The natESM Journey for Improving Software Quality in Earth System Modelling

Wilton Jaciel Loch
natESM | German Climate Computing Center (DKRZ)

The state of affairs

Earth
System
Modelling

HPC
Scientific

Software

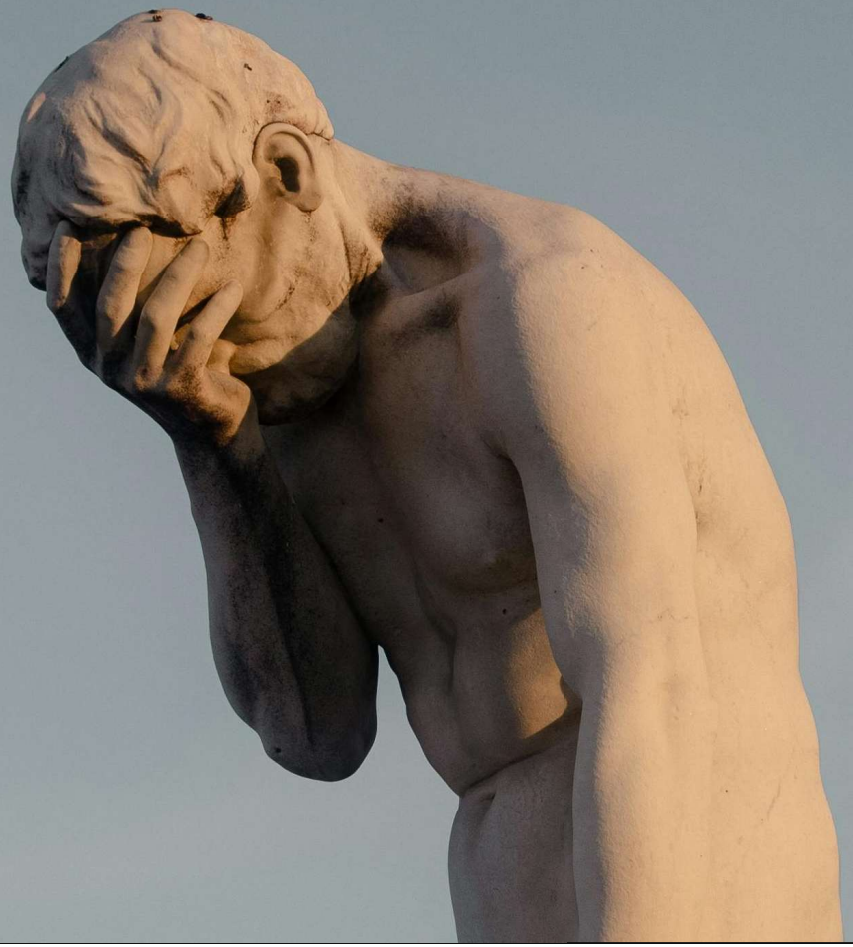
Like Atlas, HPC software is not feeling well

Organic development

No trained software engineers

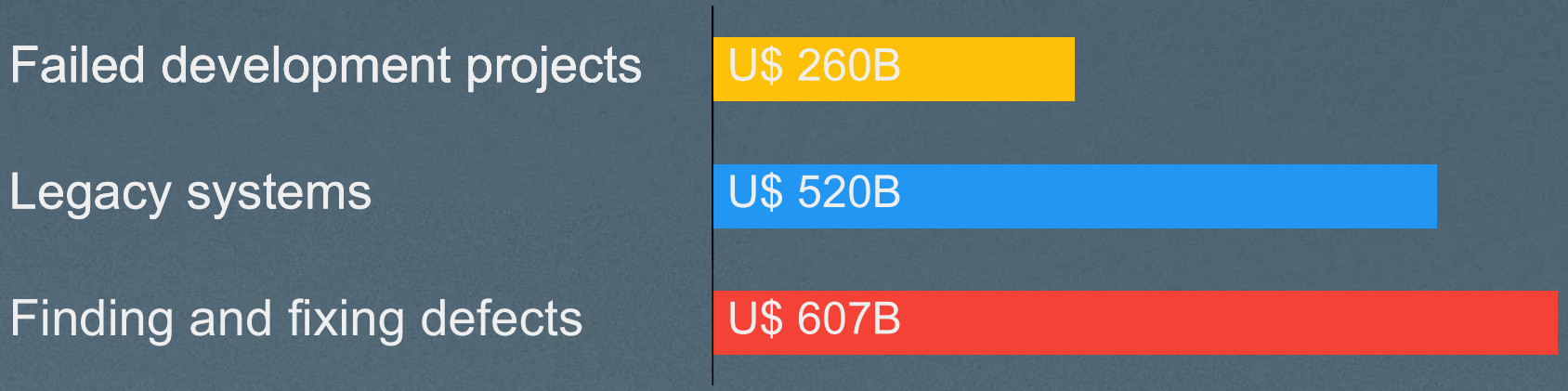
Understaffed institutions

Software seen only as a tool



Poor software quality is a huge waste of resources

Software quality costed the US economy **U$ 2.41 Trillion** in 2022[1]



