

Research software as an enabler and research output



Anelda van der Walt

Snr eResearch Analyst, eResearch Centre
University of Cape Town
Strategic Lead, RSSE Africa

Peter van Heusden

Snr Bioinformatics Scientist
South African National Bioinformatics Institute
University of the Western Cape
Strategic Lead, RSSE Africa



Think about your last project...



In the chat

Type one piece of software your last project depended on.

Anything counts.

How a global community began

2012 · Oxford

A small group names the role: Research Software Engineer (RSE).

2014 · Cape Town

The first talk on Research Software Engineering in Africa.

2013 · UK

The first national RSE association forms.

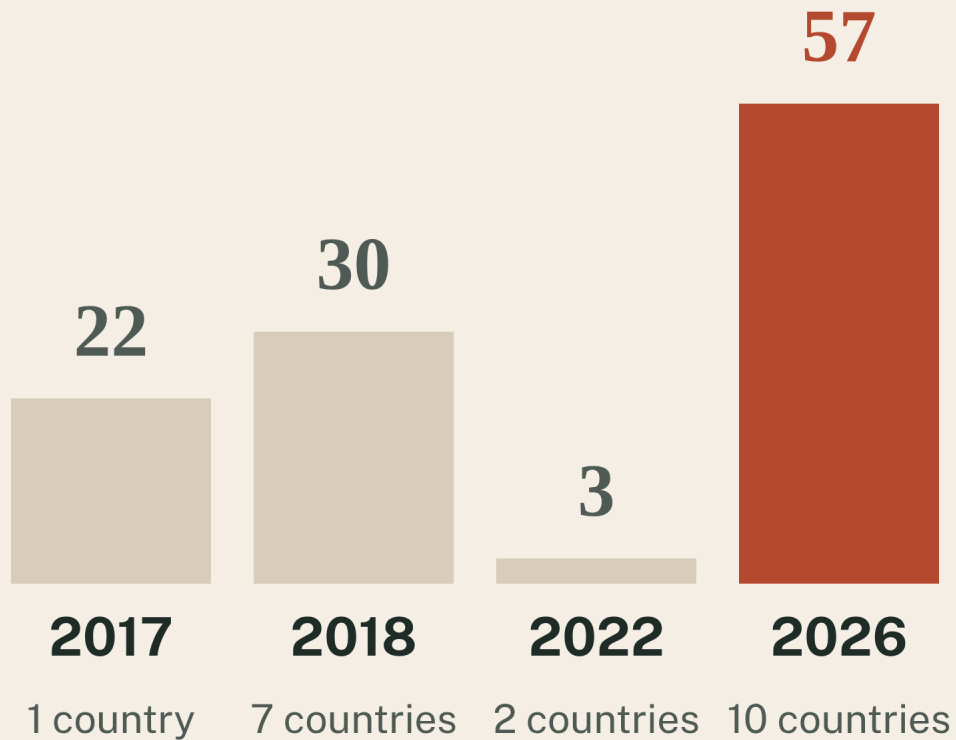
2019 · Africa

RSSE Africa begins, part of a global wave.

A global landscape



African responses to the RSE survey



Data: Hettrick et al. (2026), RSE Survey longitudinal dataset, Zenodo.

This forum
300+
people registered

AI in coding

chat assistants, code completion and coding agents



< — > — > — (

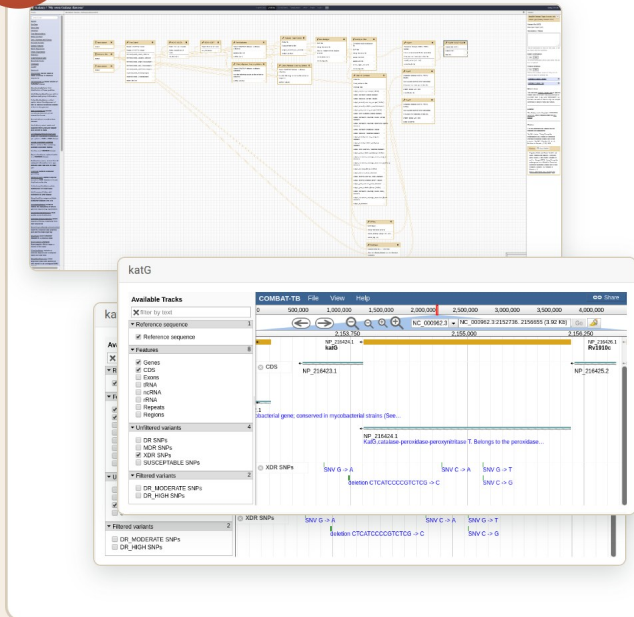
“Research Software includes source code files, algorithms, scripts, computational workflows and executables that were created *during the research process* or *for a research purpose*.”

