research than many similar fields [8].
- Most research on sustainable software focuses on energy efficiency [10], [12], [14]–[16] or technical sustainability [17].
- Most research focuses on the sustainability of software products, rather than process [11], [14].
- Tool support for sustainability concerns is lacking [11].
- Sustainability is complicated and difficult to measure [18]; metrics for ecological sustainability in software have been neglected [19].
- Research into the sustainability of software ecosystems remains in its infancy [21].

Goal and contributions

> « *The five-pillar model has become a notable tool for understanding the impacts of SE [4]. However, many scholarly articles on SSE have been published since.*

Research Question:

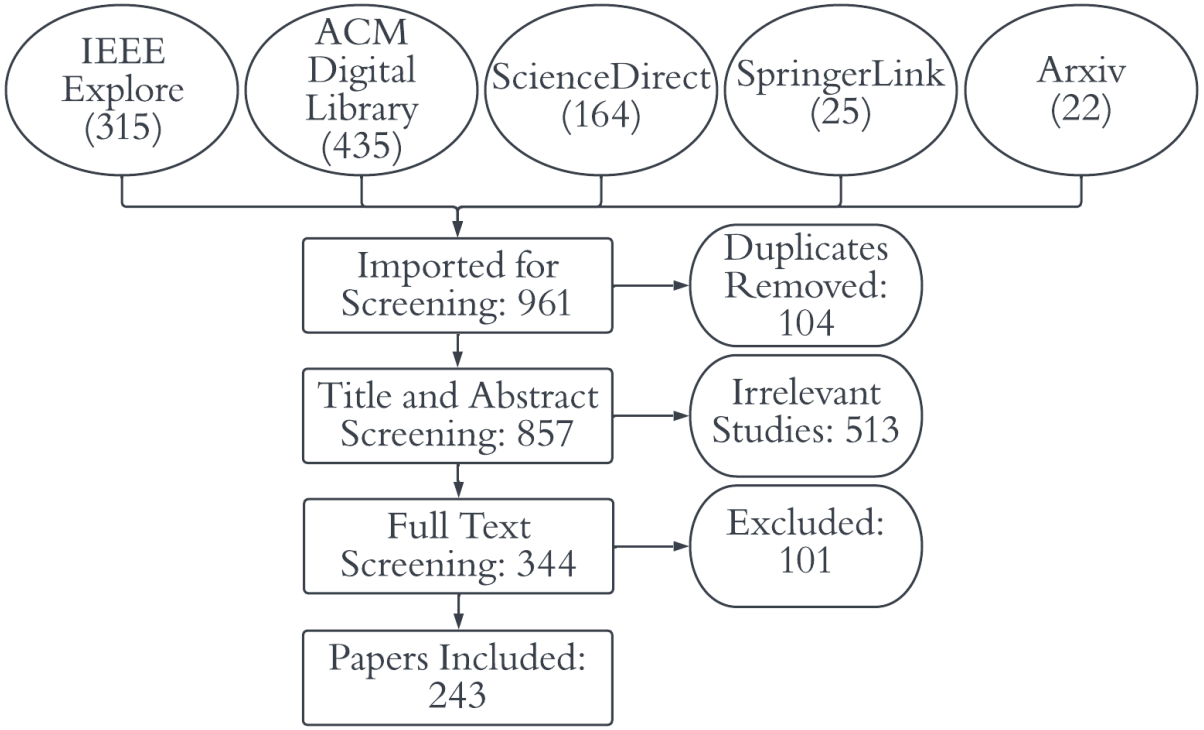
What is the current state of research on sustainability in software engineering?

Contributions:

- A scoping review of the literature (ie. a systematic review) on SE sustainability,
- A qualitative meta-synthesis,
- A novel theory of software sustainability.

Method

- Systematic review scoped to "*results related to sustainability or greenness in software or software development.*"
- Use of the Covidence commercial tool for systematic reviews.



