

What, why and how:

Software Management Plans (SMPs) & best practices in research software development

Carlos Martinez-Ortiz

netherlands
eScience center

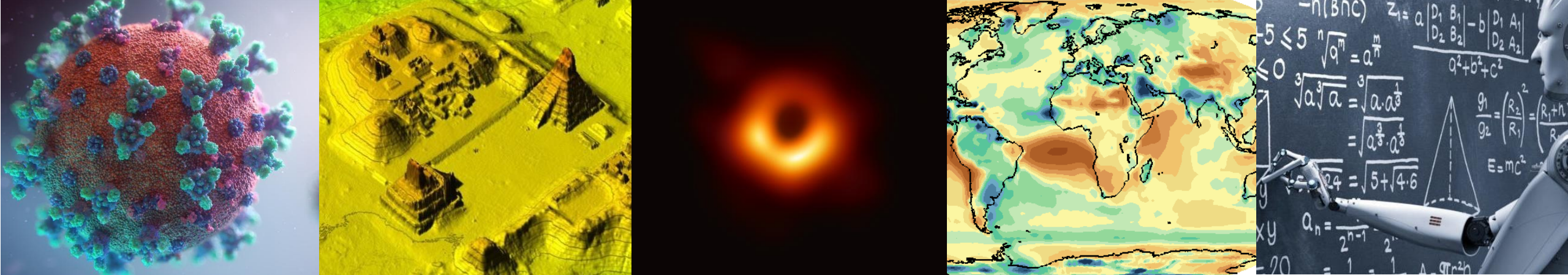
e

e

WHY?

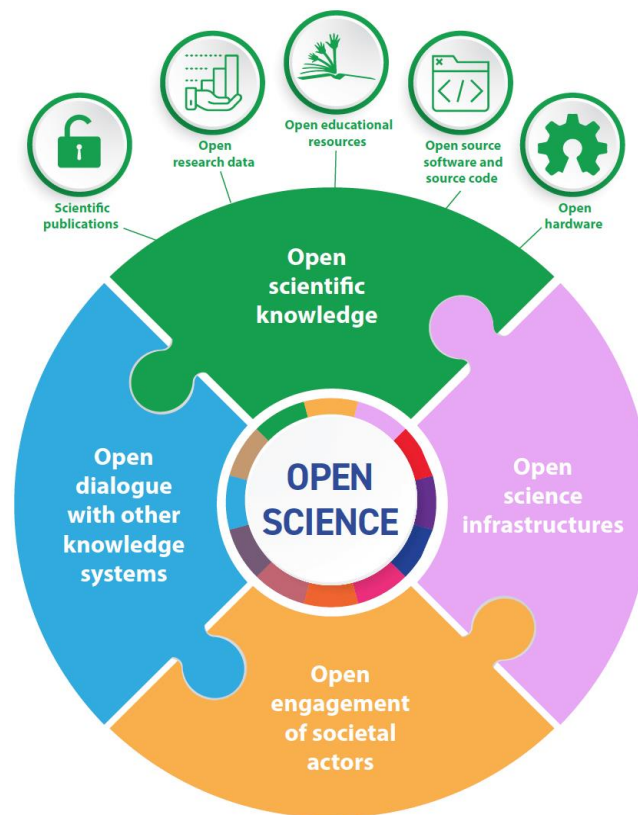
Why should you have a SMP?