— Gruenpeter et al. (2021)

One workflow, three projects

1

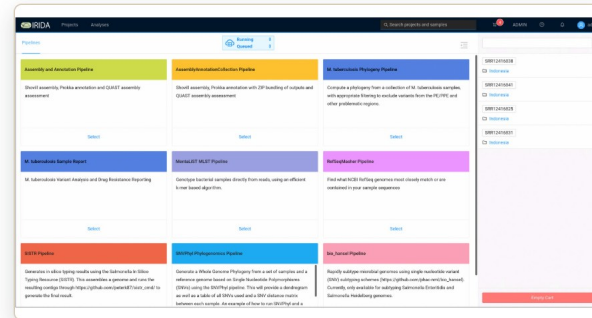
output + enabler



Built in Galaxy for our TB research

2

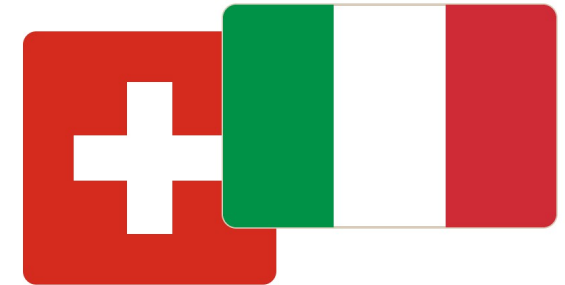
enabler: research



Picked up in the COMBAT-TB Workbench

3

enabler: teaching



Led to a training collaboration with groups in Switzerland and Italy

None of this software stands alone.

It all rests on other people's software.

Research software

Project-specific code

Scripts, notebooks, workflows, ...

Domain-specific tools

GROMACS, MMTK, ...

Scientific infrastructure

BLAS, HDF5, SciPy, ...

Non-scientific infrastructure

gcc, Python, ...