Selection of direct results

Lack of Empirical Methods

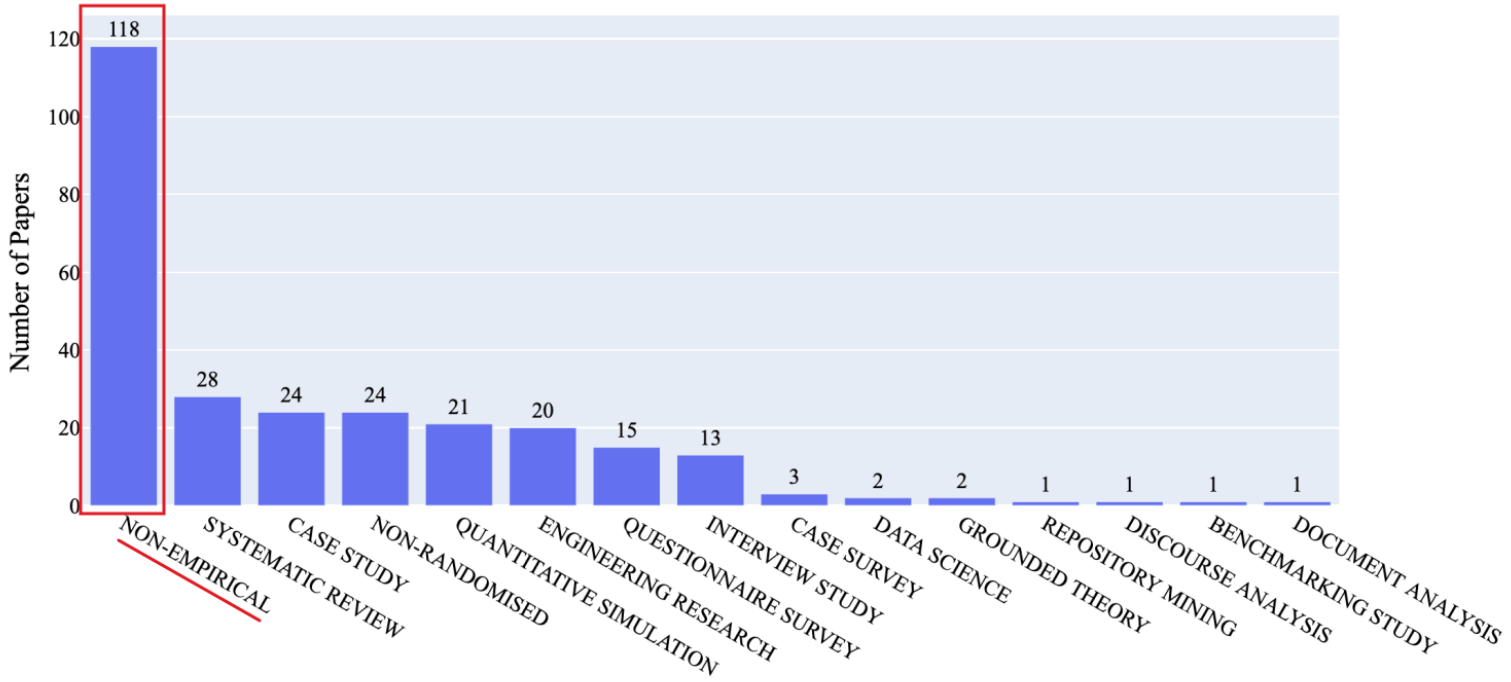


Fig. 3. Research Approaches used by the primary studies

> « *[...] our analysis discovered zero controlled experiments; indeed, the dominant research method was non- empirical (e.g. position papers). While non-empirical scholarship can make important contributions, the ratio of essays to empirical studies [...] is concerning.*

