Technische Universität Braunschweig

MUSEN

SURESOFT:

Towards Sustainable Research Software

Technische Universität Braunschweig

Suresoft
SUSTAINABLE RESEARCH SOFTWARE

Who are we?

Dr. Jan Linxweiler
j.linxweiler@tu-braunschweig.de

Sören Peters
soe.peters@tu-braunschweig.de

Sven Marcus
sven.marcus@tu-braunschweig.de

Who are we?

18 People from 7 Institutes and Facilities

INSTITUT FÜR AKUSTIK

Institute of Operating Systems and Computer Networks

fN Institut für Nachrichtentechnik

iRMB

Institut für Physikalische und Theoretische Chemie

Technische Universität Braunschweig

University Library & Gauß-IT-Zentrum

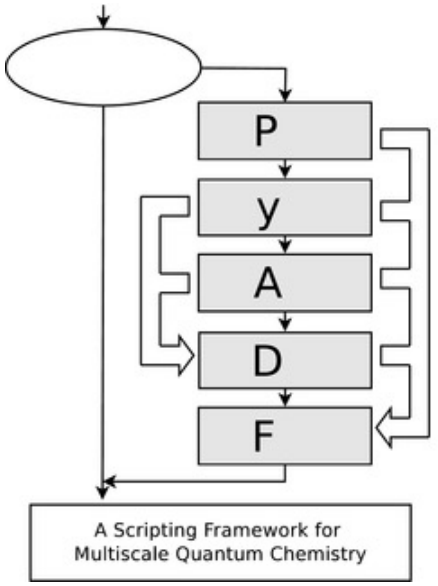
