What do software practitioners think about sustainability?

Tiffany Babey, Catherine Happer and Tim Storer

Timothy.Storer@glasgow.ac.uk
Software efficiency hasn’t been a dominant concern in the software industry since at least the 1990s.
Modern conceptions of software sustainability focus on maintainability, readability and re-use.
About the Software Sustainability Institute

Cultivating research software to support world-class research

Software is fundamental to research: 7 out of 10 researchers report their work would be impossible without it. From short, throw-together temporary scripts to solving a specific problem, through an abundance of complex spreadsheets analysing collected data, to the hundreds of software engineers and millions of lines of code behind international efforts such as the Large Hadron Collider and the Square Kilometre Array, there are few areas of research where software does not have a fundamental role.

Since 2010, the Software Sustainability Institute has facilitated the advancement of software in research by cultivating better, more sustainable, research software to enable world-class research (“Better software, better research”). In 2018, we were awarded funding from all seven research councils. Our mission is to become the world-leading hub for research software practice.
Wider conceptions of sustainability in software practice as a growing concern in the literature.
Thus, we propose the following initial set of principles and commitments:

**Sustainability is systemic.** Sustainability is never an isolated property. Systems thinking has to be the starting point for the transdisciplinary common ground of sustainability.

**Sustainability has multiple dimensions.** We have to include those dimensions into our analysis if we are to understand the nature of sustainability in any given situation.

**Sustainability transcends multiple disciplines.** Working in sustainability means working with people from across many disciplines, addressing the challenges from multiple perspectives.

Sustainability applies to both a system and its wider contexts. There are at least two spheres to consider in system design: the sustainability of the system itself and how it affects overall sustainability of the wider system of which it will be part of.

System visibility is a necessary precondition and enabler for sustainability design. Strive to make the status of the system and its context visible at different levels of abstraction and perspectives to enable participation and informed responsible choice.

Sustainability requires action on multiple levels. Seek interventions that have the most leverage on a system and consider the opportunity costs: Whenever you are taking action towards sustainability, consider whether this is the most effective way of intervening in comparison to alternative actions (leverage points).

It is possible to meet the needs of future generations without sacrificing the prosperity of the current generation. Innovation in sustainability can play out as decoupling present and future needs. By moving away from the language of conflict and the trade-off mindset, we can identify and enact choices that benefit both present and future.

Sustainability requires long-term thinking. Consider multiple timescales, including longer-term indicators in assessment and decisions.
Dimensions of sustainability in the Karlskrona Manifesto
Methods for incorporating sustainability considerations into software design.
Little evidence as to how practitioners conceive of or are influenced by sustainability considerations.

• What aspects of sustainability are of concern to software practitioners?
• How do software practitioners address sustainability in their practice, if at all?
• What resources do software practitioners rely on when addressing sustainability, if anything?
Survey Design

• Lightweight/exploratory intended to gather initial perceptions and potential recruits for follow on focus groups.
• Opportunistic sampling. ~70 respondents to date (58 analysed here).
• Gathered
  • Demographics
  • Perceptions of sustainability in software practice
  • Experience of incorporating sustainability in software practice.

https://glasgow-research.onlinesurveys.ac.uk/sustainability-in-software-practices-survey
Respondent gender
Age
Residency

- Americas
- United Kingdom
- Other Europe
- Asia
- Prefer not to say
Sector
Organisation size

1-10 11-50 50-250 250+ N/A
Roles (multiple)

- Executive role, e.g. CTO, CEO
- Agile Coach
- Application Developer
- Business Analyst
- Software Tester
- Infrastructure/CI/CD Engineer
- User Experience Designer
- Product Owner
- Growth Engineer
- Scrum Master
- Mobile Developer
- Hardware/electronics engineer
- Data Science
- Education, training or coaching
Which of [the Karlskrona Dimensions of Sustainability] do you consider to be relevant to your software practice?
Explanations of sustainability focus on technical concerns.
10. To what extent do sustainability considerations influence your software practice?