Focus on Products instead of Processes

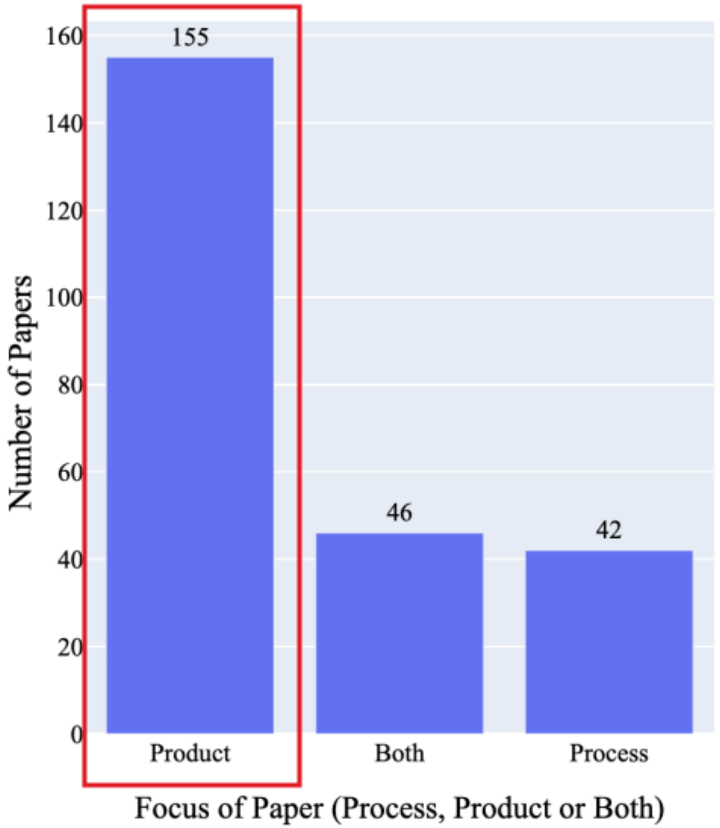
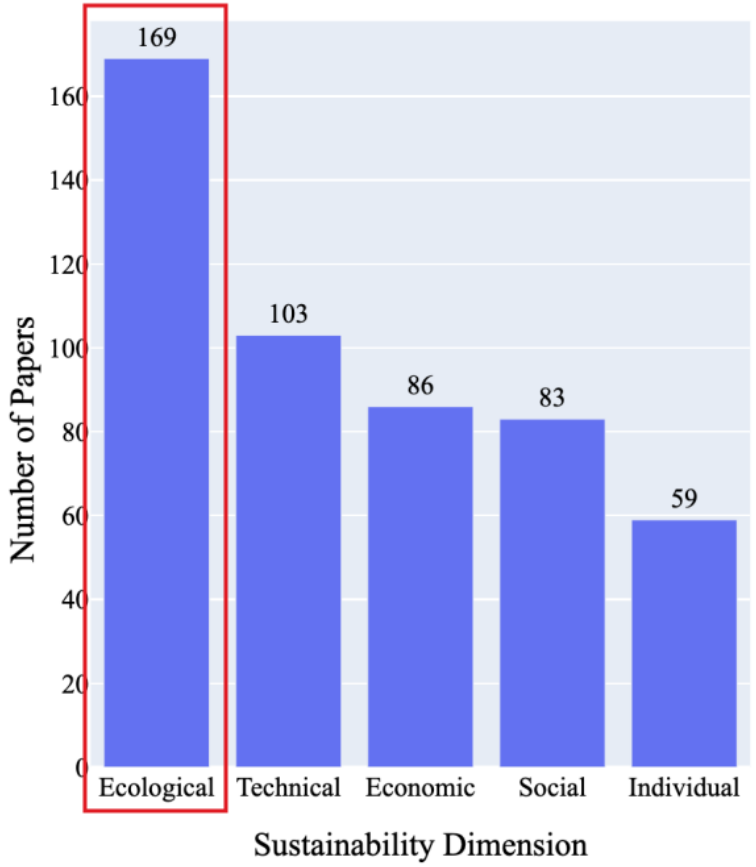


