Workshop:

How do software management plans advance open science and contribute to research quality?

Carlos Martinez Ortiz, Maria Cruz & Maaike de Jong

Open Science Conference 2023
27-29 June
doi: 10.5281/zenodo.8093501

Workshop programme

09:00 - Welcome

09:05 - Presentation: **Research software and open science**

09:20 - Presentation: **Software management plans**

09:35 - Introduction to breakout session 1

09:40 **- Breakout session 1: Classifying research software**

09:55 - Breakout session 1 discussion

10:05 - Short break

10:10 - Introduction to breakout session 2

10:15 - **Breakout session 2: Get started with a software management plan**

10:40 - Breakout session 2 discussion

10:55 - Closing the workshop

Open science conference

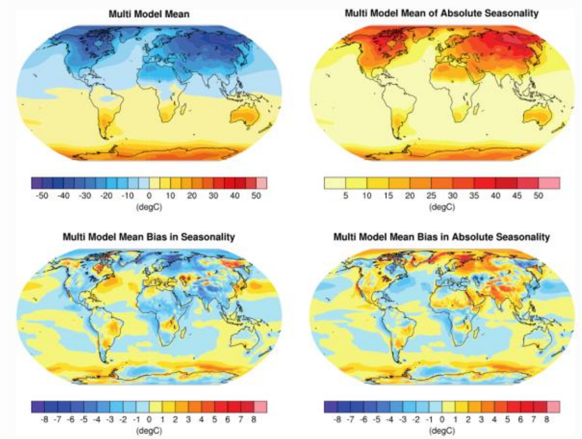
Research software and Open Science

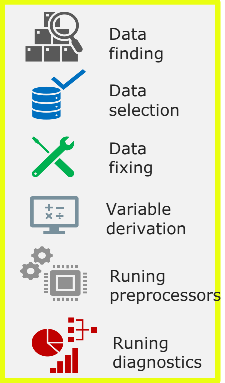
Maaike de Jong

Open software practices have impact

ESMValTool

Earth System Model Evaluation Tool




Data finding

Data selection

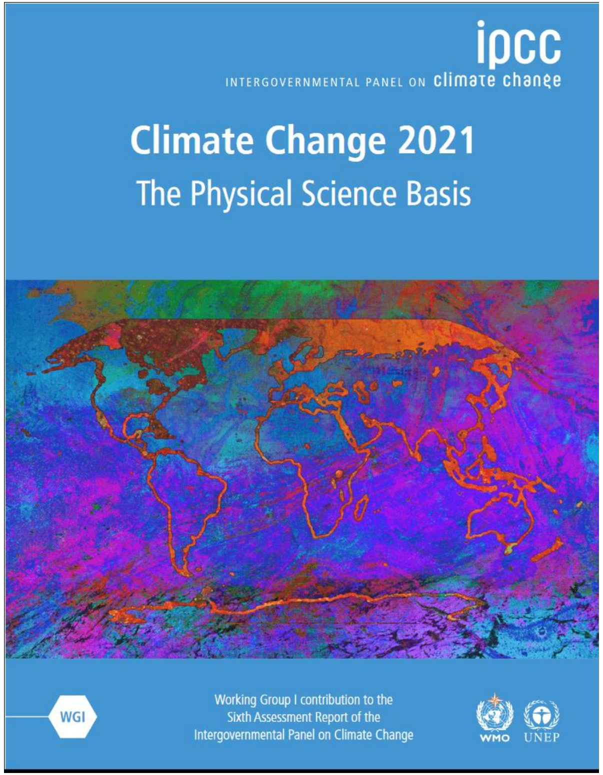
Data fixing

Variable derivation

Runing preprocessors

Runing diagnostics

- Existing tool with a small user base
- Best practices improved ease of use
- Grown to a large community of climate researchers


ipcc

INTERGOVERNMENTAL PANEL ON climate change

Climate Change 2021

The Physical Science Basis

WGI

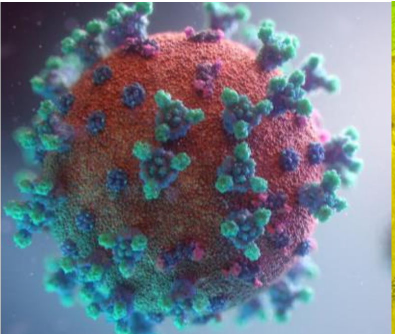
Working Group I contribution to the
Sixth Assessment Report of the
Intergovernmental Panel on Climate Change

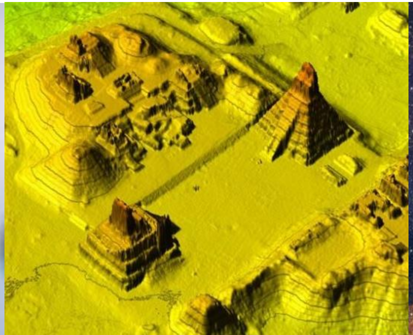
WMO UNEP

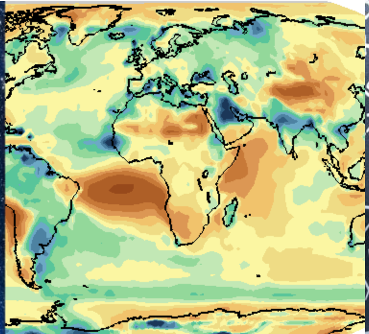
https://www.ipcc.ch/report/ar6/wg1/

Research software is crucial for today’s academic research

Climate modelling, gene sequencing, radio telescopes, particle accelerators, natural language processing, etc.,… all depend on research software!