netherlands
eScience center



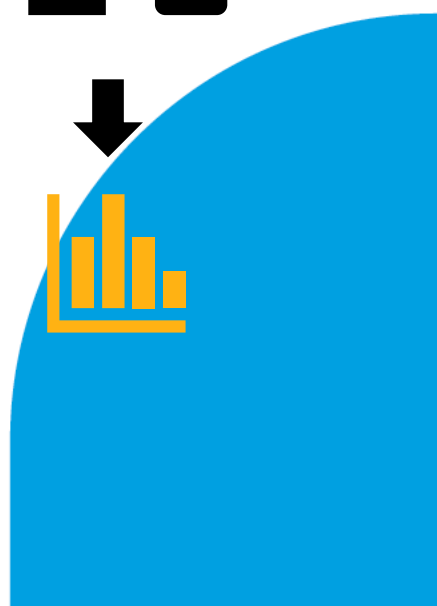
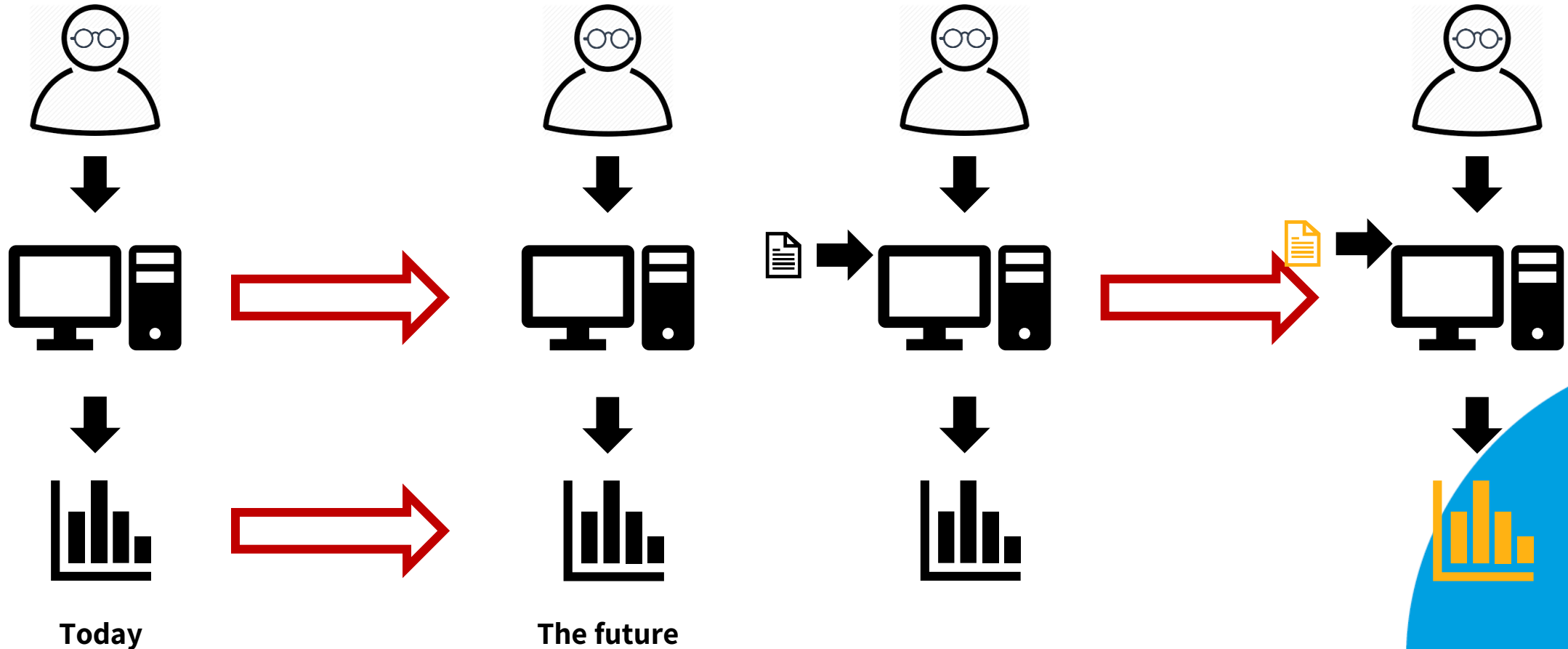
Research software is crucial for today's academic research

