Data-Driven Software Sustainability

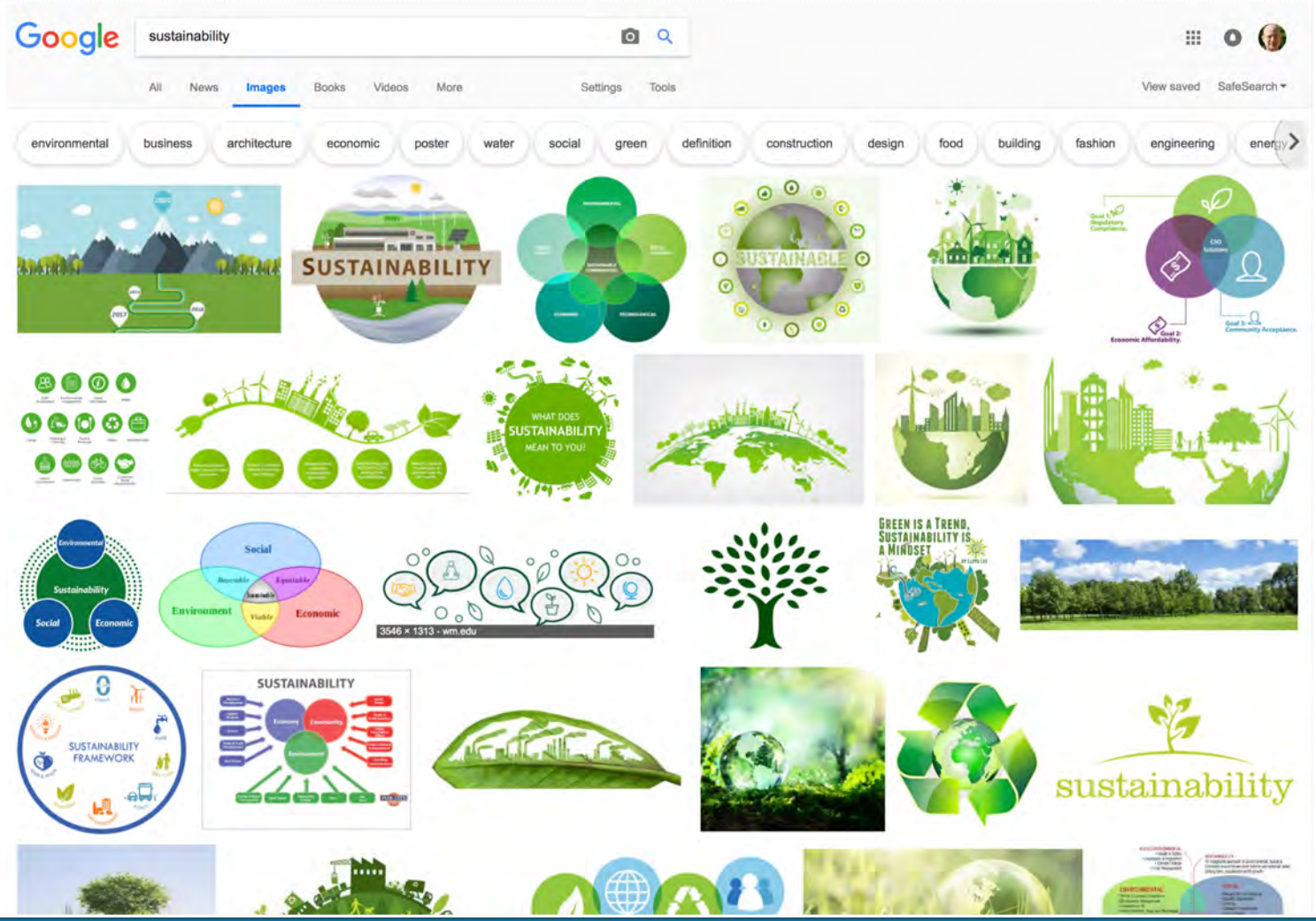
CW3S19, 24 July 2019

Daniel S. Katz
(d.katz@ieee.org, http://danielskatz.org, @danielskatz)

Assistant Director for Scientific Software & Applications
Research Associate Professor, CS, ECE, iSchool

ILLINOIS
NCSA | National Center for Supercomputing Applications

What is sustainability?


Google
sustainability
All
News
Images
Books
Videos
More
Settings
Tools
View saved
SafeSearch
environmental
business
architecture
economic
poster
water
social
green
definition
construction
design
food
building
fashion
engineering
SUSTAINABILITY
SUSTAINABLE
WHAT DOES SUSTAINABILITY MEAN TO YOU?
Environmental
Sustainability
Social
Economic
Social
Environment
Economic
3546 × 1313 - wm.edu
GREEN IS A TREND, SUSTAINABILITY IS A MINDSET
SUSTAINABILITY FRAMEWORK
SUSTAINABILITY
sustainability

What is sustainability?