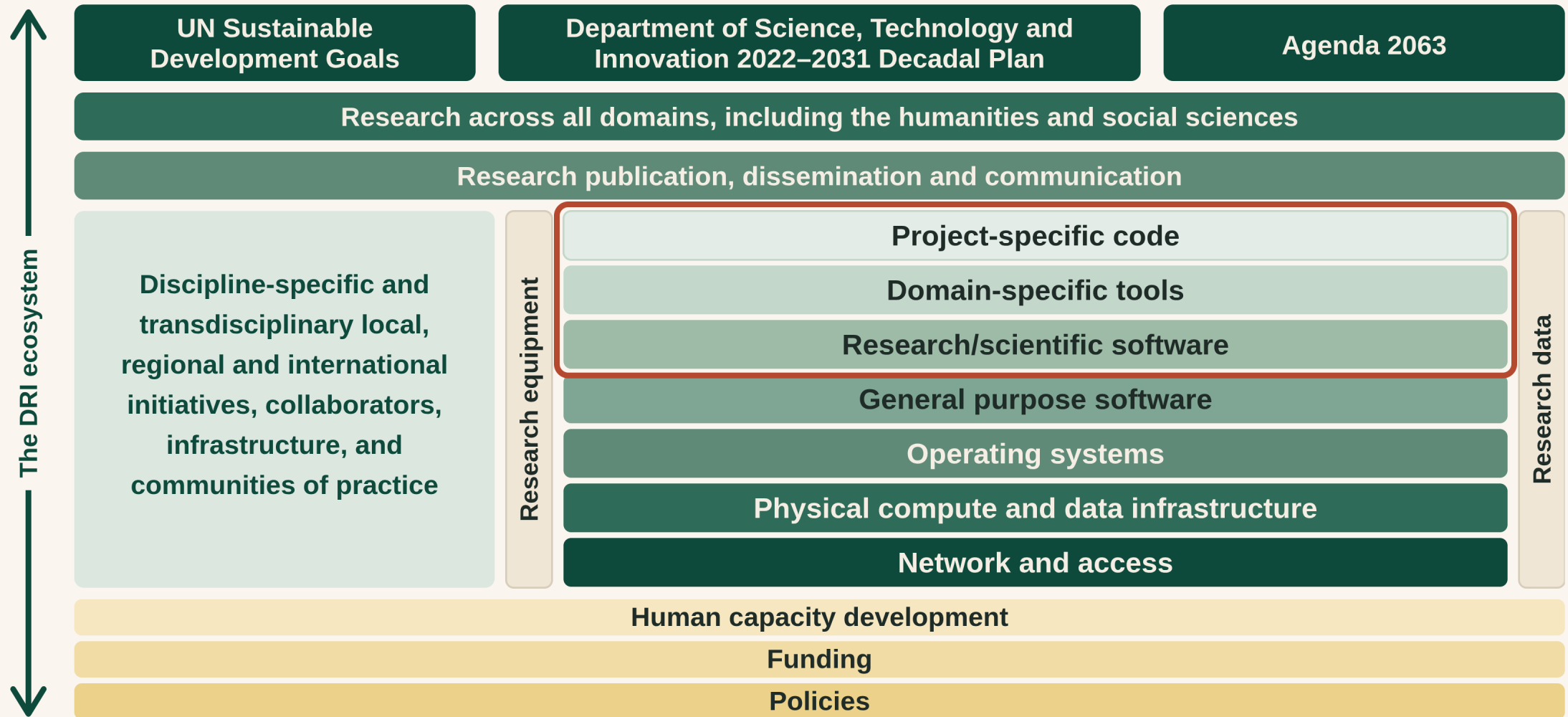
Operating system

GNU/Linux, ...

Hardware

x86 processor, ...

