National Open Science Festival
22 October 2024, Maastricht, The Netherlands

Developing better open research software by using modern software templates and conformity check tools

Dr. Serkan Girgin[1,*], Manuel Garcia Alvarez[2,], Robert Ohuru[1], Angelina Momin[1], Dr. Zafer Öztürk[3], Dr. Marianna Avetisyan[3], Dr. Raoul Schram[4]**

s.girgin@utwente.nl m.g.garciaalvarez@tudelft.nl

[1] Faculty of Geo-information Science and Earth Observation
University of Twente

[2] Digital Competence Centre
TU Delft

[3] Digital Competence Centre
University of Twente

[3] Digital Competence Centre
Utrecht University

Following the **best practices** for research software development is fundamental for a **modern, open, and sustainable** research software.

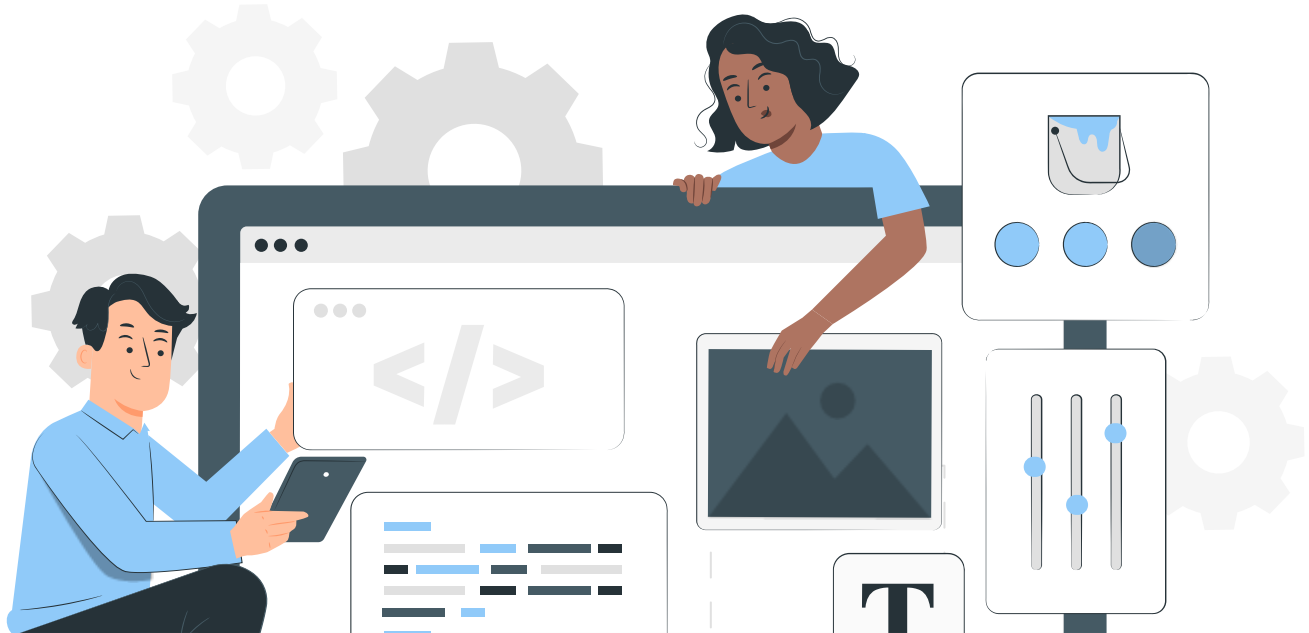


Illustration by Storyset.com

However, **implementing** rapidly changing research software development best practices **is challenging** for research software developers*.

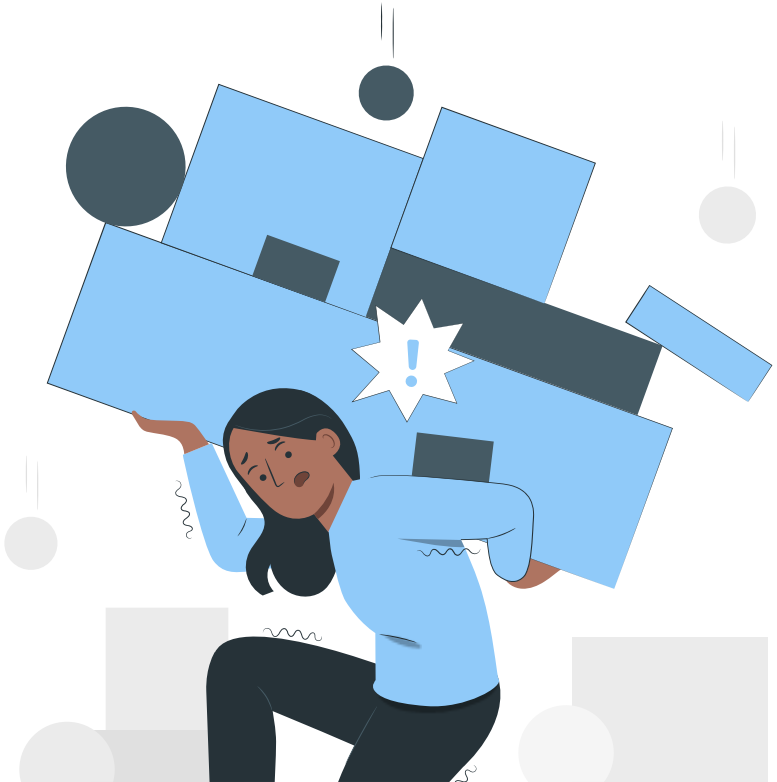


Illustration by Storyset.com

Modern **research software templates** to initiate research software projects and **tools to check conformity** with best practices throughout the development can significantly lower the barriers.