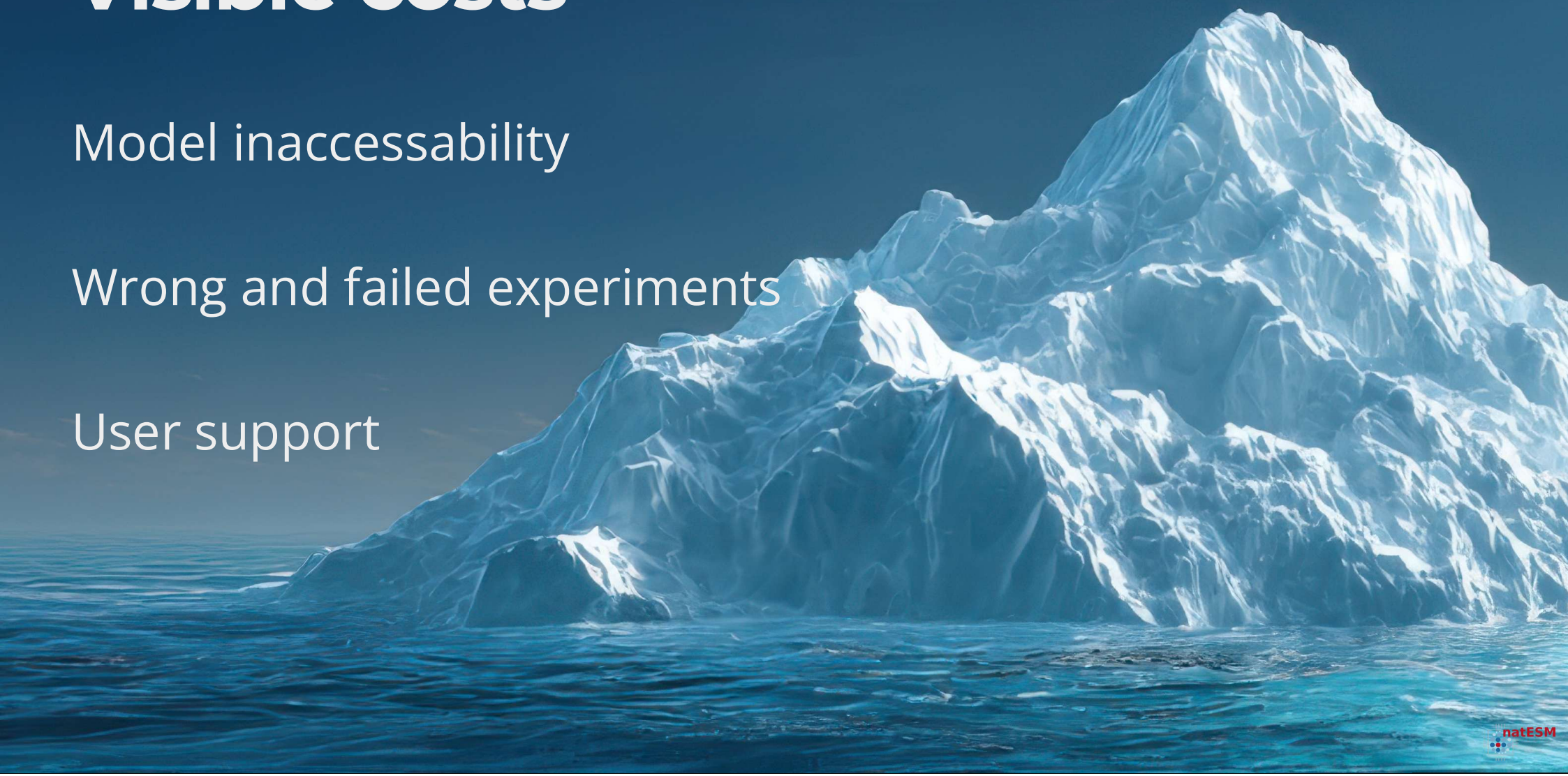
The costs

Visible costs

Model inaccessability

Wrong and failed experiments

User support



Hidden costs

Fixing bugs

Complex code

Technical debt

Longer onboarding

Excessive systems costs

Lost research opportunities

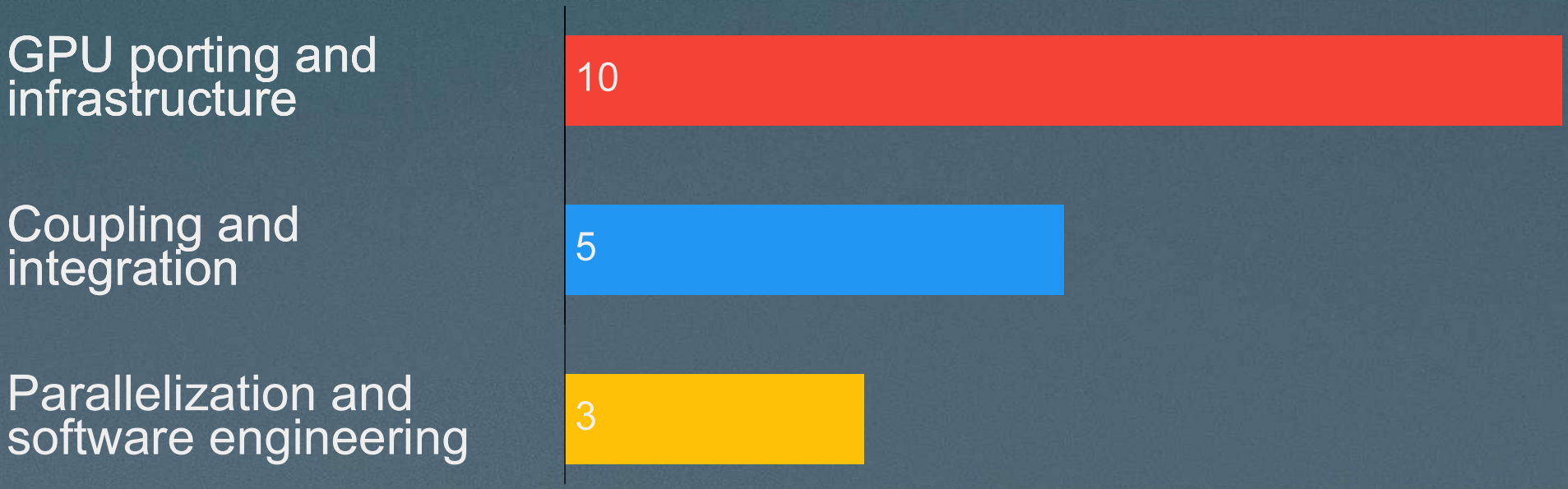
natESM comes to help

Improving **technical infrastructure to serve Science**

6 months projects (sprints)

Well-defined **goals and timeline**

Improvements done by natESM

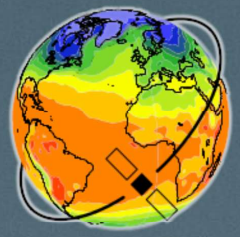


natESM has improved many ESM models

ICON

Modular Earth Submodel System

FESOM2



ESMValTool

Earth System Model Evaluation Tool

palm

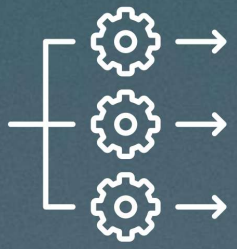
PISM

PARALLEL ICE SHEET MODEL

And more...

The case of CLEO

Cloud microphysics superdroplet model



Parallelization
with MPI

Coupling
with ICON

Versionining
Releases
Git workflows
CI/CD

CLEO - CI expansion

Semantic versioning

Releases 141

v0.65.1 Latest
on Feb 18

+ 140 releases

Precommit

```
Check YAML file formatting.....Passed
No large Files.....Passed
Check Valid Python.....Passed
End of File Fix.....Passed
No direct commit to main.....Passed
Trim trailing whitespaces.....Passed
ruff.....Passed
ruff-format.....Passed
cpplint.....Passed
```

Conventional commits

Commits on Jan 21, 2026

refactor: add print after timestepping
yoctoyotta1024 committed on Jan 21 · ✓ 25 / 28

fix: missing pip requirement
yoctoyotta1024 committed on Jan 21 · ✗ 23 / 28

fix: prevent bad directories at root of installations
yoctoyotta1024 committed on Jan 21 · ✓ 25 / 28

refactor: include making directory in instructions