Fig. 4. Count of papers considering each focus category

> « *Process refers to the activities that make up software development, while product refers to a software artifact that is created by a software development process.*

> « *155 papers (64%) addressed only the sustainability of software products*

Focus on the Ecological dimension



> « *The most commonly considered dimension is ecological sustainability, which is addressed by 169 (70%) of the 243 papers, followed by technical (42%),*

Selection of authors' key findings/claims

Sustainability is Stratified

> « *Sustainability has different meanings at different strata, and a project that appears sustainable at one stratum may seem unsustainable at another.*

Example

Economic sustainability of *a company that generates novels using a large language model* and charges readers a flat monthly fee to read as much as they want

- organization layer: product's capacity to produce enough revenue to pay off its cost
- individual layer: extent to which consumers can afford the subscription fee
- industry layer: product's effects on the profitability of conventional publishing
- national layer: product's effects on a country's economy

Sustainability is Multisystemic

> « *A phenomenon is multisystemic when it emerges from the interactions among several systems.*

Example

The sustainability of an *investing app* emerges from the intersection of several systems:

- user's neurophysiological system: the app may present information in a manner that is inaccessible or confusing, leading to bad investment decisions
- national banking system: apps may be subject to government regulations
- collective technical infrastructure: if developed using unstable technologies in an unstable software ecosystem, our retirement savings may be at risk.
- investment vehicles: investing in hydro- ponics companies and solar panel manufacturers is more sustainable than investing in oil companies and bitcoin

Process Sustainability differs from Product Sustainability

> « *SE impacts sustainability through both the process of development and the software products generated*

Examples of improving a product

- reduced energy consumption
- software quality
- pro-social impact
- technical durability

