The Software Gardening Almanack

Dave Bunten[1], Will Davidson[1], Faisal Alquaddoomi[1], Vincent Rubinetti[1], Gregory P. Way[1]

[1]*Department of Biomedical Informatics, University of Colorado Anschutz Medical Campus*

Scan for coSMicQC!

Department of Biomedical Informatics
SCHOOL OF MEDICINE
UNIVERSITY OF COLORADO **ANSCHUTZ MEDICAL CAMPUS**

I. Fragile code & sustainability debt

Scientific software often becomes brittle: documentation gaps, weak tests, and ad-hoc practices combine with time to create **sustainability debt**. This debt hides defects, slows research, and undermines reproducibility. Existing guidance is scattered and qualitative. Teams need **measurable signals** and **actionable checks** that map to better long-term outcomes.

II. Handbook + Python package

We present **The Software Gardening Almanack**, an open-source project with two complementary components:

- **Handbook (Jupyter Book):** teaching the "why" and "how"—concepts, examples, and tutorials.
- **`almanack` Python package:** the "do"—compute metrics (including software information entropy), build reports, and run lint-style **sustainability checks** that you can gate in CI.

Together, they help labs quantify maintenance risk, prioritize fixes, and cultivate code that lasts.

III. Getting started with `almanack`

Installation

```
# install from PyPI
pip install almanack

# or install directly from source
pip install git+https://github.com/software-
gardening/almanack.git
```

Use the CLI

```
# generate a JSON table of sustainability metrics
for a repository
almanack table path/to/repository

# run lint-style sustainability checks (nonzero
exit if failing)
almanack check path/to/repository
```

Pre-commit integration

```
# .pre-commit-config.yaml
- repo: https://github.com/software-gardening/
almanack
  rev: v0.1.1
  hooks:
    - id: almanack-check
```

Python API

```
import almanack
import pandas as pd

# Build a table of repository metrics
table = almanack.table("path/to/repository")
pd.DataFrame(table).head()
```

Compute a quick sustainability score

```
from almanack.metrics.data
import compute_almanack_score

score = compute_almanack_score(table)
print(score)  # {'passed': 18, 'total': 24,
'score': 0.75}
```

List failed checks for fast triage

```
from almanack.metrics.data
import gather_failed_almanack_metric_checks

failed
= gather_failed_almanack_metric_checks("path/to/
repository")
for item in failed:
    print(f"{item['id']}: {item['description']}
(why it matters: {item['rationale']})")
```

IV. What the metrics capture

- **Repository hygiene:** README, license, citation, contribution guide, code of conduct, docs location.
- **Process health:** CI configuration, build status, test presence, coverage parsing (e.g., Python).
- **Community signals:** contributors, tags/releases, issue tracker availability.
- **Evolution/complexity: information entropy** at file and repo levels to highlight hotspots and decay risk.
- **Scoring & gating:** boolean checks mapped to maintenance direction (positive/negative) plus an overall **Almanack score**.

V. Visual analytics & examples

The **Seed Bank** demonstrates how to reproduce and explore measures (e.g., normalized Shannon entropy) across thousands of research repositories, relating **time**, **complexity**, and **engagement**.

VI. Typical workflows in a lab

1. **Survey repositories** to establish a baseline metrics table and identify immediate fixes (missing license, broken badge, etc.).
2. **Add guardrails** with pre-commit/CI so regressions are caught early (e.g., missing citation file).
3. **Plan maintenance** using entropy hotspots and failed-check rationales to guide refactoring and documentation.

VII. Real-world value for scientists

- **Reproducibility & trust:** standard checks and reports make expectations visible for collaborators and reviewers.
- **Onboarding faster:** new team members see structure (docs/tests/CI) and can contribute earlier.
- **Longevity:** entropy and process signals help choose where to invest limited maintenance time.
- **Automation-ready:** outputs integrate cleanly with dashboards, CI logs, and PR comments.

VIII. Acknowledgements

We thank contributors and supporters of the Almanack effort, including the **Better Scientific Software (BSSw) Fellowship Program**, **U.S. Department of Energy (DOE)**, **National Nuclear Security Administration (NNSA)**, and the **National Science Foundation (NSF, award 2327079)**; and collaborators across the University of Colorado Anschutz Medical Campus.

- Project: https://github.com/software-gardening/almanack
- Handbook: https://software-gardening.github.io/almanack
- PyPI package: https://pypi.org/project/almanack/