The digital research infrastructure ecosystem



Research software is not only analysis code



research infrastructure software

makes the research environment usable

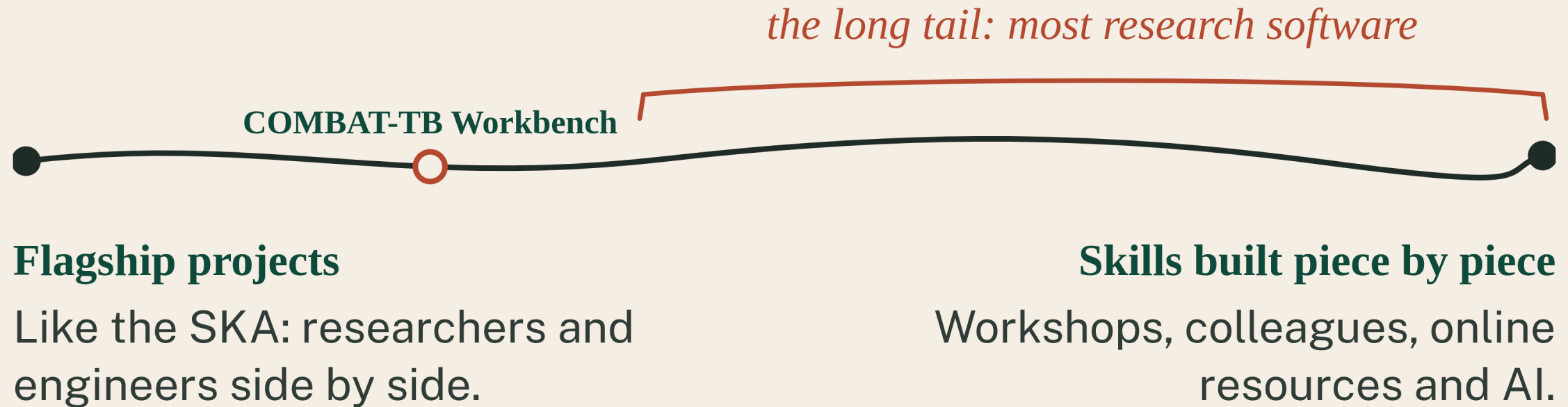
proof-of-concept software

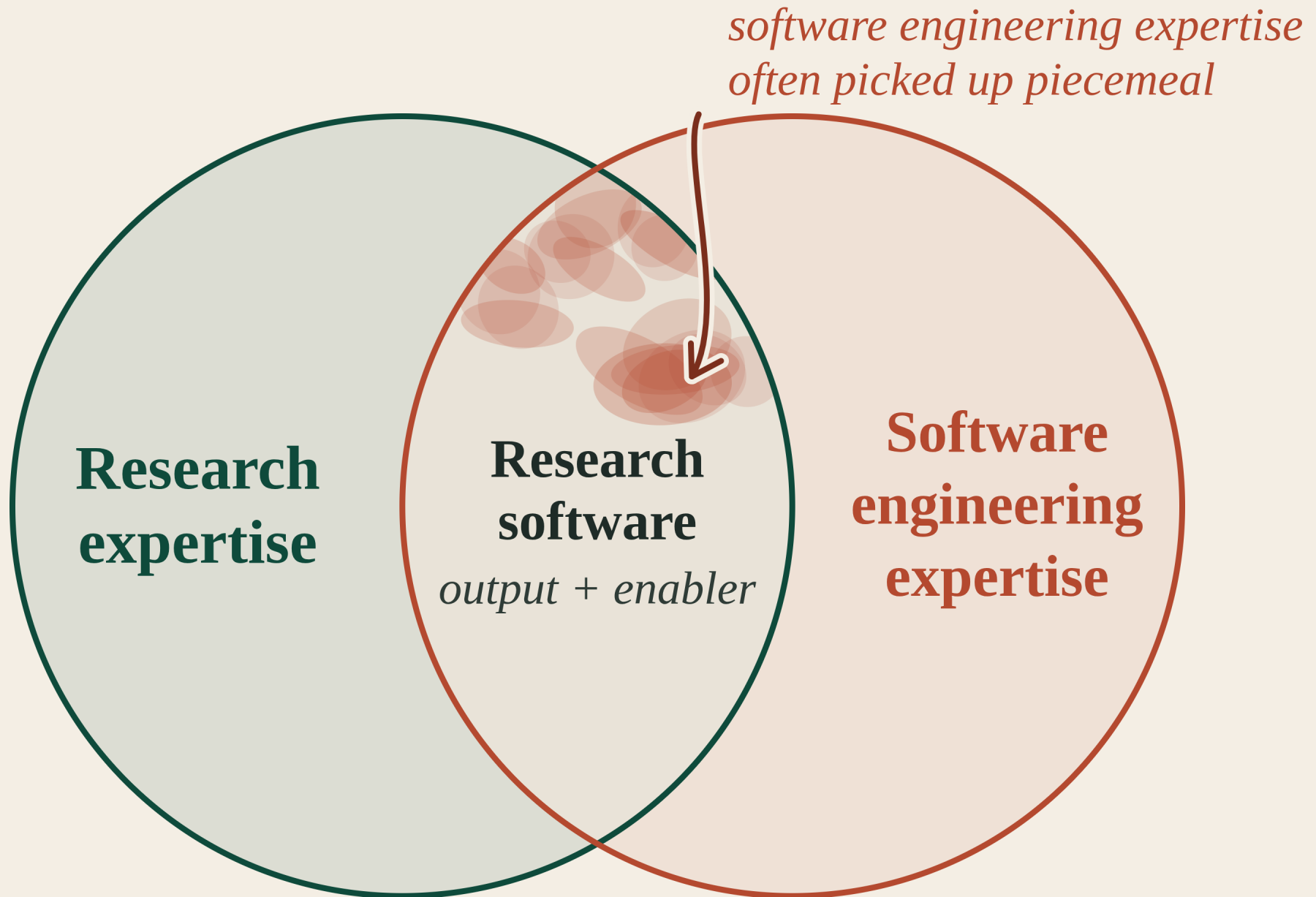
tests ideas, methods and possibilities

modelling, simulation and data analytics

turns data, models and questions into insight

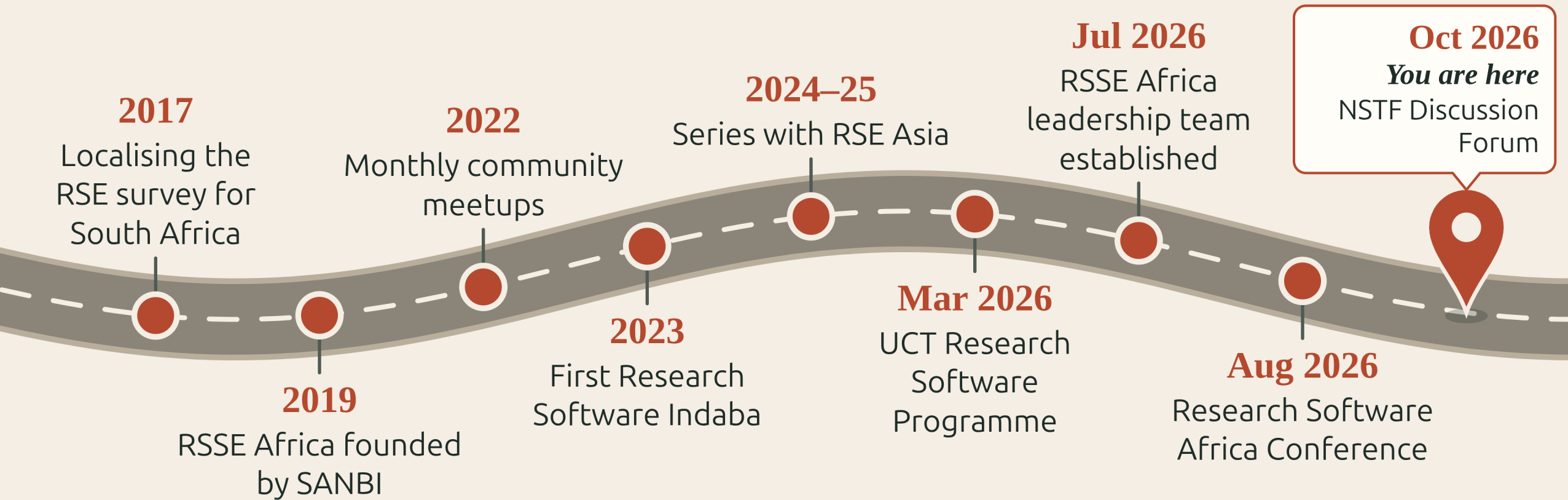
Where (and by whom) research software gets made





< — > — > — (— > — > —] —

“If you build it, they will not necessarily come.”



RSSE Africa Leadership Team



Photo: RSSE Africa (2026)

From left to right:

Juan Steyn

SADiLaR

Anelda van der Walt

UCT eResearch / Talarify

Barbara Ojur

NRF-SARAO

Mthetho Sovara

CHPC, CSIR

Peter van Heusden

SANBI, UWC

Sumir Panji

eLwazi Open Data
Science Platform



*“How and where can we learn to
build good research software?”*

*...and the feeling of working in
isolation.*

It works

Learning to code



**Learning to code is only
the beginning**

Maintainable

Reusable

Readable

Reproducible

Extensible

Portable

Testable

*built with software