- Most often used in the context of ecology, often specifically in the relationship between humans and the planet

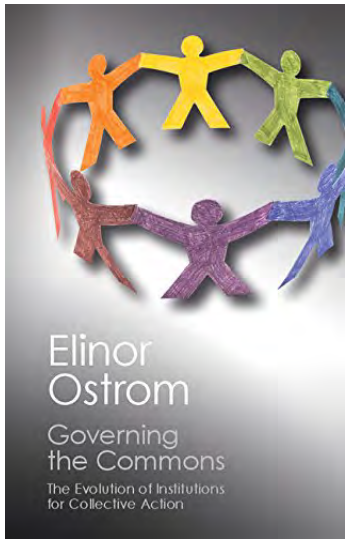
- Example: Karl-Henrik Robèrt (via Wikipedia & paraphrased)
  - Natural processes are cyclical but we process resources linearly
  - We use up resources, resulting in waste
  - Waste doesn’t find its way back into natural cycles; not reused or reassimilated
  - Call for "life-styles and forms of societal organization based on cyclic processes compatible with the Earth's natural cycles"

Two views on software sustainability

1. Software sustainability focuses on the software ecosystem
   - The software ecosystem should be permanently sustained, even though individual packages will die over time

Sustainability in the context of software

- Elinor Ostrom's (Governing the Commons) definition of sustainability for a common-pool resource (CPR): "As long as the average rate of withdrawal does not exceed the average rate of replenishment, a renewable resource is sustained over time."
  - Notion of a cyclic property, though cycle period not specified
  - But rate (sustainability) of what?
- Titus Brown: "the common pool resource in open online projects is effort"
- We need to sustain overall effort by encouraging/rewarding open source activities
- With enough effort, needed software will be sustained



C. T. Brown, "A framework for thinking about Open Source Sustainability?"
http://ivory.idyll.org/blog/2018-oss-framework-cpr.html

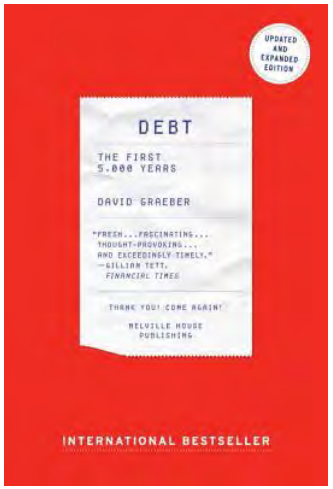
Two views on software sustainability

1. Software sustainability focuses on the software ecosystem
   - The software ecosystem should be permanently sustained, even though individual packages will die over time

2. Software sustainability focuses on individual packages
   - Permanently sustaining software packages is not a goal
   - But some packages need to be kept working
   - Define sustainability as the capacity of the software to endure
     - Will the software will continue to be available in the future, on new platforms, meeting new needs?