https://doi.org/10.5281/zenodo.10002995

SURESOFT Projects

INSTITUT FÜR AKUSTIK
TECH elPaSo

VIRTUAL FLUIDS

Themis



https://doi.org/10.5281/zenodo.10002995

HELMHOLTZ
Open Science

NFDi4ing

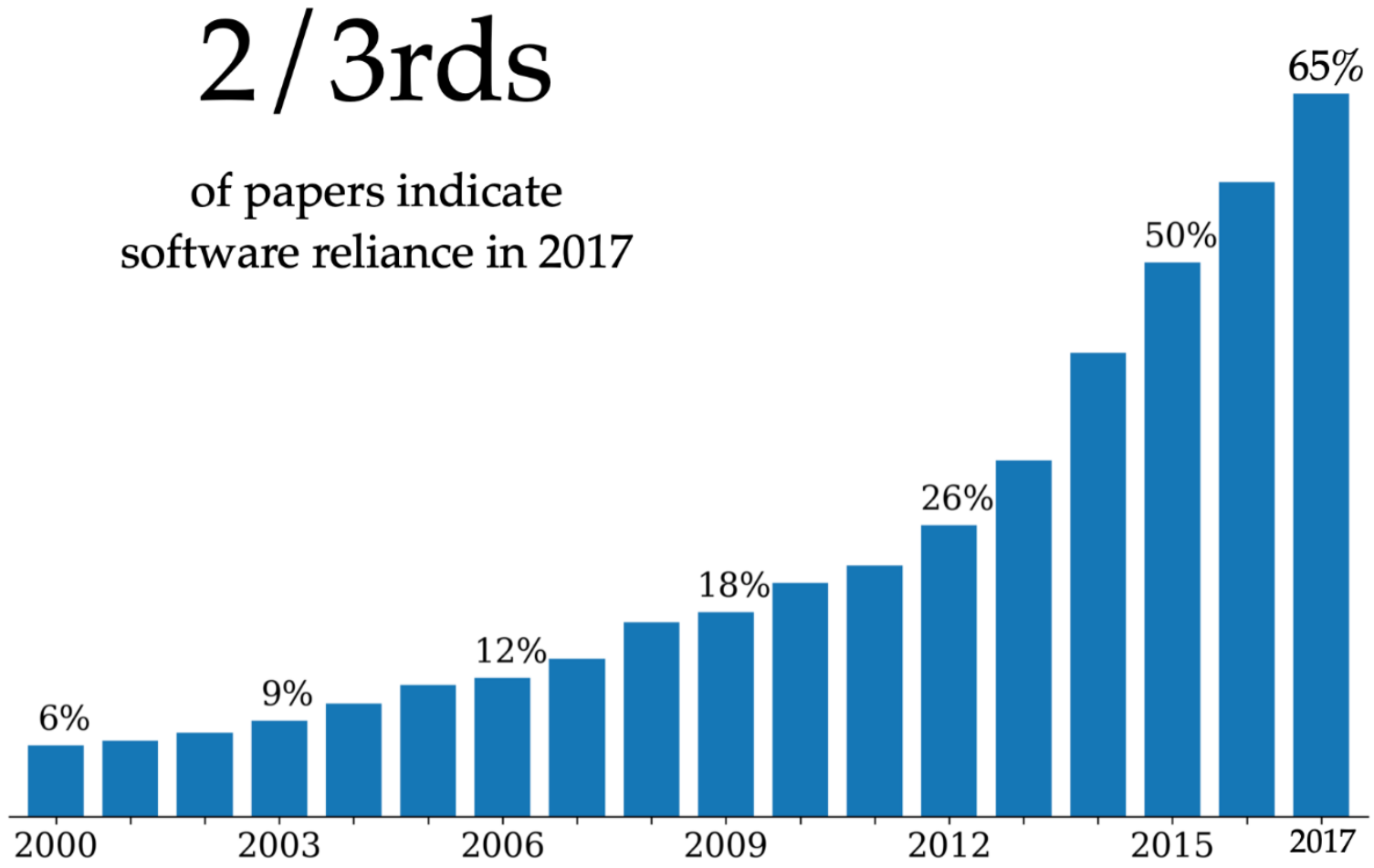
HeFDI

HERMES
Helmholtz Rich Metadata
Software Publication

HZDR
HELMHOLTZ ZENTRUM
DRESDEN ROSSENDORF

https://doi.org/10.5281/zenodo.10002995

Publications relying on software



https://bit.ly/37XEJ2u

Use of research software

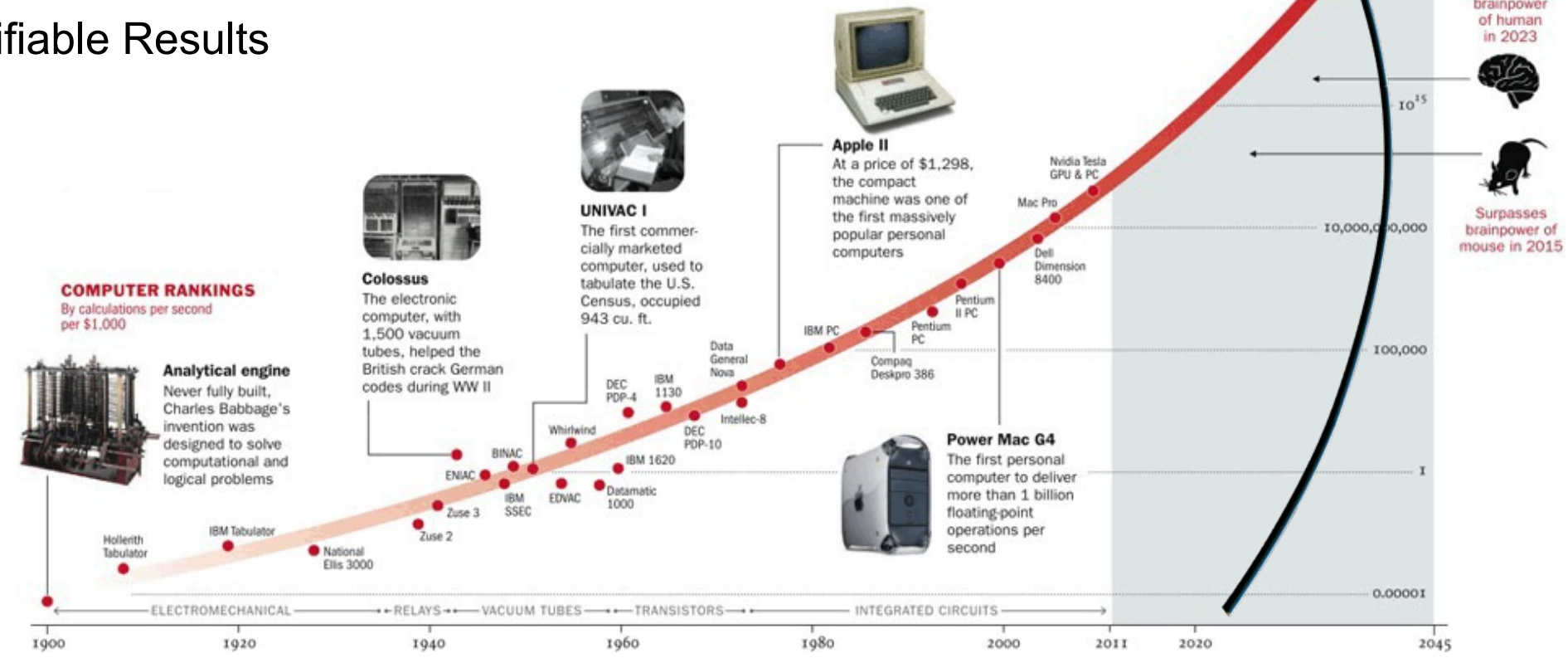
- 92% of academics use research software
- 69% say that their research would not be practical without it
- 56% develop their own software