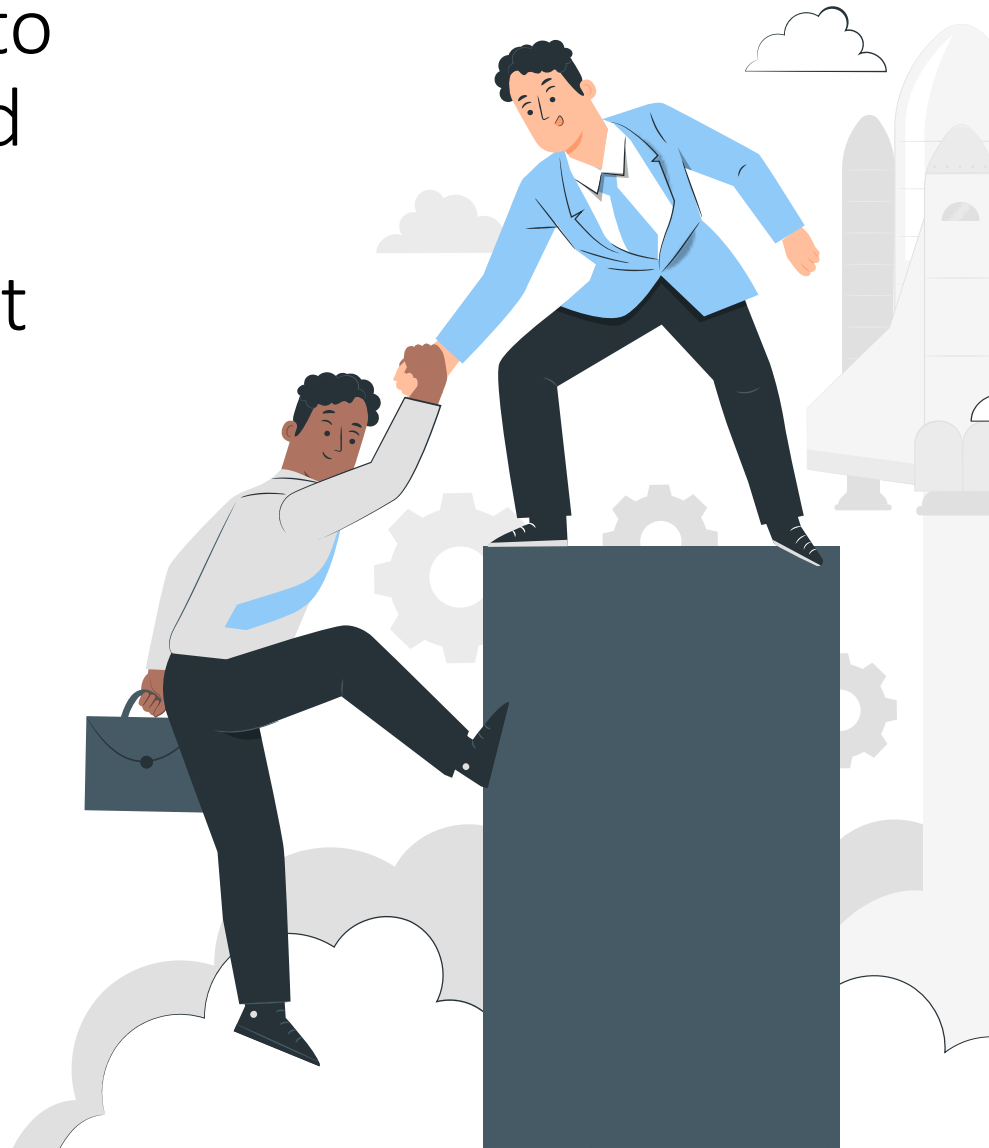
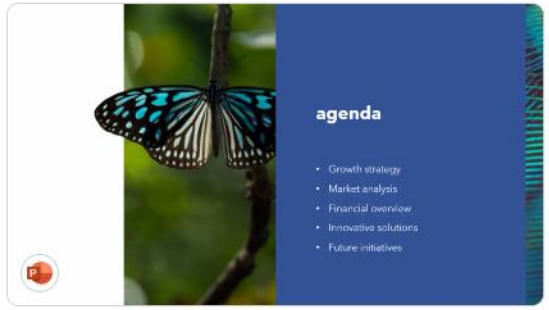


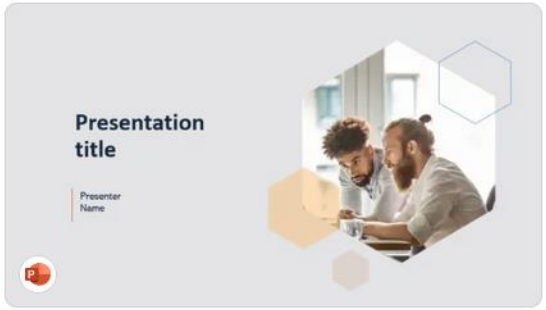
Illustration by Storyset.com

They can also help to **reduce errors**, **enhance software longevity**, and **facilitate collaboration** by automating the use of standards and promoting maintainable code