Open source software included in UNESCO recommendations on open science

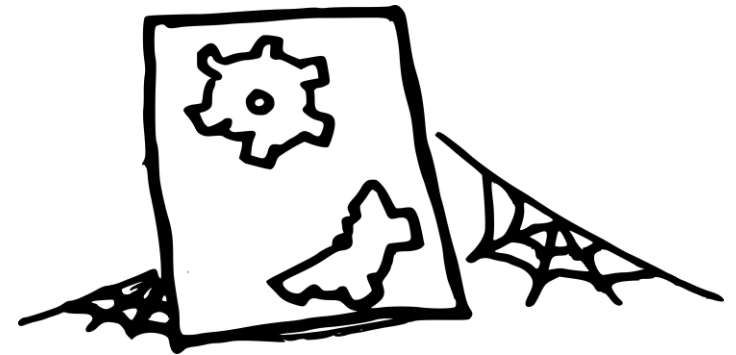
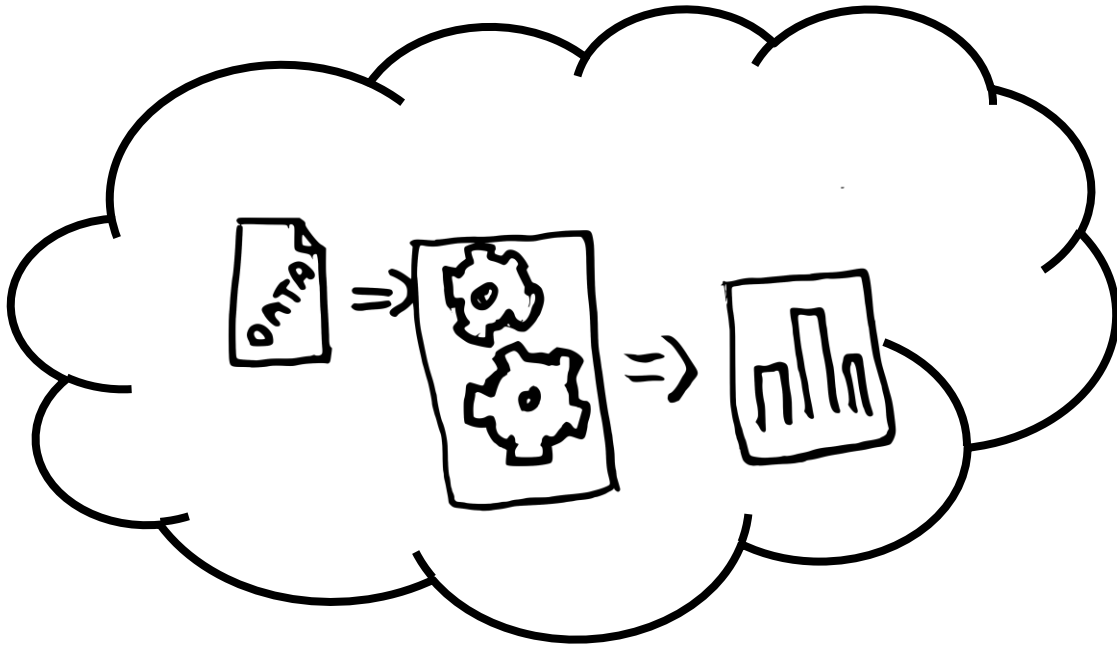


Various studies highlight importance of research software, but attention for software management is still limited

Research software reproducibility



Software breaks over time



WHAT?

What should be covered in your SMP?

Why use a SMP?

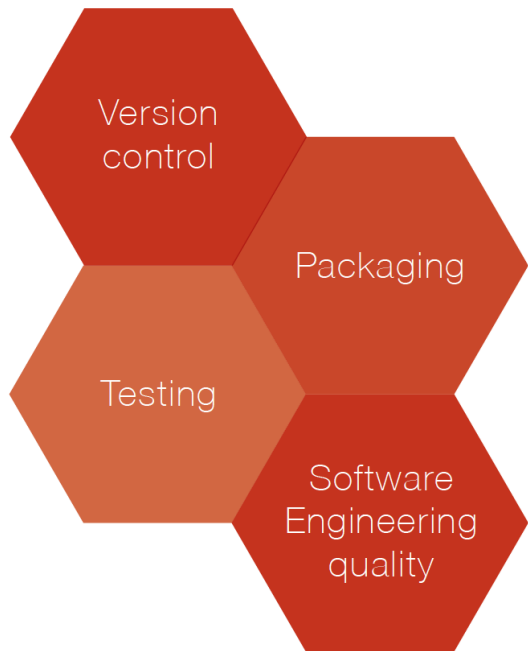
- Make technical choices explicit
- Plan for necessary resources
- Assess whether new software is really needed
- **NOT** another bit of admin: software management leads to better science!