Some statistics

- Survey among UK universities 2014: 92% of academics use research software, 70% say it’s impossible to do research without it

- OECD International Survey of Scientific Authors 2018: research software is made across all research fields, with an average of 33% of research resulting in new code.

- Nation-wide analysis France 2021: 48% of French publications mention using software, 17% mention creating software

https://www.software.ac.uk/blog/2014-12-04-its-impossible-conduct-research-without-software-say-7-out-10-uk-researchers

2018 OECD International Survey of Scientific Authors: https://doi.org/10.1787/1b06c47c-en

Barometre de la Science Ouverte: https://barometredelascienceouverte.esr.gouv.fr/

Open science conference

Research software as part of open science

- Transparency, reproducibility

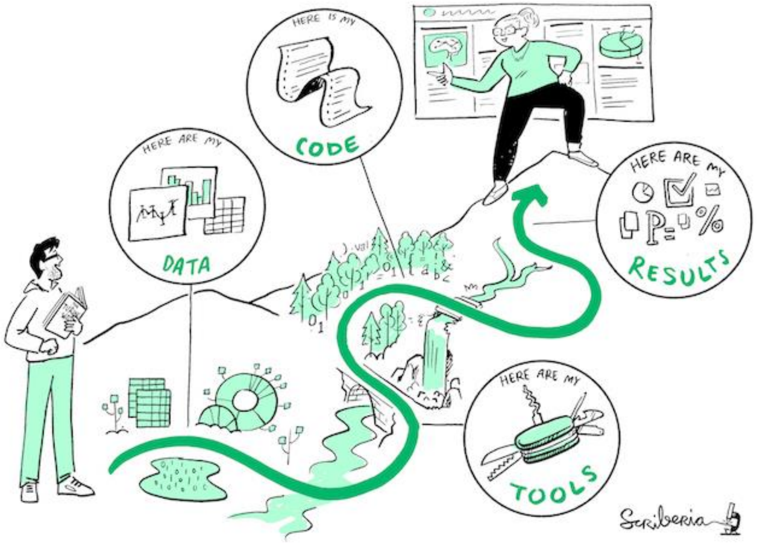


Illustration by Scriberia for The Turing Way Community Shared under CC-BY 4.0

Research software as part of open science

- Transparency, reproducibility

- Software reuse, collaboration, communities

OPEN SOURCE COLLABORATION
BY THE COMMUNITY
FOR THE COMMUNITY
OPEN NOT SOURCE
DRIVEN BY NEEDS OF THE INSTITUTION OR THE INDIVIDUAL
Scriberia