Illustration by Storyset.com

We use **templates** quite often in our professional life

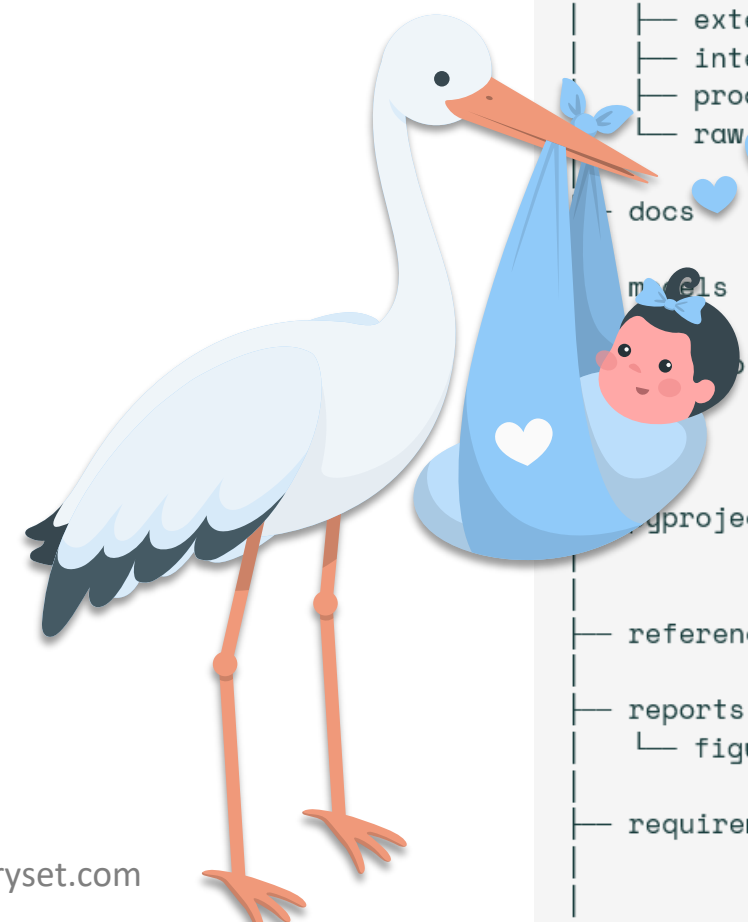




https://create.microsoft.com/en-us/templates/presentations

A research software template provides the **structure and boilerplate code** for a research software project*

- Code and resource layout
- Version control
- License
- Citation
- Documentation
- Code documentation
- Packaging
- Unit testing
- Publishing
- …



```
├── LICENSE            <- Open-source license if
├── Makefile           <- Makefile with convenier
├── README.md          <- The top-level README fo
├── data
│   ├── external       <- Data from third party s
│   ├── interim        <- Intermediate data that
│   ├── processed      <- The final, canonical da
│   └── raw            <- The original, immutable
│
├── docs               <- A default mkdocs projec
│
│   ls                 <- Trained and serialized
│
│   ks                 <- Jupyter notebooks. Nami
│                         the creator's initials
│                         `1.0-jqp-initial-data-e
│
│ project.toml         <- Project configuration f
│                         {{ cookiecutter.module_
│
├── references         <- Data dictionaries, manu
│
├── reports            <- Generated analysis as H
│   └── figures        <- Generated graphics and
│
├── requirements.txt   <- The requirements file f
│                         generated with `pip fre
│
│
```

Illustration by Storyset.com

Do you use software templates to develop research software?