https://bit.ly/2zZPhSa

matlab
microsoft
excel
python
R
spss
sas
git
stata
eclipse
endnote
maven
gap
cuda
aws
linux
ccp4
slack
latex
fortran
labview
nvivo
origin
sage
jupyter
gimp
docker
ape
trello
visual studio
access
imagej
graphpad prism
mathematica
galaxy
travis
java
bash
intellij
jenkins
gatk
zotero
chemdraw
gnuplot
paraview
solidworks
ansys
matplotlib
atlas.ti
igor
vasp
tophat
ruby
prism
perl
octave
django
castep
microsoft office
sql
dropbox
qualtrics
sublime text
meep
fiji
weka
postgresql
antconc
mendeley
skype
minitab
word
jira
wikis
sigmaplot
samtools
emacs
gfortran
mlwin
cufflinks
coreldraw
gromacs
pymol
php
atom
mercury
visio
coot
arcgis
haskell
bwa
petsc
maple
shell
mpi
amos
bowtie
vim
mercurial
mysql
qt
gaussian
fenics
root
qgis
vba
bitbucket
google docs
arcmap
subversion
openrefine
javascript
blast
trac
hdf5
cbmc
pandas
vmware
powerpoint
server
editors
chimera
version control
gitlab
plink
flowjo
praat
abaqus
pycharm
gromacs
qiime
fsl
dl_poly
opus
spm
nwchem
bioedit
adobe photoshop
xcode
idris
lasergene
amber
compilers
gulp
tophat
solidworks
gmin
dmacrys
sql
dropbox
jenkins
intellij
word
fedit
gams
origin
jupyter
gfortran
emacs
comsol
mercury
ide
auto
trello
mpi
amos
vba
blast
cbmc
galaxy
vmware
bioconductor
castep
microsoft office
fortran compilers
minitab
microsoft word
sage
origin
jupyter
gfortran
emacs
samtools
igv
text editor
velocity
gimp
nvivo
mercury
ide
auto
trello
visual studio
haskell
shell
mpi
amos
vba
bitbucket
blast
cbmc
galaxy
vmware
mathematica
prism
graphpad
javascript
openrefine
subversion
hdf5
qgis
gsea
root
mysql
idl
atk
c++
emake
bwa
petsc
maple
coot
arcgis
pymol
access
ape
atom
php
docker
labview
coreldraw
cufflinks
fortran

Growing demands on scientific software

- Increasing Complexity (e.g. multi physics, multiple groups)
- Longer Life Span (base your work on the work of others)
- Reproducible and Verifiable Results