Illustration by Scriberia for The Turing Way Community Shared under CC-BY 4.0

Research software as part of open science

- Transparency, reproducibility

- Software reuse, collaboration, communities

- Recognition of the work of researchers and RSEs

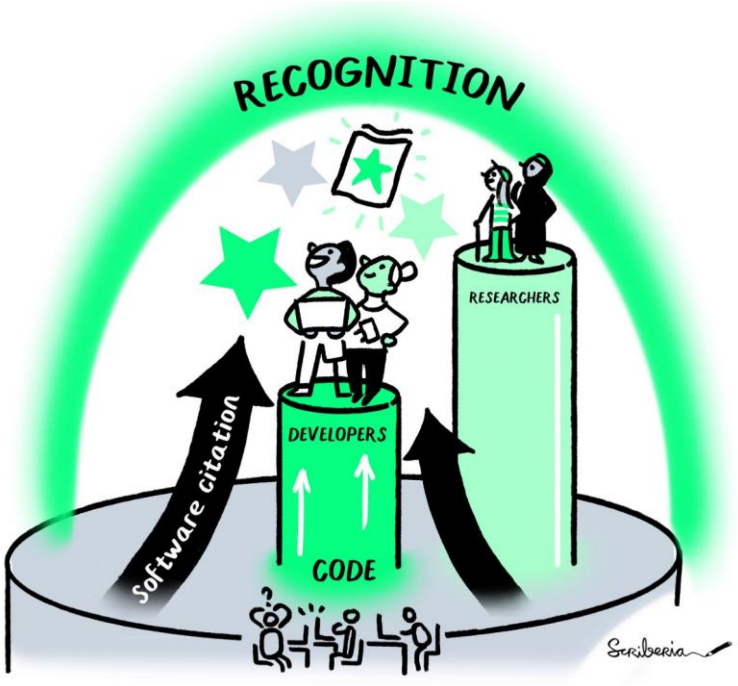


Illustration by Scriberia for The Turing Way Community Shared under CC-BY 4.0

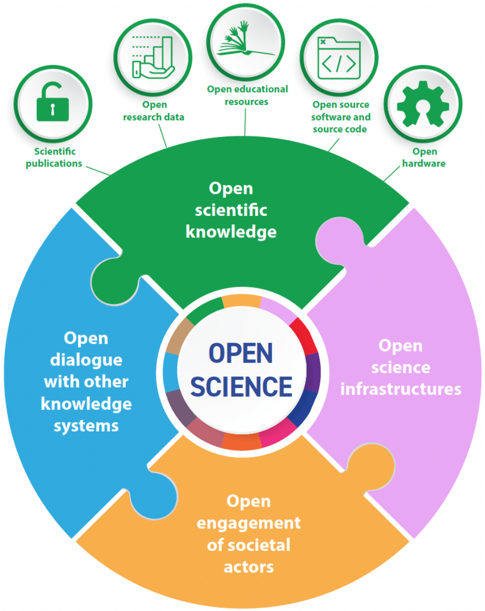
Recognition of importance of research software

- Open source software included in UNESCO recommendations on open science

- Increasing attention for research software in guidelines, policies, and support

- But recognition of its importance is still limited compared to, for example, research data

unesco


Scientific publications
Open research data
Open educational resources
Open source software and source code
Open hardware
Open scientific knowledge
Open dialogue with other knowledge systems
OPEN SCIENCE
Open science infrastructures
Open engagement of societal actors

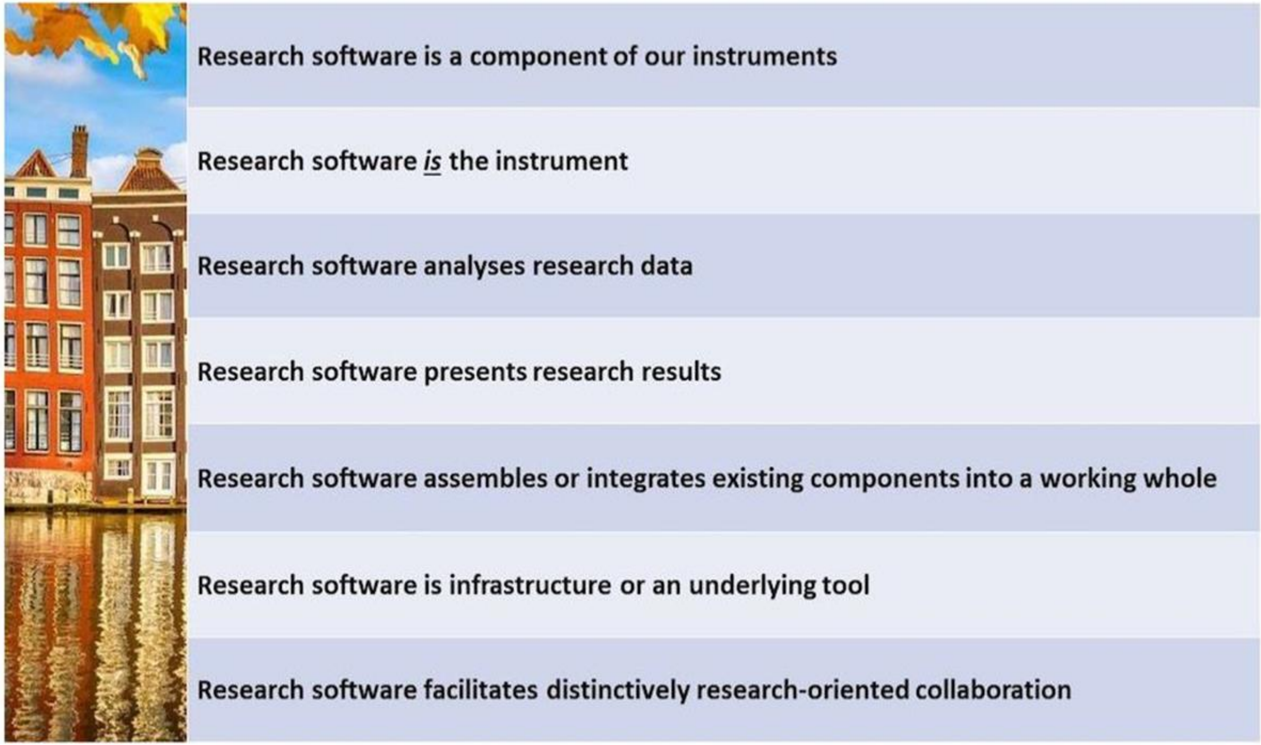
Defining research software

“Source code files, algorithms, scripts, computational workflows and executables that were **created during the research process or for a research process**.”

“Software components (e.g. operating systems, libraries, dependencies, packages, scripts, etc.) that are used for research but were not created during or with a clear research intent should be considered software in research and not research software.”