engineering practices*



My research question

- R script inherited from a former student

- Package nobody remembers installing

- VPN to reach the cluster

- Library pinned to an old version

- Job-restart script for load-shedding

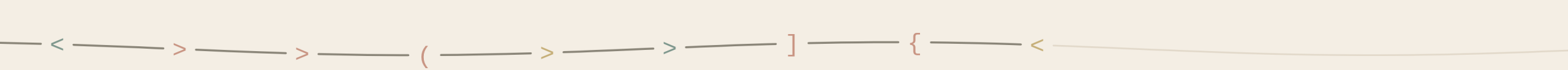
- “Temporary” fix from 2019

- Shared drive nobody can reorganise

- Spreadsheet that feeds the pipeline

- Python environment that only works on one laptop

- Institutional licence (expires in March)



Software rot

Nothing in your code changed. *The world underneath it did.*

COMBAT-TB Workbench

one application

IRIDA and Galaxy

own release schedule

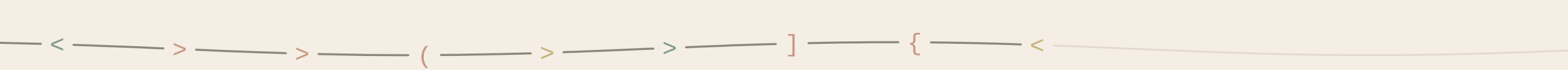
Pathogen genomics tools

own release schedule

Libraries, containers, OS

own release schedule

Hinsen (2019), Dealing with software collapse; van Heusden et al. (2022).



Software rot

Nothing in your code changed. *The world underneath it did.*

output format changed



TB-Profiler changed its output format, and my tool broke.

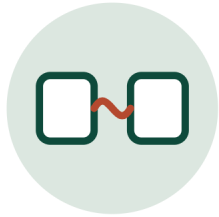
< — > — > — (— > — > —] — { — < —

You can't stop the world underneath your software from changing.