https://doi.org/10.5281/zenodo.10002995

Suresoft
SUSTAINABLE RESEARCH SOFTWARE

Motivation

Researchers

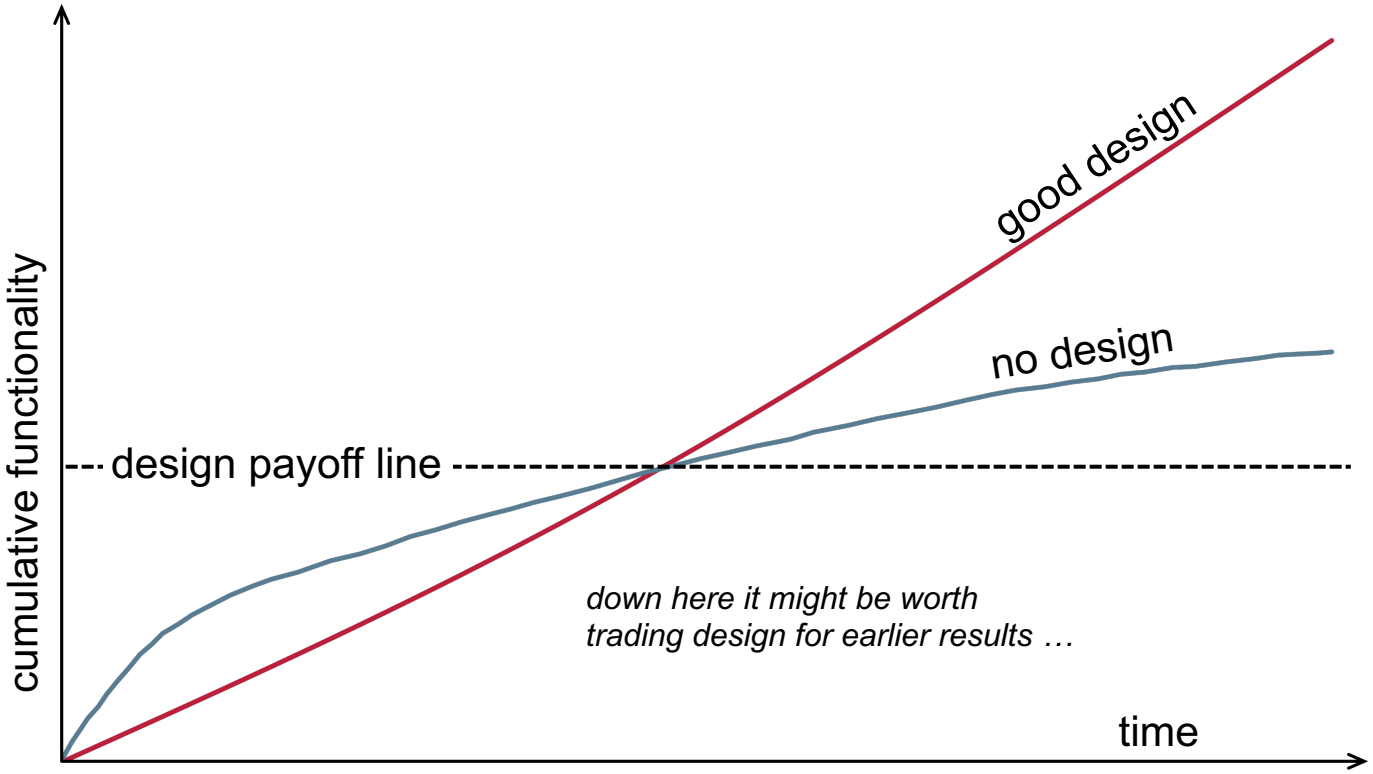
- often lack knowledge about the principles and practices of the software engineering discipline [5, 4]
- don’t gain a reputation for developing software
- are pressed to publish results as fast as possible [2]

https://doi.org/10.5281/zenodo.10002995

Productivity Crisis

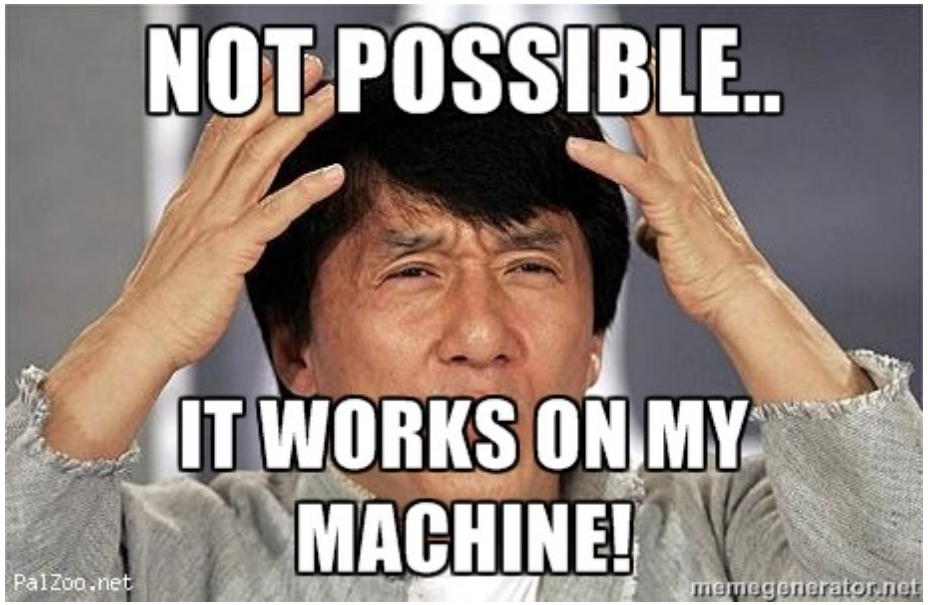
- floating point performance is constantly rising
- time-to-solution is inceasing
- scientists spend 50% of the time finding bugs

[P. Prabhu, A Survey of the Practice of Computational Science, 2011]