yoctoyotta1024 committed on Jan 21 · ✓ 25 / 28

docs: update pre-commit hooks install
yoctoyotta1024 committed on Jan 21 · ✓ 25 / 28

refactor: path2kokkostools and kokkos_tools_lib are different
yoctoyotta1024 committed on Jan 21 · ✓ 25 / 28

feat: new helper scripts to install kokkos tools
yoctoyotta1024 committed on Jan 21 · ✓ 25 / 28

CLEO - Release improvements

Automatic releases

Automatic changelog

Linear Git history

Documentation generation

CI builds for all examples

Serial and parallel CI runs

Changing software and workflows can be challenging

Convincing about utility

Old habits are hard to change

Lack of enforcing turns into lack of use

Delayed merging can require large reworks

You can only improve what you can measure

We are creating a simple framework to list software quality deficiencies

Evaluation produces a **score** for the model

List of actionable items that can be worked on during a sprint

How the framework works

Collection of bad points of models

Reproducible and fairly objective

Can be used to track the model evolution

Conclusion

"Over a 25-year life expectancy of a large software system, almost **fifty cents out of every dollar** will go to finding and fixing bugs."[1]

References

1 - Consortium for Information & Software Quality (CISQ). The Cost of Poor Software Quality in the U.S.: A 2022 Report. CISQ, 2022, https://www.it-cisq.org/the-cost-of-poor-quality-software-in-the-us-a-2022-report/

2 - Krasner, Herb. “The cost of poor quality software in the us: A 2018 report.” Consortium for IT Software Quality, Tech. Rep 10 (2018): 8.