OSCARS
Open Science Clusters' Action for Research & Society

Anas El Hounsri[1] (0009-0005-6868-4511), **Daniel Garijo**[1] (0000-0003-0454-7145),
Tom François[2] (0000-0001-5226-3089), **Thomas Vuillaume**[2] (0000-0002-5686-2078)
[1] Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
[2] LAPP, CNRS / Université Savoie Mont Blanc, Annecy, France
with Oscar Corcho and Esteban González (UPM)

CodeMetaSoft

Detecting, reporting and fixing research software metadata drift

w3id.org/codemetasoft

Summary

Your `codemeta.json` was right the day you wrote it. Is it still?

Software metadata gets written once and then **left to rot**. We built a checker that compares `codemeta.json` with the rest of the repository, a catalogue of 29 issues, and a bot and CI action that report them.

On 1,500 repositories that already ship the file, most issues are about time: versions, dates and dependencies that were right once and never updated. The gap is maintenance, not adoption.

A typical codemeta.json, one release later

```
{
  "version": "1.1.0",                                  P001
  "license": "GPL",                                    P013
  "identifier": "10.5281/zenodo.123",                  P014
  "dateModified": "2022-03-01",                        W002
  "softwareRequirements": ["numpy"]                    W001
}
```

Illustrative file. The latest release is 1.2.0, the license needs a version, the DOI is not a URL, the file has not changed since 2022, and numpy has no version.

Challenge

- Software metadata (authors, version, license, identifiers, dependencies) makes research software findable, citable and reusable.
- It is spread over many files: `pyproject.toml`, `pom.xml`, `DESCRIPTION`, `CITATION.cff`, `README`, `LICENSE`. CodeMeta gathers it in one `codemeta.json`, written by hand.
- FAIR checkers test whether a field is *present*. None checked whether its value is **correct** or agrees with the rest of the repository.
- Adoption is low too: only **0.13%** of 370,684 OpenAIRE software repositories have a `codemeta.json`.

Solution

Measure Metadata practices in the Science Clusters (MSR 2025), the OpenAIRE Graph and 1,500 CodeMeta repositories.

Diagnose Two authors and two experts inspected 150 random repositories by hand.

Repair One detection rule and one repository-specific fix per catalogued issue.

Monitor A CI check for developers; a bot and a dashboard for registries and labs.

19 pitfalls | **10 warnings**

Pitfalls break schema compliance, licensing or citation pipelines. **Warnings** lower quality without breaking them.

Results

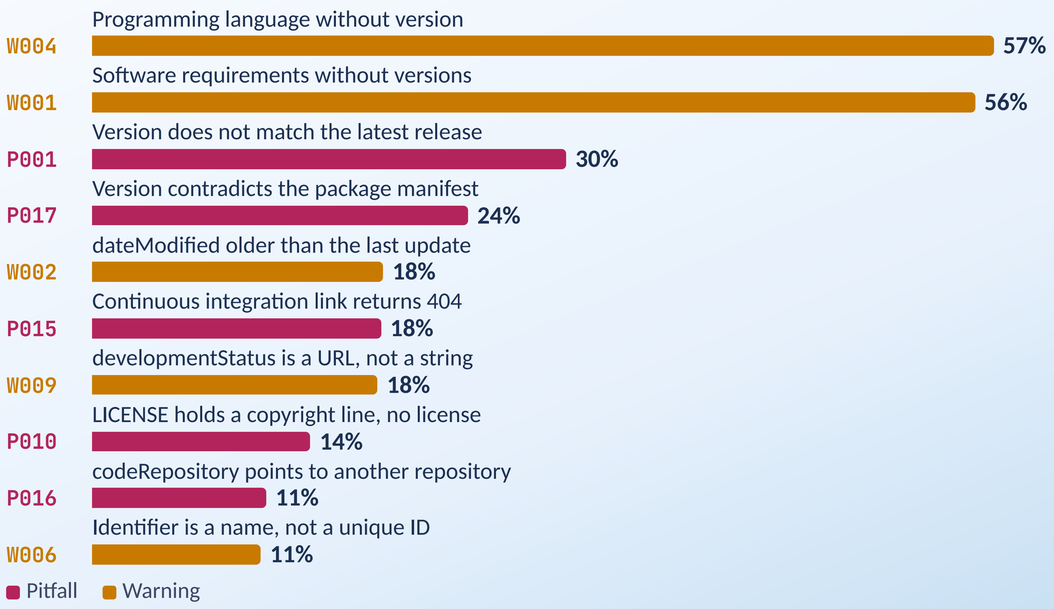
30% version does not match the latest release `P001`

24% version contradicts the package manifest `P017`

56% requirements listed without versions `W001`

18% stale `dateModified` `W002`

Ten most frequent issues in 1,500 repositories with a `codemeta.json` (share affected):



The checker is accurate

Measured against a published hand-annotated ground truth (rsmetacheck 0.3.5, 143 repositories):

0.89 **precision**: 358 of 403 findings are right

0.80 **recall**: 0.88 without W004, a rule we relaxed after maintainer feedback

Curation helps, but not over time

The **ESCAPE OSSR** (30 particle physics and astronomy projects) requires a `codemeta.json` at submission. Even there:

1.0 pitfall per record on average

1.4 warnings per record on average

53% list requirements without versions, 40% have a stale `dateModified`. The onboarding review catches broken links; drift escapes it.

Monitoring runs, behaviour does not change

The bot ran **50 times unattended** on the OEG-UPM organisation (331 → 341 repositories, Jul–Sep 2026). Pitfalls per repository went from 0.060 to 0.070, warnings from 0.193 to 0.202. An issue alone does not change practice.

Exploitation of results

- **ESCAPE OSSR** runs value checks and a public monitoring page on top of its CodeMeta completeness indicator.
- **EVERSE**: reports follow its assessment schema; LAPP's completeness indicator is registered there.
- **FAIR2ADAPT** shows rsmetacheck reports in its software catalogue.
- Nothing is physics-specific: any lab or registry can point the bot at its repositories.

What we learned

Check values, not fields.
Comparing `codemeta.json` with manifests, releases and `CITATION.cff` finds what presence checks miss.

Policy catches structure, not drift.
Versions and dates go stale after the submission review.

Fixes should arrive at release time,
as pull requests, when people care about metadata.

Next: auto-fix pull requests for the easy cases, rsmetacheck at OSSR submission, Software Heritage integration.

Sustainability

All tools are open source and run without us. Every catalogue entry has a persistent identifier, e.g. `w3id.org/rsmetacheck/catalog/#P001`, with a definition, a real example and a fix.

`rsmetacheck` library + CLI · v0.3.6	Extracts metadata with SoMEF, applies the 29 rules, writes JSON-LD reports in the EVERSE schema. 416 unit tests.
`rsmetacheck-action` GitHub Marketplace	Runs on every push or PR and annotates the diff. A GitLab CI job does the same.
`rsmetacheck-bot` v0.6.1	Watches GitHub and GitLab repositories, keeps one issue per repository up to date, respects opt-outs.
`rsmetacheck-dashboard`	Public page with the latest findings and run history.
`Auto CodeMeta` v3.0 · web app	Paste a repository URL to get a pre-filled CodeMeta 2.0/3.0 file.

Funded by the European Union

OSCARS has received funding from the European Commission's Horizon Europe research and innovation programme under grant agreement No. 101129751. CodeMetaSoft ran from 1 Nov 2024 to 31 Oct 2026.

Learn more about the project
w3id.org/codemetasoft

Project | Catalogue | CI action