23 / 32

Illustration by Storyset.com

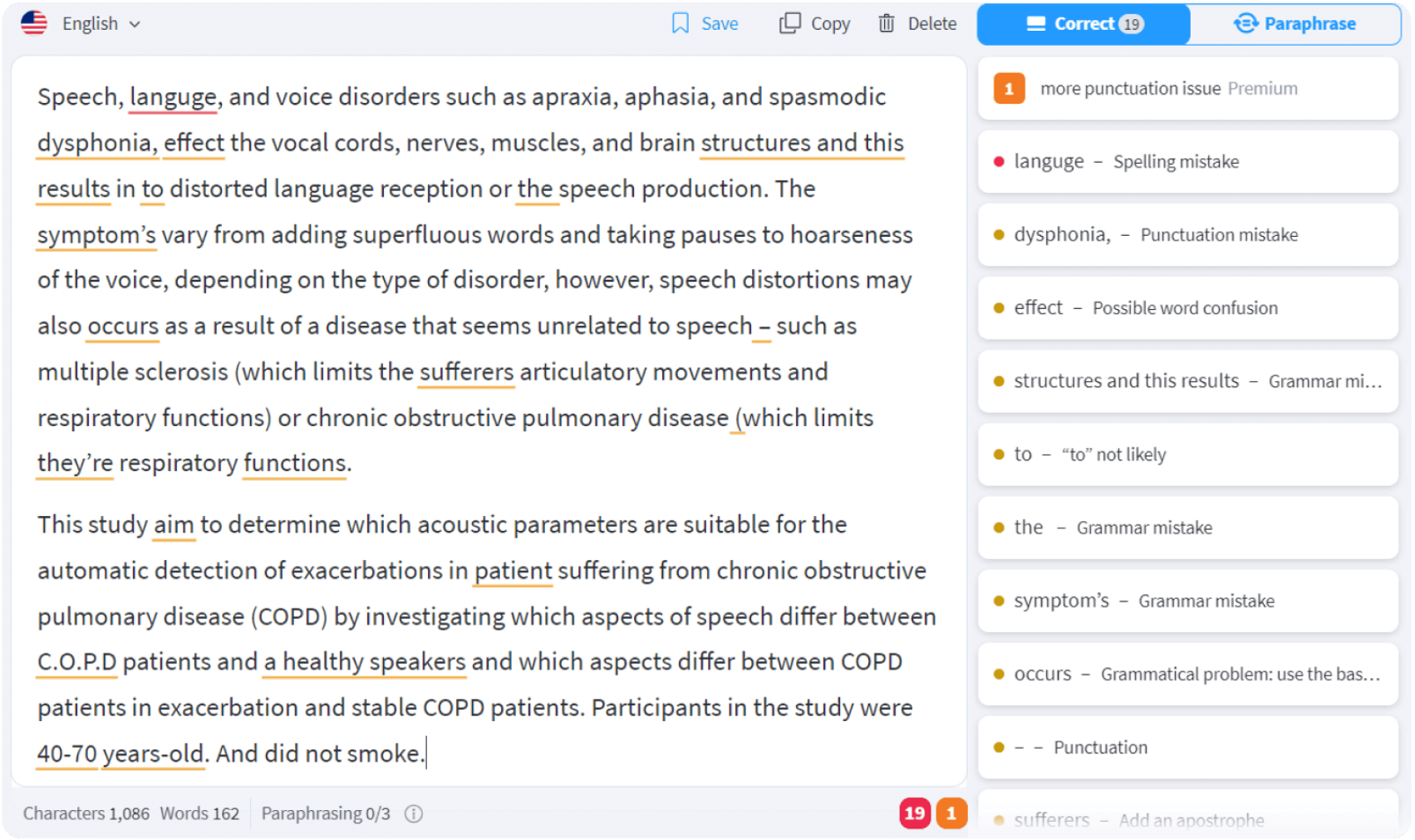
Which software templates to you use?

No

None!

Illustration by Storyset.com

We use **checking tools** quite often in our professional life



https://www.scribbr.com/language-rules/best-grammar-checker

A research software conformity check tool provides insights about the **structure and content** of a research software project*

- Code and resource layout
- Version control
- License
- Citation
- Documentation
- Code documentation
- Packaging
- Unit testing
- Publishing
- …

Illustration by Storyset.com

Do you use conformity check tools while developing research software?

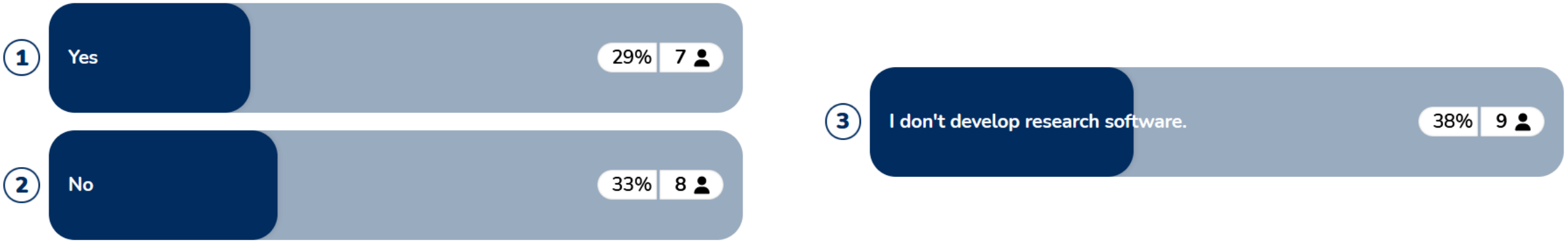


Illustration by Storyset.com

Which conformity check tools do you use?

- Check if docstrings are present
- General linting tools
- Pycharm
- Linter
- Flake8
- CRAN submission check
- Visual Studio
- Standard linter
- GitHub copilot

Illustration by Storyset.com

Some **observations** regarding research software development practices

- The list of practices asked to be followed or needs to be followed, e.g., for better uptake, visibility, recognition, impact, etc.*, is growing.
- Many practices have a wide-range of options available. Even if it is not required to implement them extensively, a good understanding of them is gradually becoming necessary*.
- Use of templates to have a quick implementation of some practices is becoming more and more common among research software developers*.
- Software templates are also promoted by initiatives and institutions.
- The "awesome list" research software templates is growing.
- Many templates are quite similar to each other, especially if they are based on the same technology (e.g. cookiecutter, copier, etc.).
- The use of tools to ensure conformity with research software development best practices is rather limited. This is also not checked by many owner or funder institutions.

Best Practices for Sustainable Software in the Natural and Engineering Sciences aims better research software by providing guidance and tools

- By providing domain-relevant guidelines, tools, infrastructure, and training, the project seeks to support awareness and uptake of best practices during the entire software life cycle.