![Bar chart showing responses to the question. The x-axis represents the scale from 1 (Not at all) to 5 (A lot), and the y-axis represents the number of responses. The chart shows a majority of responses in the '5=A lot' category.]
Increasing the lifetime and portability of software.

• “writing software that will last”
• “Longevity of the generated software/hardware.”
• “Reducing tech debt at a minimum”
• “ensuring the software is longlasting. E.g. if it can't run on older devices, then it requires new devices and is wasteful”
Assumed relationship between efficiency and environmental sustainability.

- “Low energy/resource use”
- “Building small and efficient software. And not writing software when it's not needed.”
- “developing with performance in mind, limiting the feature scope”
- “Making simple, efficient, and as energy-efficient as possible”
Can you give an example of how you a sustainability consideration has influenced your practice?
Emphasis on open standards, FOSS

- “open and text based file formats where possible”
- “Our software is open-source, ensuring that others can continue to develop it even if we cannot.”
Re-using hardware/software rather than buying or creating new.

• “Choosing established, supported software frameworks over new and experimental ones.”
• “I try to fix my computer instead of buying a new one”
• “refactoring repeated solutions into reusable components.”
Avoiding frameworks and adopting simpler solutions, custom, fine tuned code.

• “I stopped using frameworks built upon frameworks built upon frameworks ... and am trying to build solutions as low-tech and simple as possible.”

• “Less includes, less media, more handmade code.”
Blame the language

• “In the web domain, the biggest and most obvious hinder to sustainability is JavaScript.”

• “I believe compiled languages are ...usually less resource-consuming than interpreted programs like those written in python and Java.”
Can you give an example of how the main organisation you work for prioritises sustainability?
Q. 13. Can you give an example of how the main organisation you work for prioritises sustainability?

“No idea”
Q. 13. Can you give an example of how the main organisation you work for prioritises sustainability?

• “...does not prioritise sustainability in software.”
• “I don't see it prioritized or seen as central to the service/product development in any way though.”
• “I doubt every sustainability... statement my university makes”
• “Inside of work there is active resistance to anything sustainable.”
• “My employers stated this as an explicit non-goal when meeting deadlines.”
Have you used, or are you aware of support training or forms of support that help you to implement sustainability into your practice?
Have you used, or are you aware of support training or forms of support that help you to implement sustainability into your practice?

Two respondents:
- Using recycled hardware
- Promoting use of FOSS in schools
- Carbon calculators
- Green hosting checkers
- Branch Magazine, Low tech magazine
- Community knowledge sharing
Beyond technical considerations, sustainability is not a core consideration for software practitioners.

“This is very important, but it still somehow feels orthogonal to the day to day - particularly the social implications of decisions”
Software practitioners are pessimistic (and cynical) about the prospects for change.

“developers don't have the agency...Business decisions are driven by profit and growth, which is driven by consumptive culture. Developers can, to some extent, cut around those issues by promoting FOSS, but that has its own sustainability issues.”
A call for more evidence!

“I think we need to be educated using practical examples which can help us see the benefit of ‘sustainable software development’. A comparison between two projects (i.e., with and without sustainability considerations) could help us see the difference perhaps.”
Some reflections
Predominant focus on *technical* sustainability, and to some extent, environmental sustainability.
Assumption that environmental sustainability in computing means efficient compute, but practitioners are unclear how to measure this.

“simple, efficient, and as energy-efficient as possible”
Recognition of the need to extend hardware lifetimes.
Apparent tension between technical and environmental sustainability in software practices and decision making.
How we approach Computing Science Education.

“I think computing as a whole has an irresponsible view that energy and resources are infinite”
Thanks and questions

https://glasgow-research.onlinesurveys.ac.uk/sustainability-in-software-practices-survey

Please share!
Green Software Manifesto