Design Stamina Hypothesis
https://bit.ly/2A64CAR

Consequences



Common problems of research software

- low code quality
- neither published nor documented
- depends on a specific runtime environment (e.g third party libraries), which may not be available to other researchers

https://doi.org/10.5281/zenodo.10002995

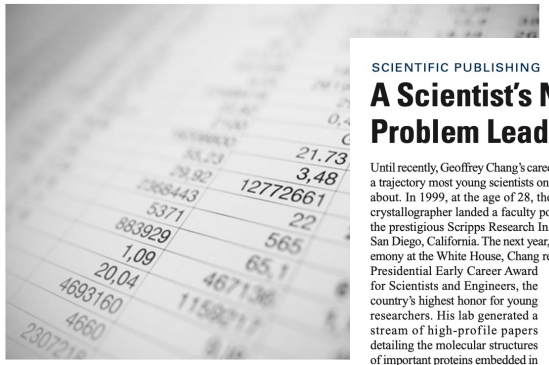
Credibility Crisis

Questionable reliability, accuracy, reproducibility and verifiability of the results …

■ 18 April 2013, 12:31 CEST

FAQ: Reinhart, Rogoff, and the Excel Error That Changed History

By Peter Coy



PHOTOGRAPH BY GREGOR SCHUSTER

Papers in economics 'not reproducible'

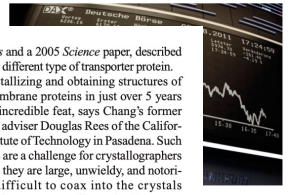
Fears that discipline is particularly susceptible to statistical 'hacking' of data to gain a positive result

October 21, 2015

By David Matthews

Twitter: @DavidMJourno

At least half of papers in economics are



· cent of papers

t.

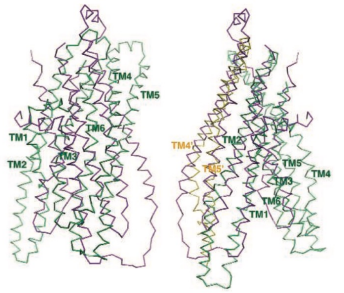
ludes.

SCIENTIFIC PUBLISHING

A Scientist's Nightmare: Software Problem Leads to Five Retractions

Until recently, Geoffrey Chang's career was on a trajectory most young scientists only dream about. In 1999, at the age of 28, the protein crystallographer landed a faculty position at the prestigious Scripps Research Institute in San Diego, California. The next year, in a ceremony at the White House, Chang received a Presidential Early Career Award for Scientists and Engineers, the country's highest honor for young researchers. His lab generated a stream of high-profile papers detailing the molecular structures of important proteins embedded in cell membranes.

Then the dream turned into a nightmare. In September, Swiss researchers published a paper in *Nature* that cast serious doubt on a protein structure Chang's group had described in a 2001 *Science* paper. When he investigated, Chang was horrified to discover that a homemade data-analysis program had flipped two columns of data, inverting the electron-density map from which his team had derived the final protein structure. Unfortunately, his group had used the program to analyze data for 2001 *Science* paper, which described the structure of a protein called MsbA, isolated from the bacterium *Escherichia coli*. MsbA belongs to a huge and ancient family of molecules that use energy from adenosine triphosphate to transport molecules across cell membranes. These so-called ABC transporters perform many

