

Releasing software in CSIRO

J-M

2023-01-25

Table of contents

Introduction	1
Why? What for?	2
Dimensions	2
How	2
Technical aspects (mostly)	2
Distribution channels	3
Code versioning	3
Data and Software as first class publications	3
Stories from the frontline	3
Circa 2010 R and .NET interop	3
Case study: streamflow forecasting	3
Dimensions	4
Distribution channels	4
Overheard recently	4
Discussion	5

Introduction

An opinionated (?) conversation starter for the 2023 Science Reading club

The confluence page [Software Release Process in CSIRO](#). Apparently some Data61 legal and/or commercial minds are currently revising this. This remains the main entry point cited for CSIRO regarding software release, usually in the context of thinking of open source.

Definitions

- Installable software: `myprog_installer_x64.exe`
- Online service, e.g. [australian-water-data-service](#)

Let's approach the topic starting with the “why”, even when many question arise on the “how”.

Why? What for?

Dimensions

- Legal and Contractual
- Commercial
- Freedom to operate
- Craftmanship and learning the SOTA
- Visibility. Personal and corporate.
- Impact
- [Reproducibility](#), Quality. Also [Toward practical transparent verifiable and long-term reproducible research using Guix](#), [Supporting computational reproducibility through code review](#)
- Maintenance
- **Community**. Overlapping the What and the How: [The art of community](#) by Jono Bacon (Ubuntu employee)

[FAIR Principles for Research Software \(FAIR4RS Principles\)](#)

How

Technical aspects (mostly)

- Code Version management. Git pretty much a given these days.
- Documentation
- Dependency management.
- Unit testing, code coverage
- Reference doc (docstrings); Architecture.
- Tutorials, vignettes, sample workflows
- Sample data
- Build pipeline
- Persistent DOIs

Distribution channels

- DAP? what for? why?
- GitHub releases
- pipy, conda
- CRAN
- matlab functions etc.
- Server for service
- Debian/RPM package
- Installer for Windows

Code versioning

Whether it sits on [GitHub](#), [Gitlab](#), [Bitbucket](#), [CSIRO bitbucket server](#) should largely not matter in theory, for code versioning per se. Some additional services (build workflows, third party code coverage, etc.) are however more readily accessible from e.g. GitHub, and can be compelling.

Data and Software as first class publications

Publishing in the [The Journal of Open Source Software](#)

Overlapping the What and the How: [The art of community](#) by Jono Bacon (Ubuntu employee)

Increasingly, my guess is that there is a commercial or impact potential in platform products [Effective platform product](#). Senaps, modelling engines, catalogues.

A bit dated but very short, from a Google employee: [Release engineering](#)

Stories from the frontline

Circa 2010 R and .NET interop

Back to 2010 trying to release a software open source and get advice on this.

Case study: streamflow forecasting

An example: [Workspace software](#)

[Streamflow forecasting](#)

Dimensions

- Git everywhere.
- [Entry Documentation](#), [swift2 python package technical reference](#)
- Reference doc (docstrings); Architecture.
- [Dependency management]: versioning [software components](#), released on their own
- Unit testing, code coverage: [codecov](#).
- Tutorials, vignettes, sample workflows
- Sample data
- [Build pipeline](#)
- Persistent DOIs

Distribution channels

- DAP?: [efts package](#)
- GitHub releases
- `refcount` via [pipy](#), [conda](#).
 - Other more telling for impact: [nuget for dynamic-interop](#).
- CRAN
- matlab functions etc.
- Server for service
- Debian/RPM package
- Installer for Windows

Overheard recently

“In case you’re interested, our experience trying to comply with established standard so far has been:

- We obtained permission for releasing a Puppet module as open source at management level. At that time, for open source releases, there were 2 options:
 - New IP with CSIRO BSD licence
 - Contribution using existing licence of software project.
- We were developing a new module for a software which is not exactly a new IP. However it’s also not a contribution to an existing project with established licence so it seemed inappropriate to adopt the targeted software’s licence. To be on the safe side, we picked the 1st option start using CSIRO BSD licence.
- Unfortunately CSIRO BSD licence is not compliant with development kit which requires the licence to be listed in SPDX. So we tried to make it compliant with SPDX and approached the legal team.

- IIRC the legal team wasn't keen to liaising with SPDX at the time, but we got permission from them to explore and liaise between SPDX and legal.
- IIRC we went as far as trying to submit the licence for SPDX legal to process. However there was a fair bit of confusion in relation to whether CSIRO BSD licence was necessary or not (the license proliferation question). IIRC the answer we got from legal wasn't sufficiently clear for our level of legal and licencing knowledge in order to move forward.
- After a few back and forth and not understanding what we're doing, it all became too hard/risky so we abandoned the idea and looked for alternatives.

We've had 2 attempts at this, both ended up the same way."

Discussion