Practical guide to
Software
Management
Plans

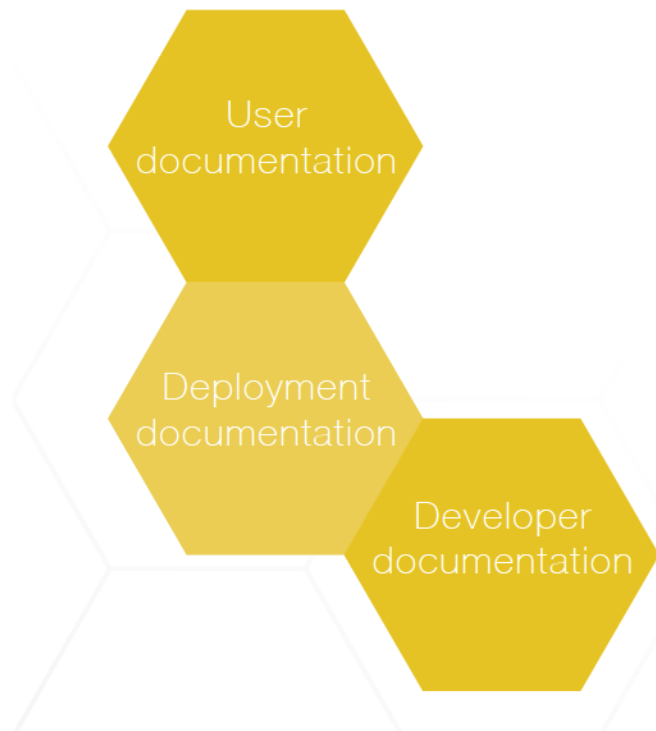


SMP requirements

Engineering Focus



Documentation



Purpose

Project management focus



Software classification

**Low
management
software**

**Medium management
software**

**Mission critical
software**

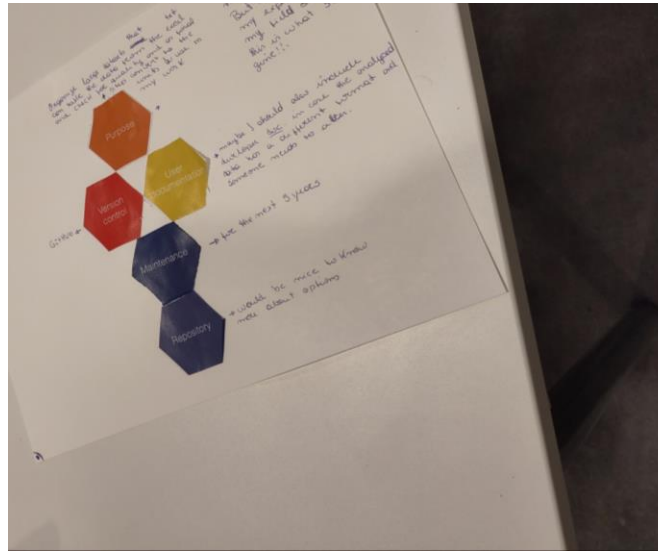


SMP classes and requirements

	I	II	III
I	✓		
II		✓	
III		✓	✓
IV		✓	✓
V			✓
VI			✓



Tailor made SMPs



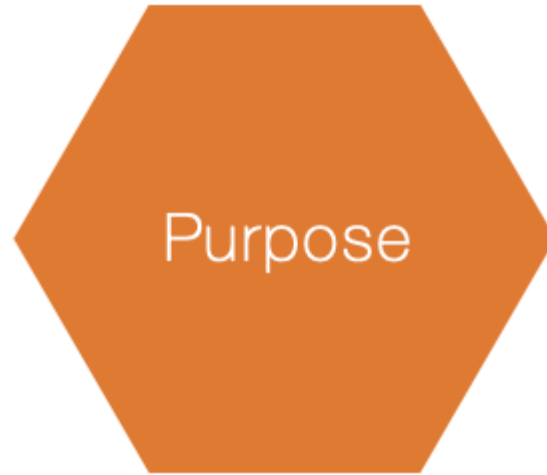
e

e

HOW?