Gruenpeter et al. (2021). Defining Research Software: a controversial discussion. Zenodo. https://doi.org/10.5281/zenodo.5504016

The roles of research software



Research software is a component of our instruments

Research software *is* the instrument

Research software analyses research data

Research software presents research results

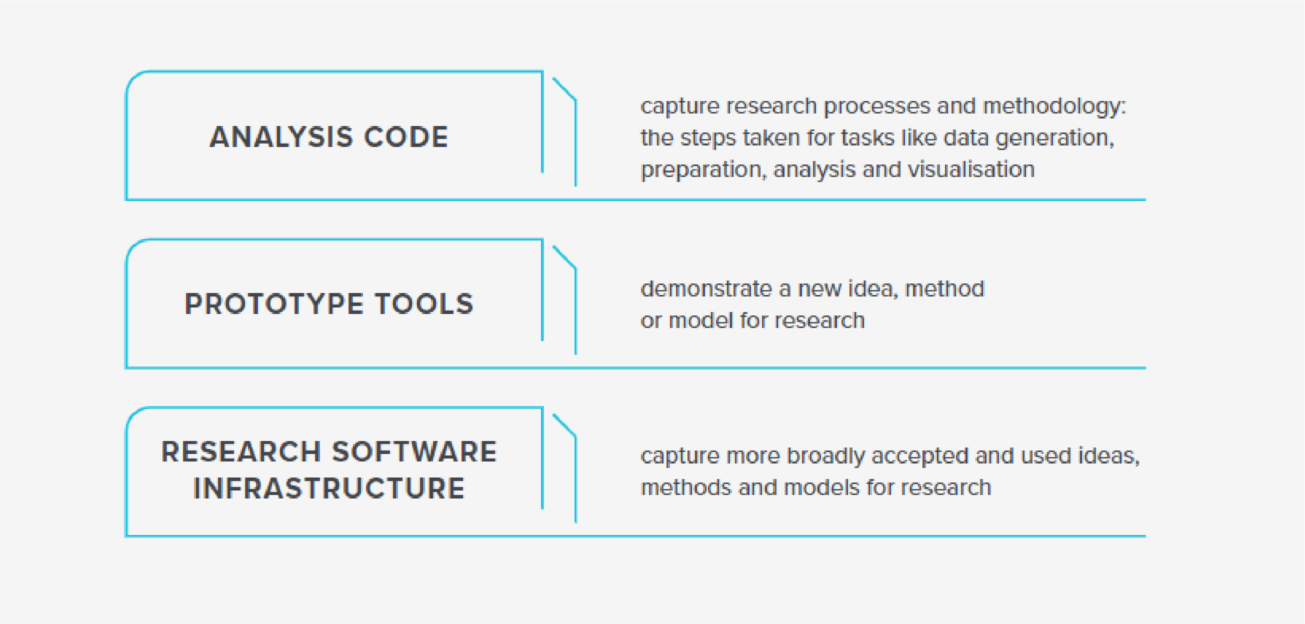
Research software assembles or integrates existing components into a working whole

Research software is infrastructure or an underlying tool

Research software facilitates distinctively research-oriented collaboration

Rob van Nieuwpoort and Daniel S. Katz, March 2023, https://upstream.force11.org/defining-the-roles-of-research-software

Software classification



ANALYSIS CODE

capture research processes and methodology: the steps taken for tasks like data generation, preparation, analysis and visualisation