- WP 1: Develop guidance on sustainable software.
  WP 2: Provide tools to facilitate application of the guidance.
  WP 3: Improve and integrate digital infrastructure for sustainable software.
  WP 4: Training, community building, and dissemination.

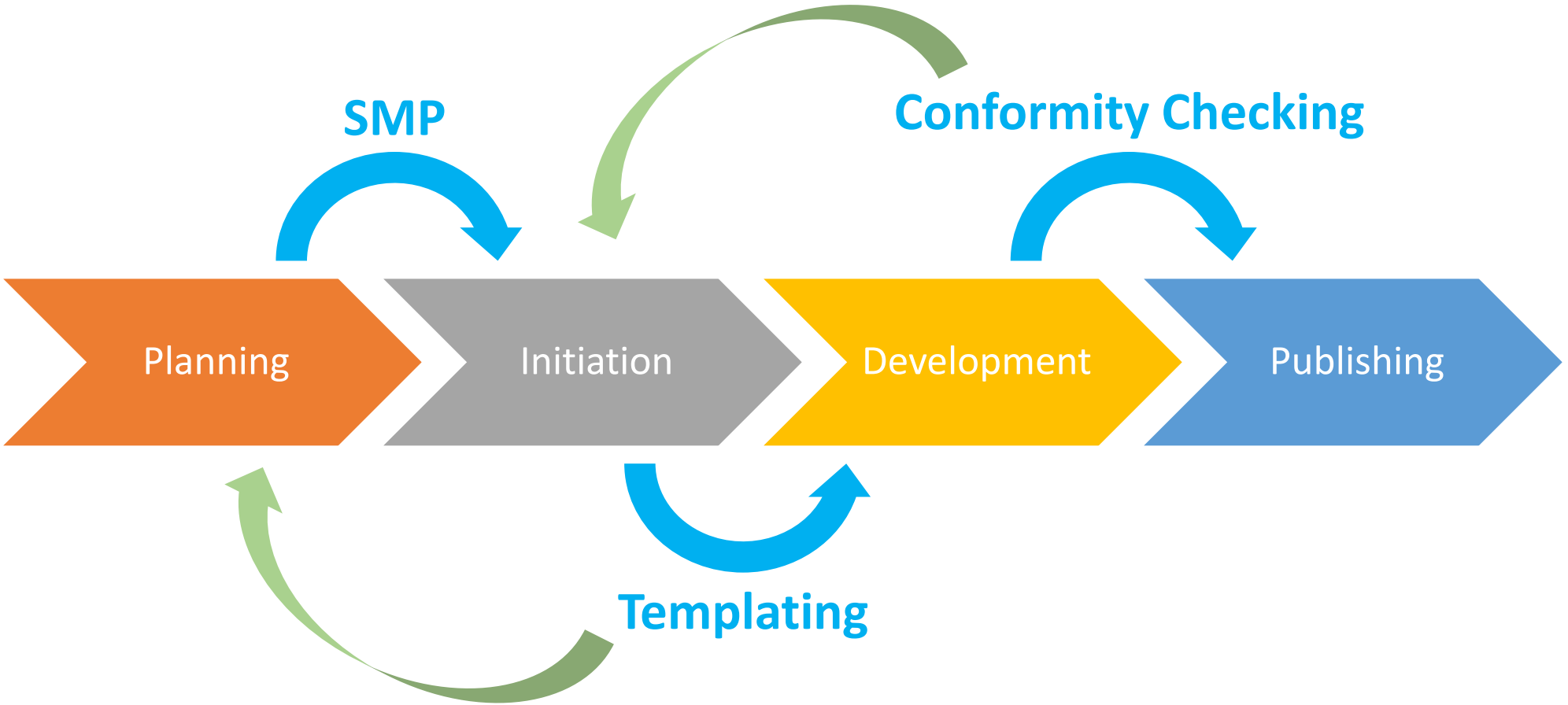
- Project partners are eScience Centre, University of Twente, 4TU.ResearchData, TU Delft, University of Groeningen, KNMI, University of Utrecht, University of Leiden

- The project is funded by the TDCC NES Fund Call 2023

TDCC-NES
Natural & Engineering Sciences

https://tdcc.nl/projects/project-initiatives-nes/tdcc-nes-bottleneck-projects/best-practices-for-sustainable-software/

The vision of the tool development effort is to move forward from templating to **streamlined research software development**



We also aim to **improve sustainability** of tooling and reduce repetitive efforts through **interoperability and reuse**