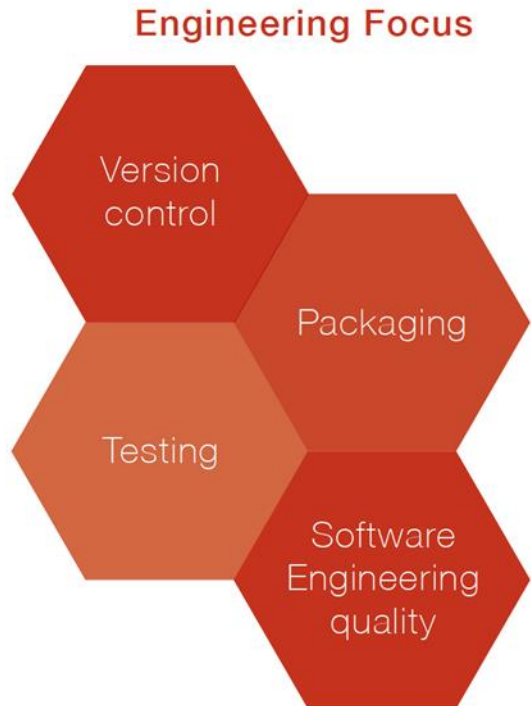
How to go about it?

netherlands
eScience center



- What problem does it solve?
- Who is the intended audience?
- What are its advantages and limitations?





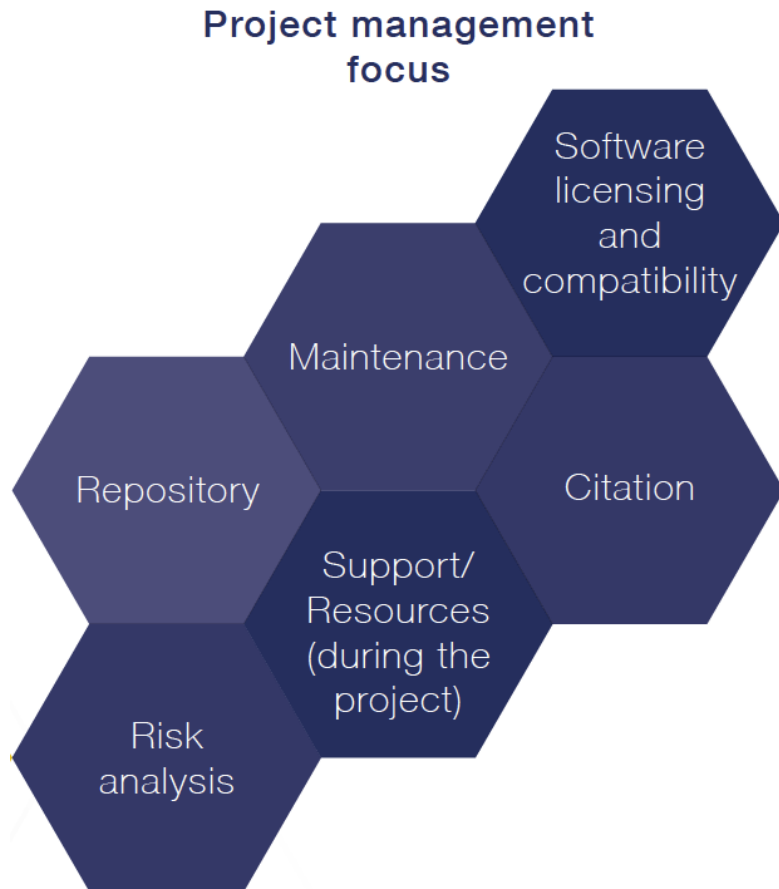
- What version control will you use?
- How will you test the software?
- How will the project be organized or packaged?
- How will you manage code quality?