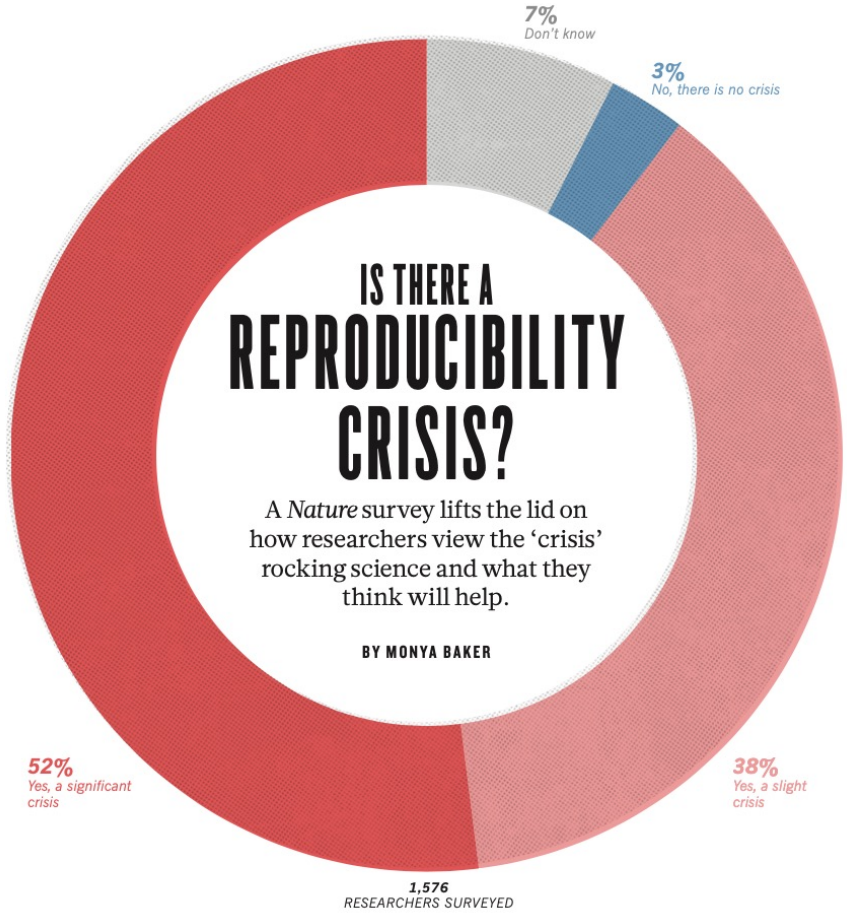


Flipping fiasco. The structures of MsbA (purple) and Sav1866 (green) overlap little (*left*) until MsbA is inverted (*right*).

Sciences and a 2005 *Science* paper, described EmrE, a different type of transporter protein.

Crystallizing and obtaining structures of five membrane proteins in just over 5 years was an incredible feat, says Chang's former postdoc adviser Douglas Rees of the California Institute of Technology in Pasadena. Such proteins are a challenge for crystallographers because they are large, unwieldy, and notoriously difficult to coax into the crystals needed for x-ray crystallography. Rees says determination was at the root of Chang's success: "He has an incredible drive and work ethic. He really pushed the field in the sense of getting things to crystallize that no one else had been able to do." Chang's data are good, Rees says, but the faulty software threw everything off.

Ironically, another former postdoc in Rees's lab, Kaspar Locher, exposed the mistake. In the 14 September issue of *Nature*, Locher, now at the Swiss Federal Institute of Technology in Zurich, described the structure of an ABC transporter called Sav1866 from *Staphylococcus aureus*. The structure was dramatically—and unexpectedly—different from that of MsbA. After pulling up Sav1866 and Chang's MsbA from *S. typhimurium* on a computer screen, Locher says he realized in minutes that the MsbA structure was inverted. Interpreting the "hand" of a molecule is always a challenge for crystallographers,