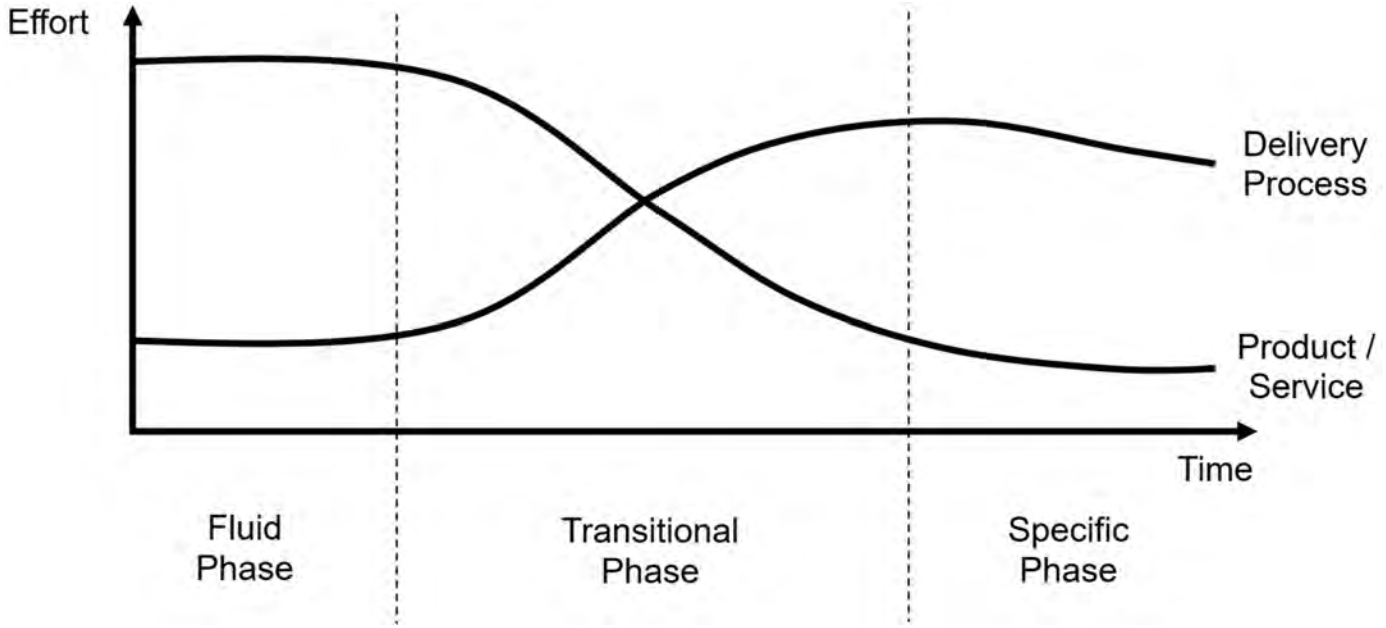
“Equations” of software sustainability

- Software sustainability ≡ sufficient Δ software state
  - Sufficient to deal with: software collapse, bugs, new features needed
- Δ software state = (human effort in – human effort out - friction) * efficiency
  - Software stops being sustained when human effort out > human effort in over some time
- Human effort ⇆ $
  - All human effort works (community open source)
  - All $ (salary) works (commercial software, grant funded projects)
  - Combined is hard, equation is not completely true, humans are not purely rational
- Δ software state $\overset{?}{\rightarrow}$ users choose to volunteer effort or $
  - Development choices might take this into account



Debt: The First 5,000 Years by David Graeber

Needed type of effort changes over time



- For both single package or for ecosystem of packages

Leng J, Shoura M, McLeish TCB, Real AN, Hardey M, McCafferty J, et al. (2019) Securing the future of research computing in the biosciences. PLoS Comput Biol 15(5): e1006958. https://doi.org/10.1371/journal.pcbi.1006958

Who starts new software projects?

- User/Developer
  - To scratch their own itch

- Then options:

1. Keep it for myself
2. Share it
3. Accept contributions (effort), and if so:
   a. Broaden focus?
      - Bring together other (related) packages
   b. Broaden governance?
      - Collaborate with other developers

Why do people lead and contribute to projects?

- Engagement: meaningful and valuable actions that produce a measurable result
- Engagement = Motivation + Support – Friction
  - Intrinsic motivation: self-fulfillment, altruism, satisfaction, accomplishment, pleasure of sharing, curiosity, real contribution to science
  - Extrinsic motivation: job, rewards, recognition, influence, knowledge, relationships, community membership
  - Support: ease, relevance, timeliness, value
  - Friction: technology, time, access, knowledge

Systemic improvements

- Credit for developers and maintainers
  - Software citation
  - FORCE11 Software Citation Implementation Working Group
- Career paths for developers and maintainers
  - Research Software Engineer Association
  - Society of Research Software Engineers (UK)
  - US-RSE Association
- Document best practices (or good enough practices)
  - Underway by lots of communities
    - E.g. The Carpentries, SSI, URSSI, BSSw, ELIXIR, …

Project-specific choices

- Which features should be added next?
- Which PR should be accepted next?

- These decisions partly depend on the developer's needs
- And of course, on impact on current and new users

- Now, add their impact on sustainability as a factor
  - If PR 1 will make the project harder to sustain and PR 2 will make the project easier to sustain, …
  - If adding feature A will bring in new developers, and feature B will not, …
  - If Funder X is interested in …

How can we determine these?

- Measure current community health
  - By CHAOSS metrics or other means
- Estimate community health under various options
- How? Not clear, but some ideas:
  - Look at past projects and their similar decisions; use these to project the possible impact of future decisions in new projects
  - Run role-playing exercises with real developers and real users
  - Perform A/B testing with real projects
  - Gather data from successful and unsuccessful projects; tie anecdotes about these projects to their outcomes
  - Survey leaders of successful projects to understand what choices they would make in a particular situation

Summary

- Define sustainability as
  - Inflow of resources sufficient to do the needed work
  - Those resources can be turned into human effort
- Generic methods to improve sustainability
  - Bring in more resources (funding, people) - incentives
  - Reduce the needed work – best practices
- Project-specific methods to improve sustainability
  - Consider impact of project decisions on sustainability, not just developer needs and current/new user needs
  - Research needed into how to do this

ILLINOIS

NCSA | National Center for
Supercomputing Applications