You can design for it.



**Choose
dependencies
deliberately**



**Keep the pieces
loosely connected**



**Record the
environment a
result came from**



**Pin down what
you can**

< > > (> >] { < >

Code written only for yourself can live in *your head*.

```
def f(d, t=0.05):  
    r = []  
    for x in d:  
        if x[3] < t:  
            r.append(x[0])  
    return r
```

It works. Only you know what it does.

Illustrative example.

< — > — > — (— > — > —] — { — < — > —

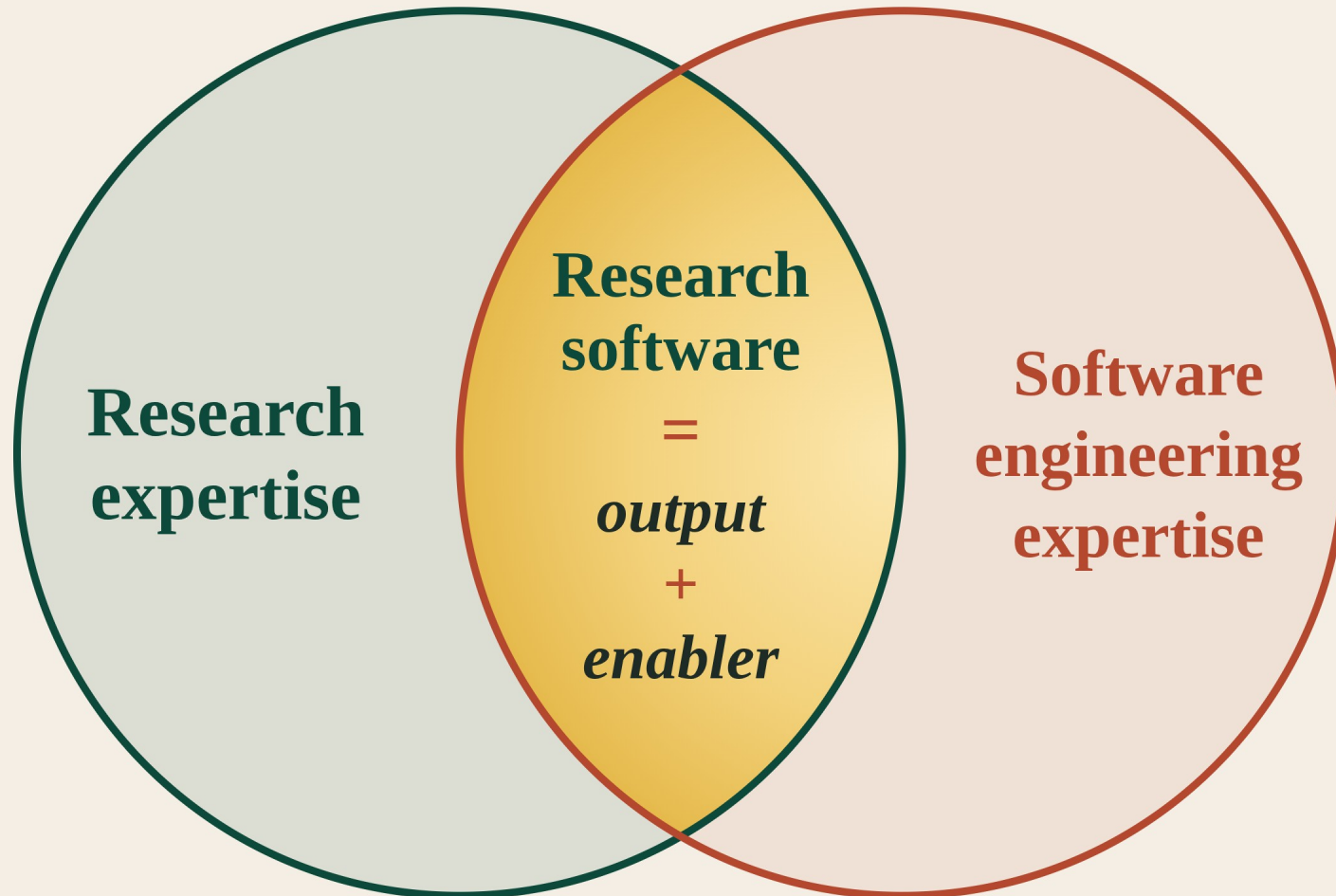
Software that works *beyond you* carries that knowledge with it.