- What documentation will you provide
 - for users?
 - for developers?
- How will you manage system requirements (e.g. dependencies)?





- What license will you use?
- How will the software be maintained? For how long?
- How should the software be cited?
- What repository(ies) will you use?
- What resources will you need?
 - e.g. infrastructure, personnel, training, hardware...
- What risks and challenges do you foresee?

Research Software Support

- FAIR software
- Reusability
- Publishing & citing

The screenshot shows the 'Research Software Support' website. At the top, there is a navigation bar with the title 'Research Software Support' and links for 'Modules', 'About', and 'Contribute'. The Netherlands eScience Center logo is in the top right corner. Below the navigation bar, a horizontal menu displays several topics: 'Learning objectives', 'What is software?', 'Software and data', 'Research Software', 'Research software', 'Research software?', and 'Research life cycle'. The main content area features a large blue and purple graphic with the text 'Research software' and the Netherlands eScience Center logo. A 'Speaker notes' panel on the right indicates 'No notes on this slide.' At the bottom, a footer states: 'These materials are developed by the Netherlands eScience Center. Licensed under CC-BY 4.0 unless otherwise noted.'

<https://tinyurl.com/researchsoftware-2024>

Best practices for Sustainable Software



WP1 - Develop guidance on sustainable software

1. A concise overview of key learning resources
2. Examples of software from selected NES domains and of different characteristics that follow best practices.
3. Completed Software Management Plans for those examples.

WP2 - Provide tools to facilitate application of the guidance

1. Open-source research software templates based on the guidance
2. Open-source conformity testing tool for the guidance



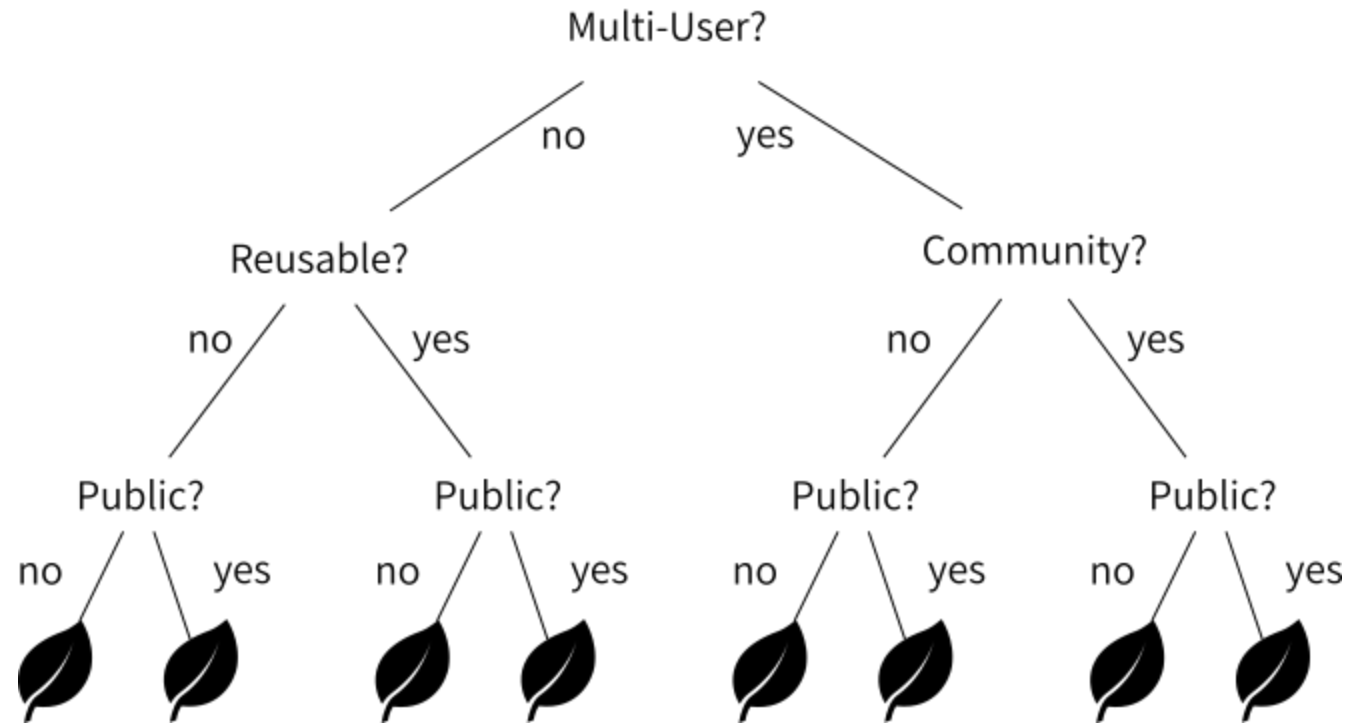
WP4 - Training, community building, and dissemination