PROTOTYPE TOOLS

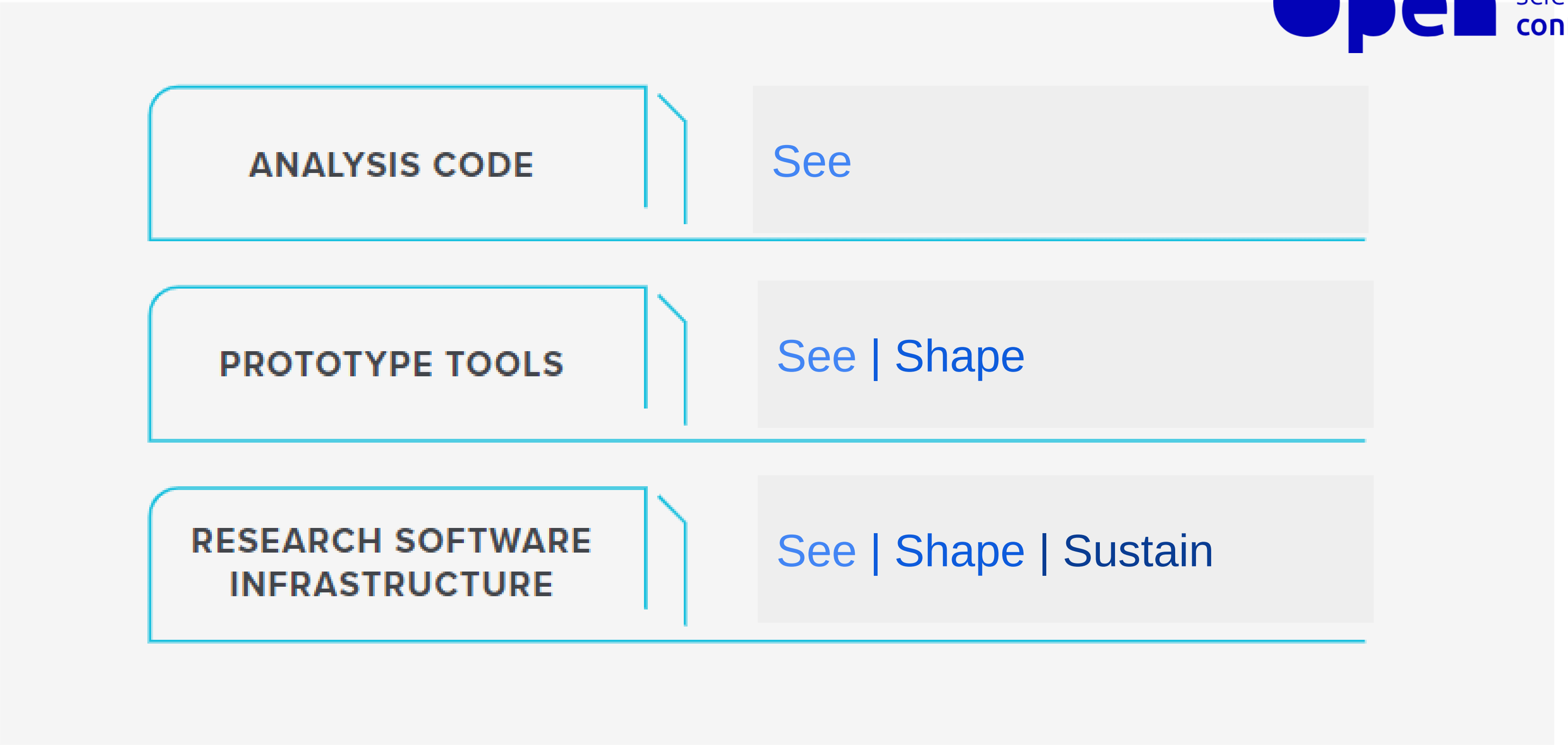
demonstrate a new idea, method or model for research

RESEARCH SOFTWARE INFRASTRUCTURE

capture more broadly accepted and used ideas, methods and models for research

Software classification according to ARDC (Australian Research Data Commons)

Australian Research Data Commons. (2022). A National Agenda for Research Software (1.0). Zenodo. https://doi.org/10.5281/zenodo.6378082



ANALYSIS CODE — See

PROTOTYPE TOOLS — See | Shape

RESEARCH SOFTWARE INFRASTRUCTURE — See | Shape | Sustain

Software classification according to ARDC (Australian Research Data Commons)

Australian Research Data Commons. (2022). A National Agenda for Research Software (1.0). Zenodo. https://doi.org/10.5281/zenodo.6378082

Software management: no one size fits all

See Research Software

- Make research software more visible

Shape Research Software

- Make new research software appropriately shaped for broadest suitable reuse

Sustain Research Software

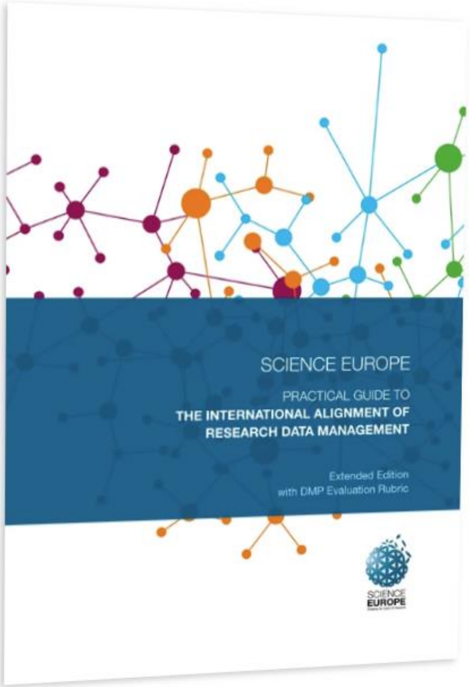
- Make existing research software more easily maintained

Any questions so far?

Software Management Plans

Carlos Martinez Ortiz

Let’s start with Data Management Plans



SCIENCE EUROPE

PRACTICAL GUIDE TO
THE INTERNATIONAL ALIGNMENT OF
RESEARCH DATA MANAGEMENT

Extended Edition
with DMP Evaluation Rubric

SCIENCE EUROPE

- Historically data has received more attention
- NWO looked at a sample of ~100 DMPs, with one question focusing on software
  - **56% listed research software as one of the outputs**
  - For **12%** of the DMPs **software is the main output or one of the main outputs** of the project
  - In general, **the answers are very short**; sometimes just a sentence

How is software different from data?

- It is **executable** – “uniquely actionable form of knowledge representation”; it can manifest itself as a tool or instrument.

- It is defined as a form of creative labour with **specific intellectual property rights** (that are different from those applied to data).

- It exhibits **complex interdependencies** – most software depends on other software and the interconnections between software are global.

- It is **not static**; it **needs to be actively maintained** to continue to meet its purpose over time.