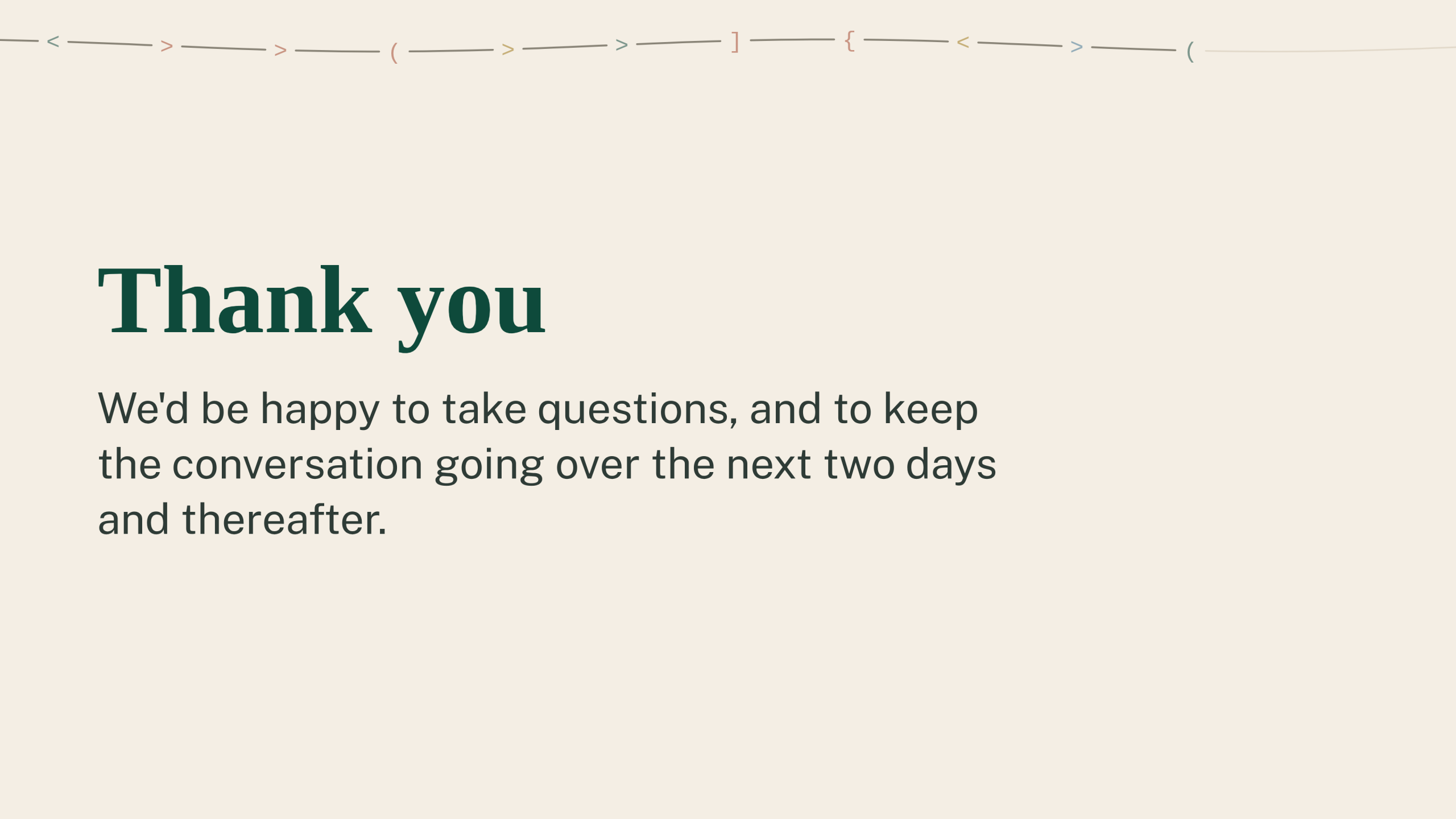
```
def significant_genes(results, cutoff=0.05):  
    """Gene IDs with a p-value below cutoff.  
    results: rows of (gene_id, log_fc, mean_count, p_value)  
    """  
    return [gene_id  
            for gene_id, log_fc, mean_count, p_value in results  
            if p_value < cutoff]
```

An output others can check. An enabler others can build on.