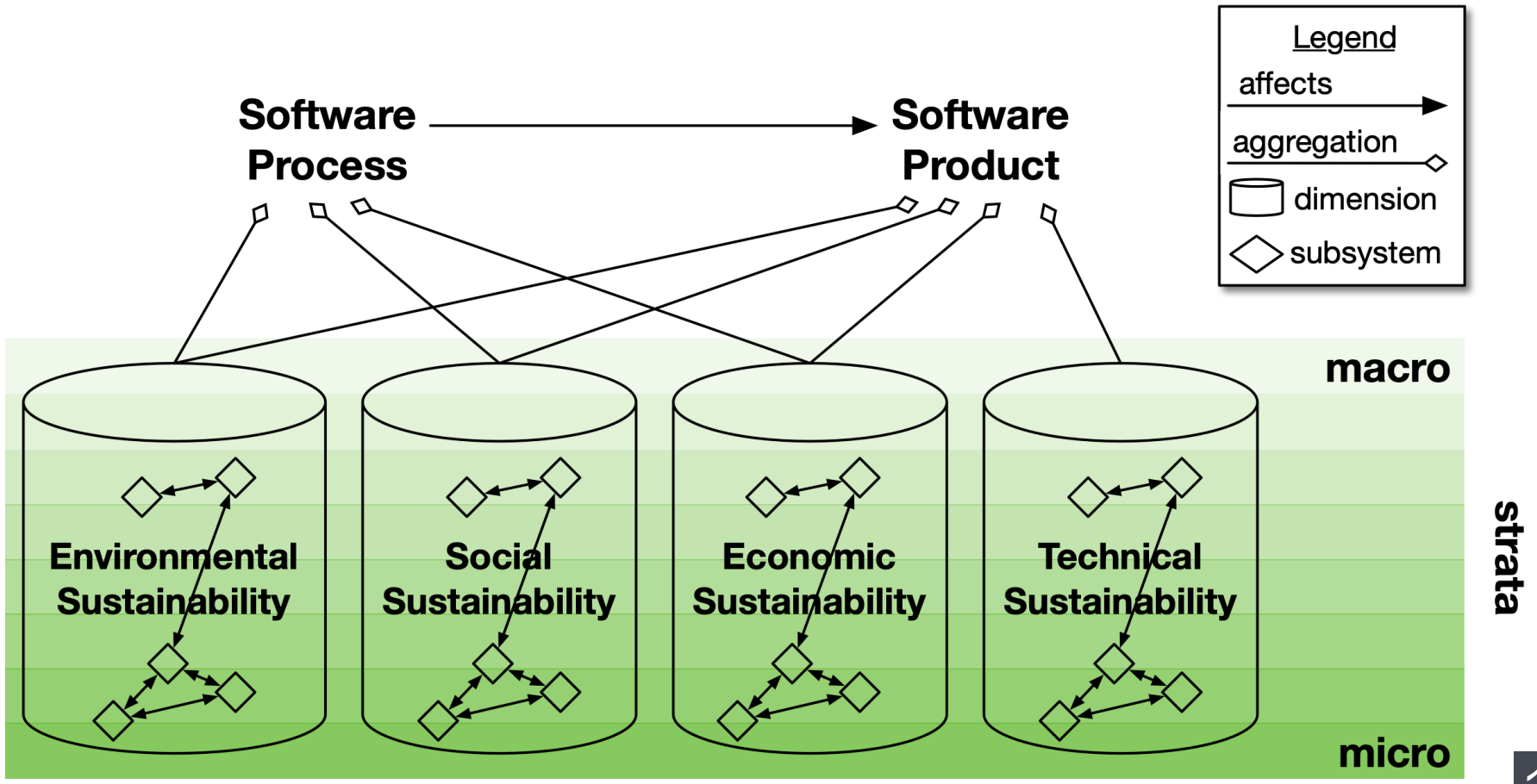
Examples of improving a process

- mitigating disruption during development
- motivating development communities
- sustaining business relationships

> « *the process of developing some software systems can have enormous sustainability challenges. For example, training large language models is highly carbon-intensive [35] while video game development is plagued by socially unsustainable cycles of crunch and burnout [36].*

New theory and conclusion

Stratified Theory of Sustainable Software Engineering



Questions implied by the model

- Are we assessing the software product or its development process?
- Which dimension(s) of sustainability are we assessing — environmental, social, economic or technical?
- For each dimension we are assessing, what strata are relevant? What does sustainability mean at each stratum?
- What are the pertinent subsystems comprising each dimension we are assessing? What does sustainability mean from the perspective of each subsystem?
- How does the software product or process affect each identified subsystem?

Some discussion points

- Benefits of multisystemic view: For some software products (e.g. a web-based video game), the environmental dimension may reduce to carbon footprint. But a software system controlling efforts to engineer the climate can have much broader effects on the world environment.
- Benefits of combined dimensional, multisystemic and stratified views: easier to visualize the *stakeholders* within those systems and emphasize with their respective roles in sustainability efforts.
- Not enough empirical work: means that all sustainable development practices should be viewed with suspicion
- Discourse centered on products rather than development processes, while processes may be the main source of problems (eg. toxic development cultures, training the GPT-3 language model, etc.).

Proposals of (empirical) research ideas

- Do more demographically diverse teams produce more socially sustainable software products?
- Does adding energy consumption tests to test-driven development lead to more sustainable software design?
- How do we reduce the carbon footprint of hot fixes?
- Does pair programming make software teams more resilient?
- Is object-oriented code with higher cohesion and looser coupling more maintainable?
- Do undergraduate ethics courses reduce the tendency to design anti-social algorithms?
- How do we design software more people will actually want instead of tricking people into buying software they will hate?
- How can video game micro-transaction systems be profitable without promoting gaming compulsion?

Wrap-up

My view on the paper

- First time I spend some time to really understand the academic meaning of the concept of sustainability
  - Hard to apprehend all "dimensions"
  - We clearly mostly focus on ecological sustainability
- I liked the paper, which is well-written paper and is a strong study... but :
  - nothing extremely surprising in the results
  - the proposed "theory" feels not very impressive... but it does help thinking about the issue

Still, seems to be quite some work to do in sustainable software engineering !