1. Train the Trainer **workshop** introducing trainers and training coordinators to the resources developed in this project
2. Training **workshop** focused on the resources developed in this project, aimed at researchers
3. Adaptation of existing training provided by project partners in the NES domain, based on the project results
4. **Three "Bring Your Own Code" sessions**
5. Blogpost on project results, lessons learned and recommendations for next steps

WP3 - Improve and integrate digital infrastructure for sustainable software

1. The 4TU.ResearchData (4TU.RD) software will provide an API to allow the Research Software Directory (RSD) to harvest metadata of software from 4TU.RD.
2. The RSD will be extended to harvest metadata of software archived in 4TU.RD.
3. The RSD will be extended to present the domain specific collection of software for the NES community.
4. 4TU.RD will extend its metadata capturing for software deposited at 4TU.RD to align with the metadata of the RSD.
5. The GitLab instance of UL will be integrated with 4TU.RD to showcase the chain from developing/hosting code, publishing code to showcasing it in the RSD

SMP decision tree



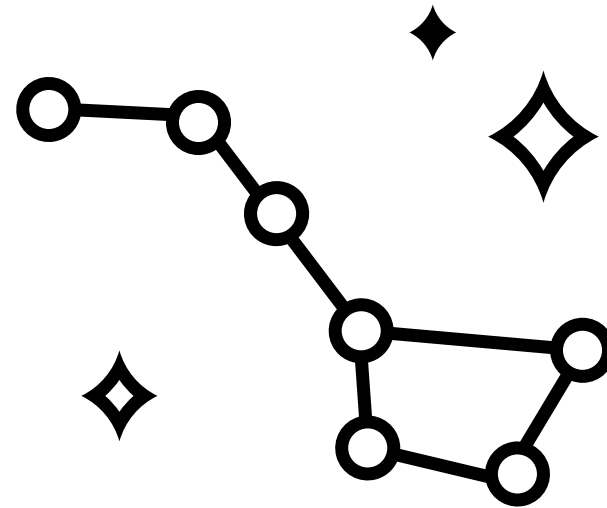
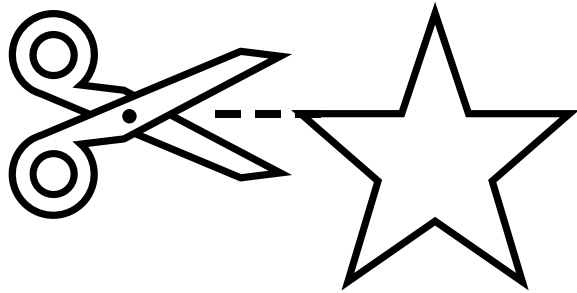
The Turing Way

- <https://book.the-turing-way.org/>
 - Version Control
 - Licensing
 - Making Research Objects Citable
 - Code Testing
 - Code Reviewing Process



Software templates (⚠️ WIP ⚠️)

- A starting point with the structure of your project



To summarise

- **Why** should you have a SMP?
 - Software is a key component of making research reproducible
- **What** should be covered in your SMP?
 - Depending on your specific needs
- **How** to go about it?
 - We are building tools to support you
- Reach out to us!



Let's stay in touch

 www.eScienceCenter.nl

 c.martinez@esciencecenter.nl

 [c-martinez](https://github.com/c-martinez)

 [@neocarlitos](https://twitter.com/neocarlitos)

e