Research software management

Ensuring **accessibility, reusability,** and **sustainability** of research software

Different aspects of research software management:

- Development
- Documentation
- Publication & access
- Archiving
- Community & collaboration

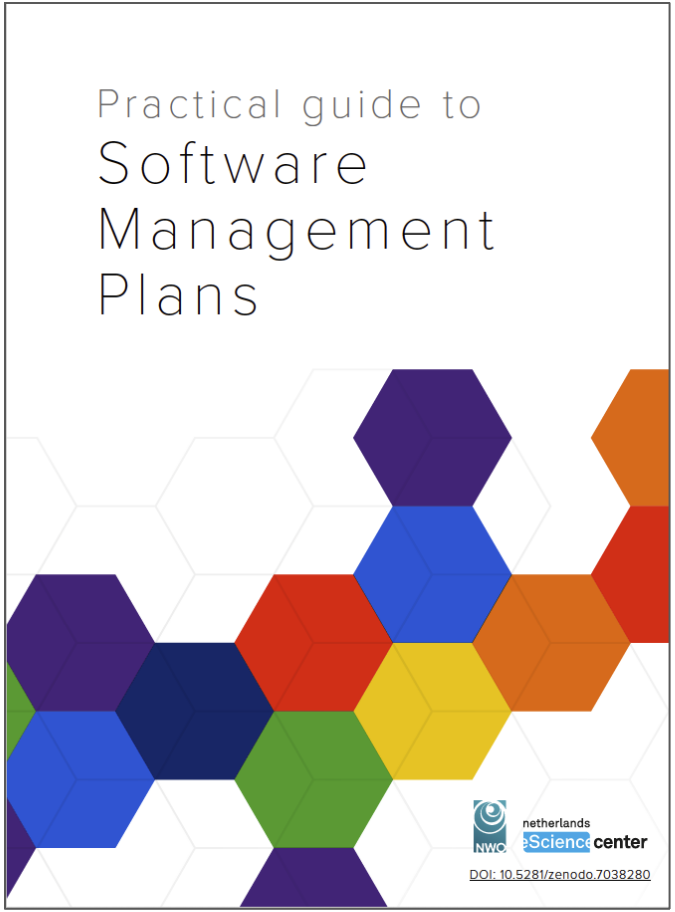
Software Management Plan (SMP)

An SMP makes explicit what research software does, who it is for, what the outputs are, who is responsible for the release and to ensure that the software stays available to the community (and for how long).

NOT another bit of admin: software management leads to better science!

Guidelines for SMPs

- Working group Dec 2021 - Oct 2022
- International sounding board
- Community consultation rounds & workshops



Practical guide to

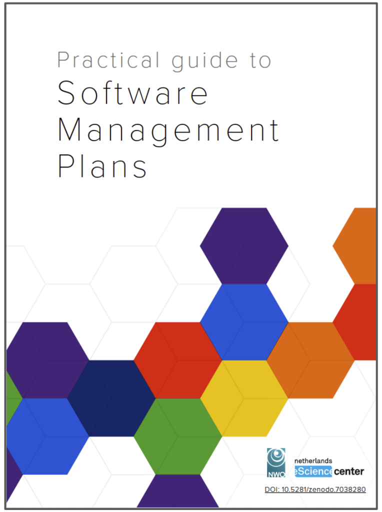
Software Management Plans

NWO netherlands eScience center

DOI: 10.5281/zenodo.7038280

doi:10.5281/zenodo.7589725

Practical guide to SMPs



Practical guide to
Software
Management
Plans

netherlands eScience center

DOI: 10.5281/zenodo.7038280

- Guide to create an SMP template (with examples)

- Core requirements

  What should your SMP consider?