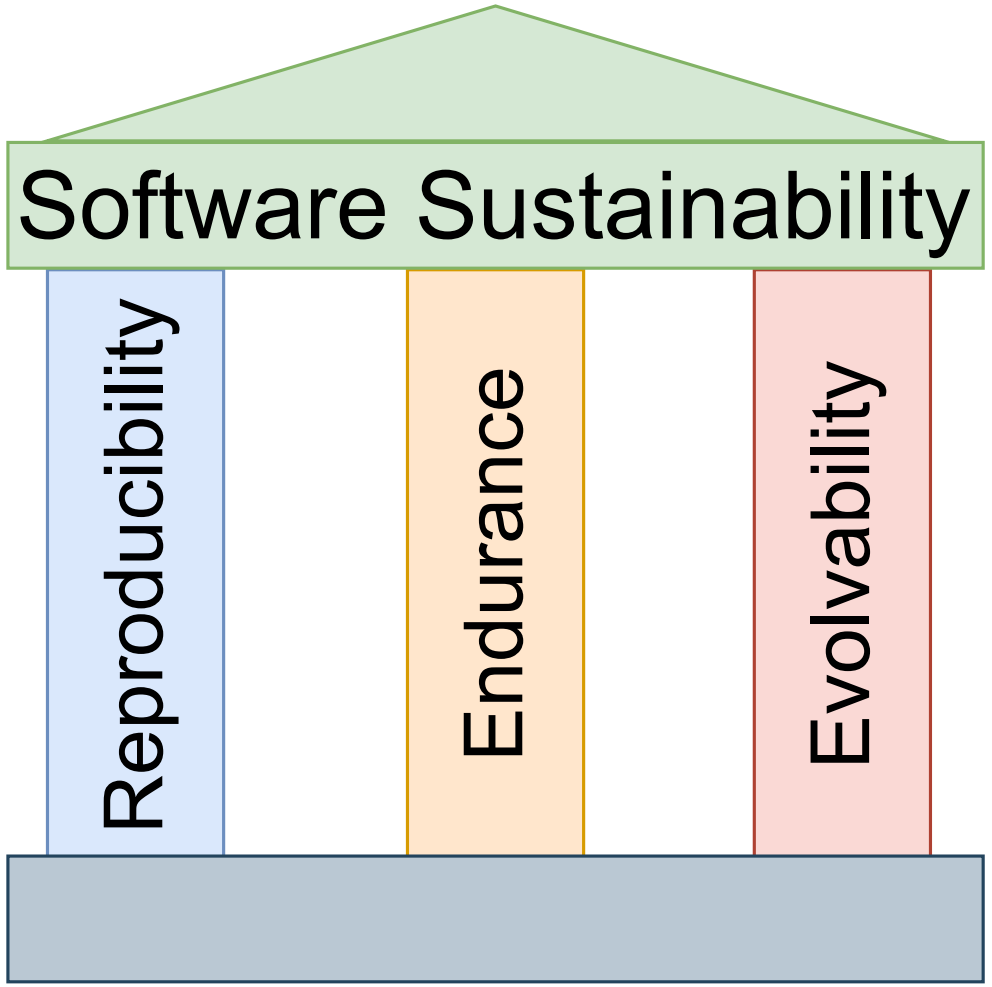
https://go.nature.com/2DgtDKR

https://doi.org/10.5281/zenodo.10002995

Approach

https://doi.org/10.5281/zenodo.10002995

Software sustainability



https://doi.org/10.5281/zenodo.10002995

SURESOFT Approach for Sustainable Software

Education

- Documentation
- Software Engineering Principles
- Testing

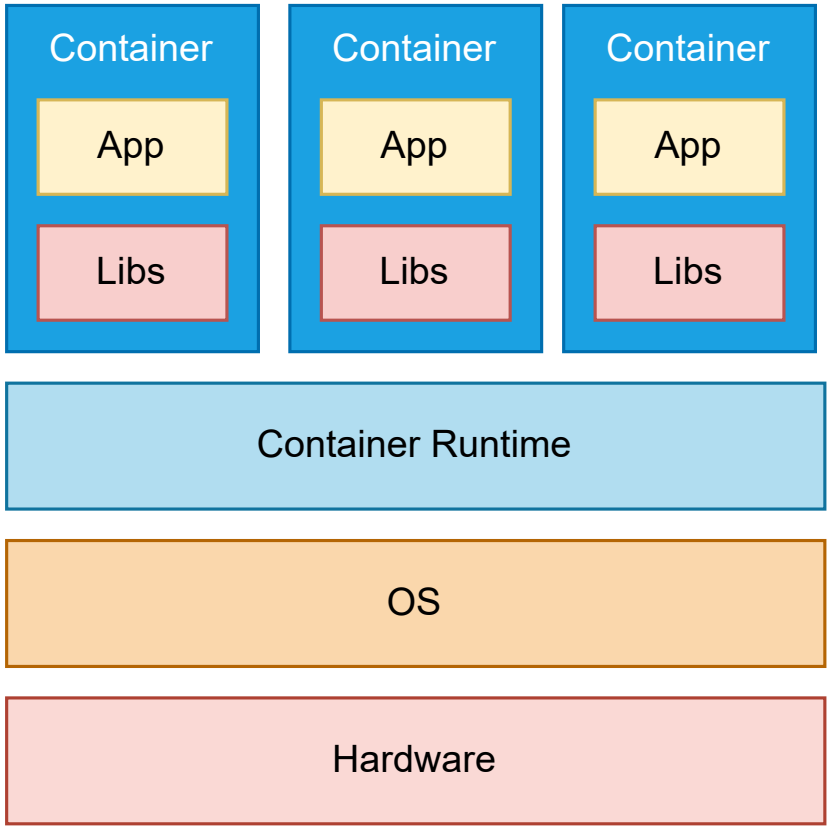
Infrastructure & Methods

- Version Control
- Archiving & Publication (LeoPARD)
- CI & Automated Testing (docker)
- Virtualization (docker)
- Issue Reporting
- Installation & Deployment (LeoPARD, docker)

https://doi.org/10.5281/zenodo.10002995

Container technologies

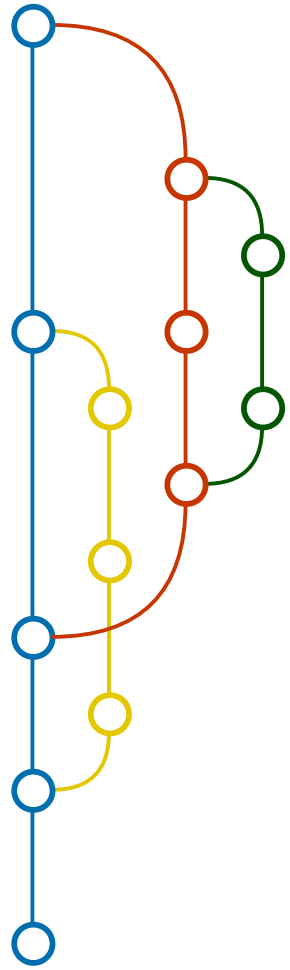
- Docker in CI, Singularity in HPC
- Encapsulate entire runtime environment, including dependencies
- Easy to share and use Ensures reproducibility
- Scripted environment provides basic documentation