Illustration by Storyset.com

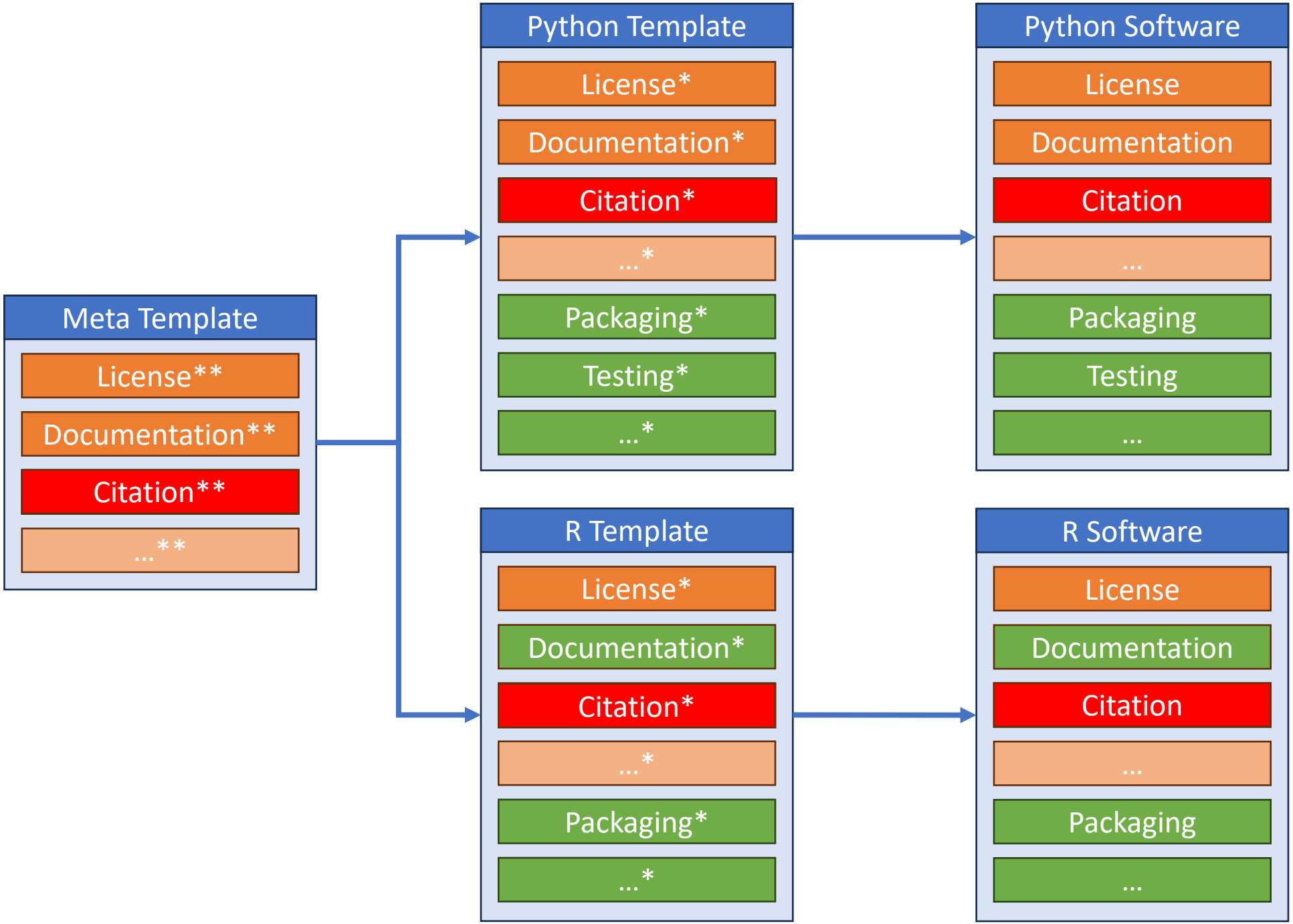
Me and My Clones (Self Portrait), Karl Birrane 2009

MetaTemplate^{alpha}

A modern meta-template for creating research software templates

- Provide code building blocks required for research software templates.
  - Enable standardization of the core building blocks.
  - Enable easy updating of the core building blocks.
- Facilitate development of language-specific templates.
- Facilitate development of institution- and discipline-specific templates.
- Research software created by using meta-template-based templates will be easily updateable.

https://github.com/SS-NES/meta-template


Meta Template
License**
Documentation**
Citation**
...**
Python Template
License*
Documentation*
Citation*
...*
Packaging*
Testing*
...*
Python Software
License
Documentation
Citation
...
Packaging
Testing
...
R Template
License*
Documentation*
Citation*
...*
Packaging*
...*
R Software
License
Documentation
Citation
...
Packaging
...

CodeScanner^{alpha}

A package and command-line tool to check code quality and conformity with research software development best practices