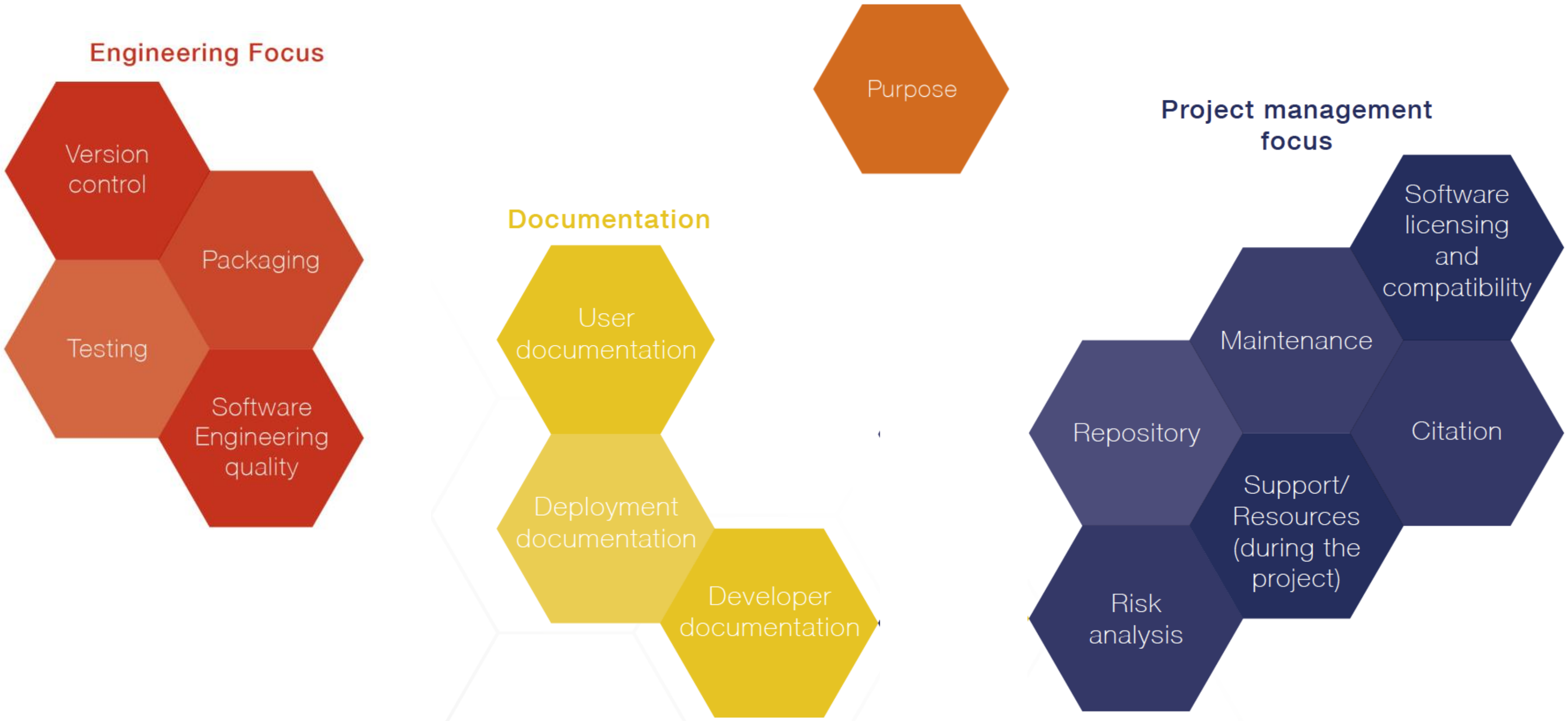
- Different levels of management: not all software is created equal!

  Low: e.g. a script to generate figures for a paper

  Medium: e.g. simulation software that implements a model

  High: e.g. pipeline pre-processing all data coming from a telescope

Practical guide to SMPs



Purpose

Engineering Focus

Version control

Packaging

Testing

Software Engineering quality

Documentation

User documentation

Deployment documentation

Developer documentation

Project management focus

Software licensing and compatibility

Maintenance

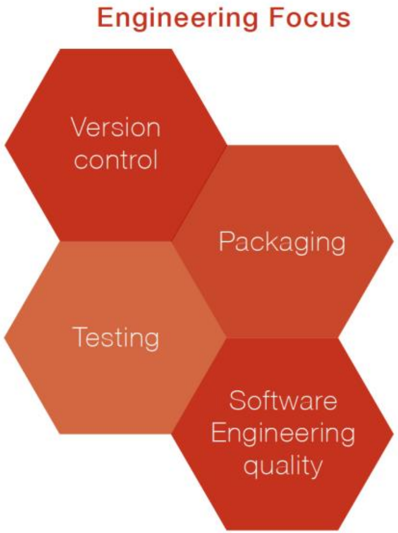
Repository

Citation

Support/ Resources (during the project)