Illustrative example. DLR Software Engineering Guidelines (Schlauch, Meinel and Haupt 2018).

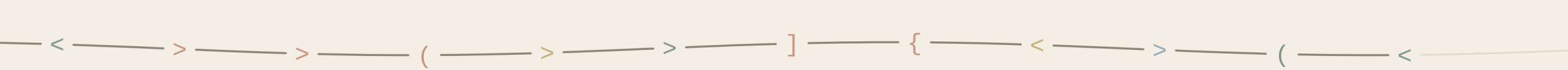
Research software needs both:
good research and good engineering





Thank you

We'd be happy to take questions, and to keep the conversation going over the next two days and thereafter.



Acknowledgements

Co-authors of our 2025 paper: Kim Martin, Sumir Panji, Angelique Trusler and Mattia Vaccari.

Eugene de Beste, Mireille Grobbelaar, Richard Dushime, and the new RSSE Africa leadership team.

Everyone who has answered the survey, joined a community meetup or come to our workshops and in-person events.

The NSTF team for bringing us all together for this discussion forum.

Get in touch

Contact details

1

RSSE Africa

rsse.africa

team@rsse.africa

2

UCT eResearch Centre

uct.ac.za/ereseach

ereseach@uct.ac.za

3

SANBI @ UWC

sanbi.ac.za

pvanheusden@uwc.ac.za

4

Talarify

talarify.co.za

anelda@talarify.co.za



References

- Gruenpeter, M., et al. (2021). Defining research software: A controversial discussion. Zenodo. doi.org/10.5281/zenodo.5504015
- Hasselbring, W., et al. (2024). Toward research software categories. arXiv preprint.
- Hettrick, S. (2016). A not-so-brief history of Research Software Engineers. Software Sustainability Institute.
- Hettrick, S., Bast, R., Crouch, S., Wyatt, C., Philippe, O., et al. (2026). softwaresaved/RSE_survey_longitudinal: RSE Survey longitudinal dataset (v2026.0.0) [Data set]. Zenodo. doi.org/10.5281/zenodo.18609993
- Hinsen, K. (2018). Verifiability in computer-aided research: The role of digital scientific notations at the human-computer interface. *PeerJ Computer Science*, 4, e158. doi.org/10.7717/peerj-cs.158
- Hinsen, K. (2019). Dealing with software collapse. *Computing in Science & Engineering*, 21(3), 104–108. doi.org/10.1109/MCSE.2019.2900945
- Phelan, J. E., et al. (2019). Integrating informatics tools and portable sequencing technology for rapid detection of resistance to anti-tuberculous drugs. *Genome Medicine*, 11, 41. doi.org/10.1186/s13073-019-0650-x
- RSSE Africa. Our History. rsse.africa/history; RSSE Africa announces new Leadership Team (2026). rsse.africa/blog/2026/08/rsse-africa-leadership-team



References, continued

Schlauch, T., Meinel, M., & Haupt, C. (2018). DLR Software Engineering Guidelines. Zenodo. doi.org/10.5281/zenodo.1344612

UCT eResearch (2026). UCT's Research Software Programme innovation strengthens digital research infrastructure. UCT News, 24 March 2026.

van der Walt, A., Martin, K., Panji, S., Trusler, A., Vaccari, M., & van Heusden, P. (2025). Research software: A key (neglected) component of the digital research infrastructure ecosystem. *South African Journal of Science*, 121(11/12). doi.org/10.17159/sajs.2025/21437

van der Walt, A., & van Heusden, P. (2026). The only way to play the game is to change the rules: the RSSE Africa journey. Research Software Africa. doi.org/10.5281/zenodo.22255120

van Heusden, P., Mashologu, Z., Lose, T., Warren, R., & Christoffels, A. (2022). The COMBAT-TB Workbench: Making powerful *Mycobacterium tuberculosis* bioinformatics accessible. *mSphere*. doi.org/10.1128/msphere.00991-21

Images

Illustrations of the researcher, the iceberg and the tangle: generated with ChatGPT from original prompts. Labels on the tangle are composite, not from any one project.

Photo of the RSSE Africa Leadership Team: RSSE Africa (2026).

Screenshots: the *M. tuberculosis* workflow in Galaxy, and the COMBAT-TB Workbench.

Figures adapted from van der Walt et al. (2025) and Hinsén (2018), both CC BY 4.0.