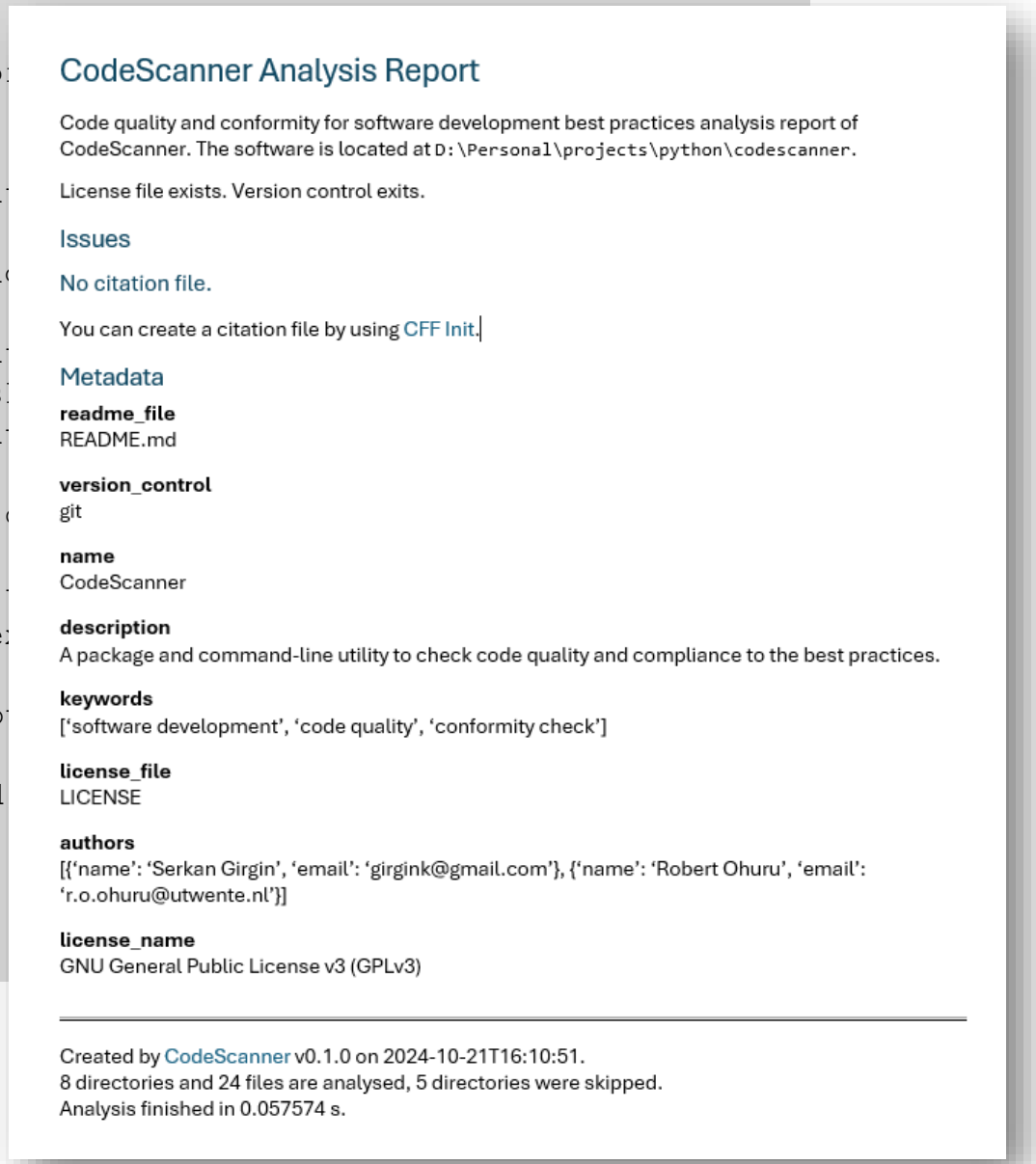
- Extract software metadata from a software code base.
- Identify issues, such as missing or conflicting practices.
- Provide suggestions to solve the issues.
- Enable comparison with reference software metadata (e.g., SMP)*.
- Enable automated corrections via software templating tools*.
- Output analysis reports in various formats.

https://github.com/SS-NES/codescanner

```
Usage: codescanner [OPTIONS] PATH

  Scans the code base, where PATH is the path or URL address o

Options:
  --skip-analyser [citation|code_jupyter|code_python|documenta
                                  List of analysers to skip.
  --skip-aggregator [citation|code|documentation|license|packa
                                  List of aggregators to skip.
  --skip-type [code|license|citation|version_control|documenta
                                  List of analysers types to s
  -r, --reference FILENAME        Path of the reference metada
                                  comparison (e.g. SMP).
  -b, --branch TEXT               Branch or tag of the remote
  -t, --path-type [zip|tar|tgz|tar.gz|git]
                                  Type of the file located at
  -m, --metadata FILENAME         Path to store the metadata e
                                  the code base.
  -o, --output PATH               Path to store the analysis o
  -f, --format [plain|html|json|yaml|md|rst|rtf|docx]
                                  Output format.  [default: md
  -d, --debug                     Enable debug mode.
  -v, --version                   Show the version and exit.
  -h, --help                      Show this message and exit.
```



CodeScanner Analysis Report

Code quality and conformity for software development best practices analysis report of CodeScanner. The software is located at `D:\Personal\projects\python\codescanner`.

License file exists. Version control exits.

Issues

No citation file.

You can create a citation file by using CFF Init.

Metadata

readme_file
README.md

version_control
git

name
CodeScanner

description
A package and command-line utility to check code quality and compliance to the best practices.

keywords
['software development', 'code quality', 'conformity check']

license_file
LICENSE

authors
[{'name': 'Serkan Girgin', 'email': 'girgink@gmail.com'}, {'name': 'Robert Ohuru', 'email': 'r.o.ohuru@utwente.nl'}]

license_name
GNU General Public License v3 (GPLv3)

Created by CodeScanner v0.1.0 on 2024-10-21T16:10:51.
8 directories and 24 files are analysed, 5 directories were skipped.
Analysis finished in 0.057574 s.

Let's demonstrate!