Risk analysis

Practical guide to SMPs



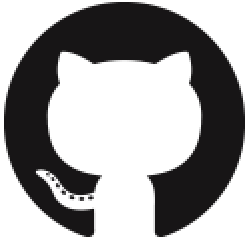
Engineering Focus

Version control

Packaging

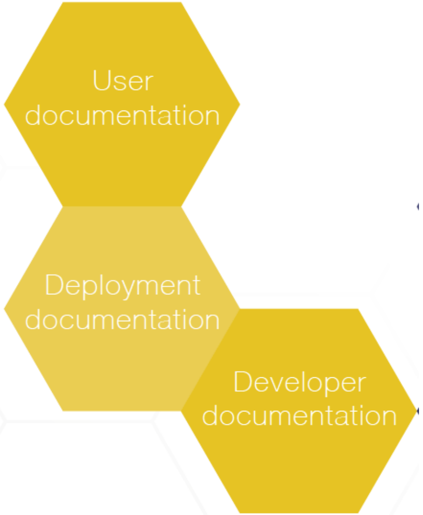
Testing

Software Engineering quality



Practical guide to SMPs

Documentation


User documentation

Deployment documentation

Developer documentation

Open science conference

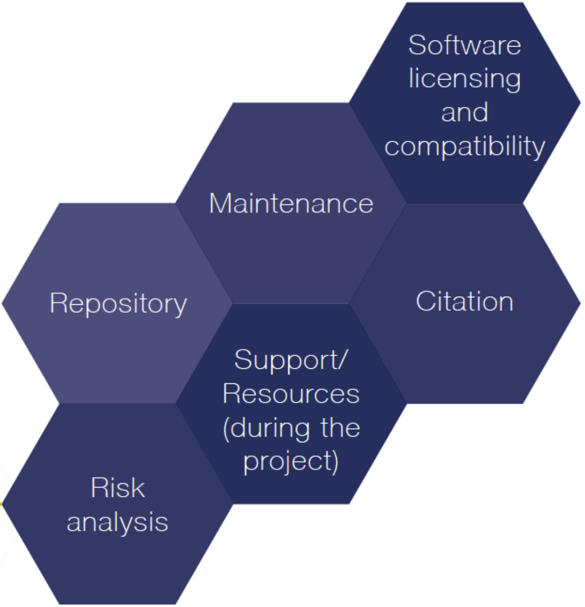
Practical guide to SMPs

zenodo

Software Heritage

THE APACHE SOFTWARE FOUNDATION ESTABLISHED 1999

Project management focus



Software licensing and compatibility

Maintenance

Repository

Citation

Support/ Resources (during the project)

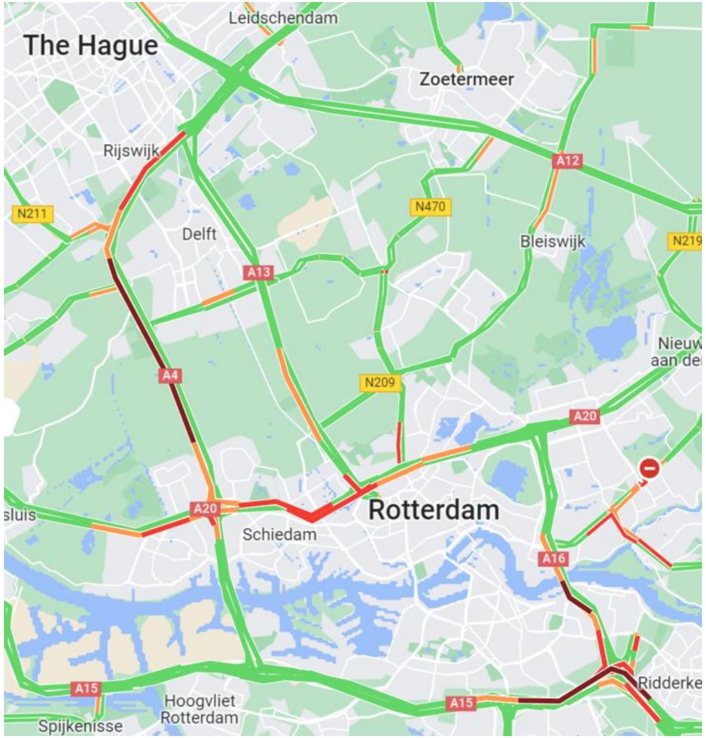
Risk analysis

CFF INIT

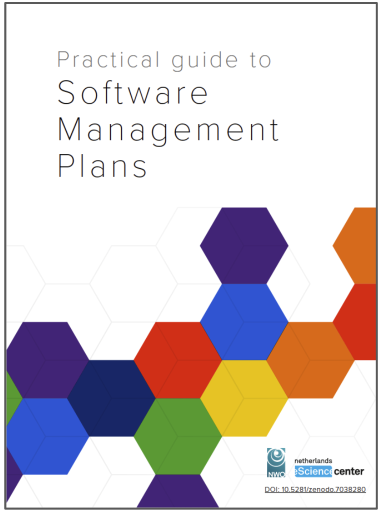
Example: where can good practices prevent mistakes

Testing traffic data

- Query broke, every record was returned twice
- Problem: nothing broke, but numbers were double what they should be.
- Writing tests helped catch the bug later.



Management requirements per level



Practical guide to Software Management Plans

NWO netherlands eScience center

DOI: 10.5281/zenodo.7038280

	LOW	MEDIUM	HIGH
Purpose	X	X	X
Version control	X	X	X
Repository		X	X
User documentation		X	X
Software licencing		X	X
Deployment documentation		X	X
Citation		X	X
Developer documentation		X	X
Testing		X	X
Software engineering quality		X	X
Packaging		X	X
Maintenance		X	X
Support			X
Risk analysis			X

Any questions?

Quick overview of requirements

- Version control → https://the-turing-way.netlify.app/reproducible-research/vcs.html
- Testing → https://the-turing-way.netlify.app/reproducible-research/testing/testing-guidance.html
- Packaging → https://the-turing-way.netlify.app/reproducible-research/renv/renv-package.html
- Documentation → https://guide.esciencecenter.nl/#/best_practices/documentation
    - User
    - Deployment
    - Developer
- Citation → https://the-turing-way.netlify.app/communication/citable/citable-cff.html#cm-citable-cff
- Software licensing and compatibility → https://the-turing-way.netlify.app/reproducible-research/licensing/licensing-software.html and https://the-turing-way.netlify.app/reproducible-research/licensing/licensing-compatibility.html