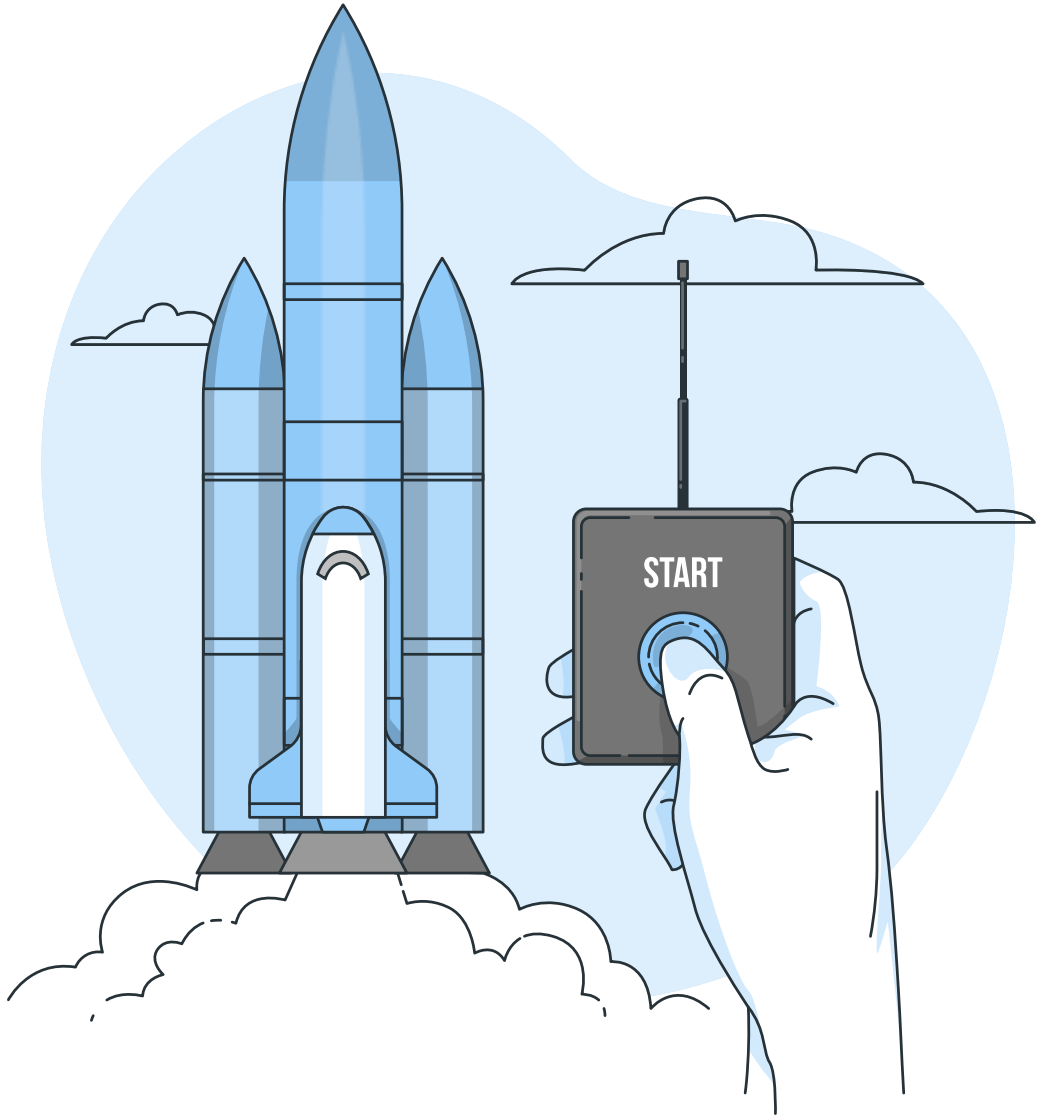


Illustration by Storyset.com

Let's discuss!

Illustration by Storyset.com

Thank you, very interesting, great work!

I would recommend doing an active, hands-on demo, so everyone can work on their laptop and follow along.

Interesting new developments.

A thing I missed is the necessity to form a community around a piece of research software. That is an important other requirement for making research software sustainable. Peter Bra

Very useful tools for people at certain technical capability. This is not most researchers. How to reach people not using CLI?

More info for the Demo part in advance for people join the workshop would be helpful. Then we can do preparations before the session to follow it easier and more efficiently.

IMPRESSIVE! (TEMPLATING I & CODESCANNER)

WE WILL FOLLOW SUIT AND START USING TEMPLATING & THE META-TEMPLATE REPO ... AND CODESCANNER, TOO :-)

Although both tools seem promising they gave me the following impression:

- it is quite challenging and difficult for non researchers to use 3 different tools, learn how to use them and then deploy them for their benefit.
- from my point of view there is not much added value [illegible] as this [illegible] is intended for research software but how

often a particular researcher develops multiple software within the lifecycle of a project so to have the need to "automate the process"

- with the fast advancements and progress of AI era - I think the whole process will be soon replaced by an AI too

Code-checker

Maybe we can make github badges for the Checks

pycharm / code Plugin

We will be happy to be in **contact**!

Dr. Serkan Girgin

Head, Center of Expertise in Big Geodata Science
Assoc. Prof., Department of Geoinformation Processing
Faculty of Geo-information Science and Earth Observation
University of Twente

s.girgin@utwente.nl
https://linkedin.com/in/serkan-girgin/

Manuel Garcia Alvarez

Research Software Engineer
TU Delft Digital Competence Center

m.g.garciaalvarez@tudelft.nl

Dr. Carlos Martinez-Ortiz

Community Manager
Netherlands eScience Center

c.martinez@esciencecenter.nl

https://github.com/SS-NES/