https://doi.org/10.5281/zenodo.10002995

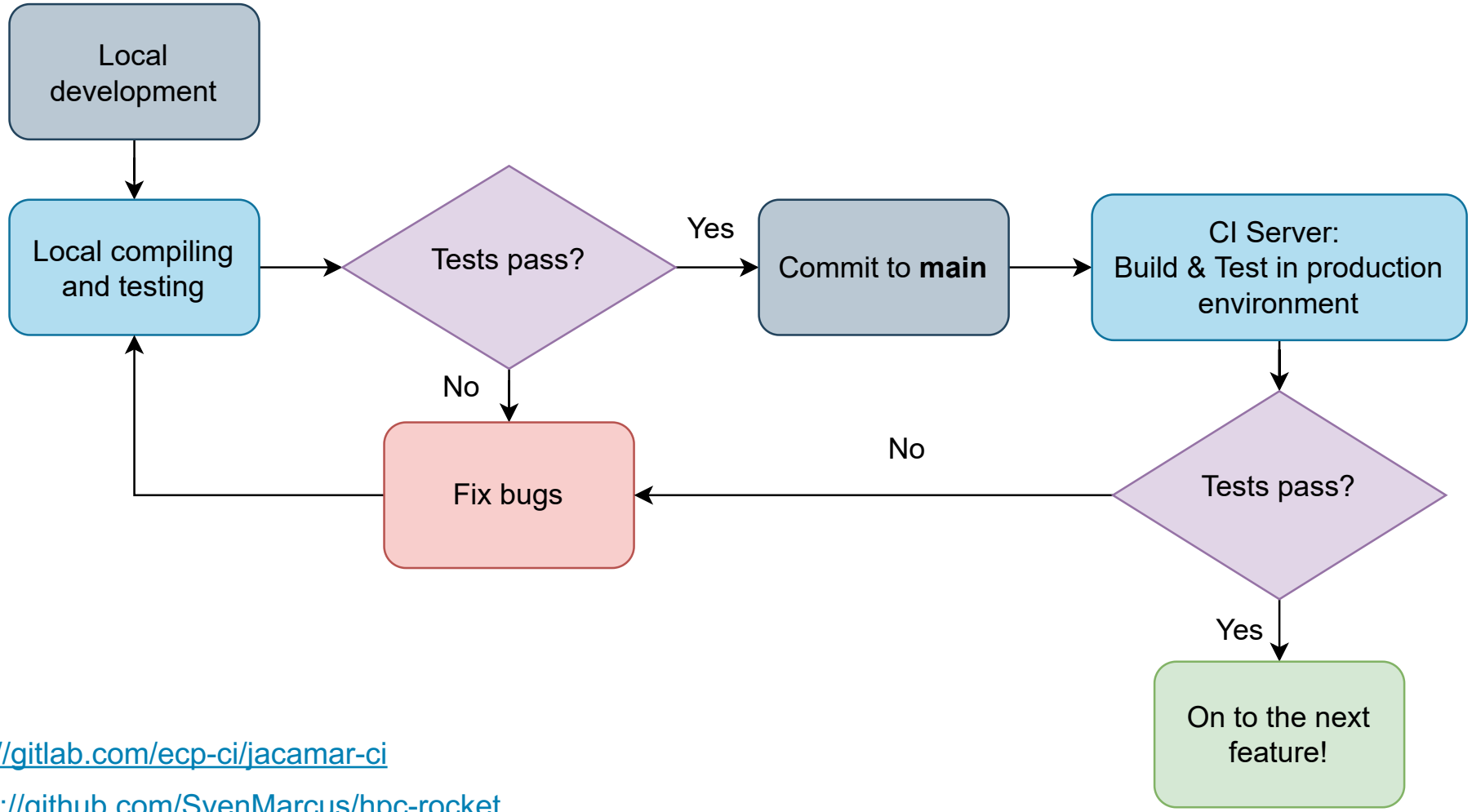
Version Control

- Track and manage changes of source code
- Commits create versions with unique identifier, documenting changes over time
- Enable collaboration through centralized repository hosting platforms (e.g. GitLab)



https://doi.org/10.5281/zenodo.10002995

Continuous integration

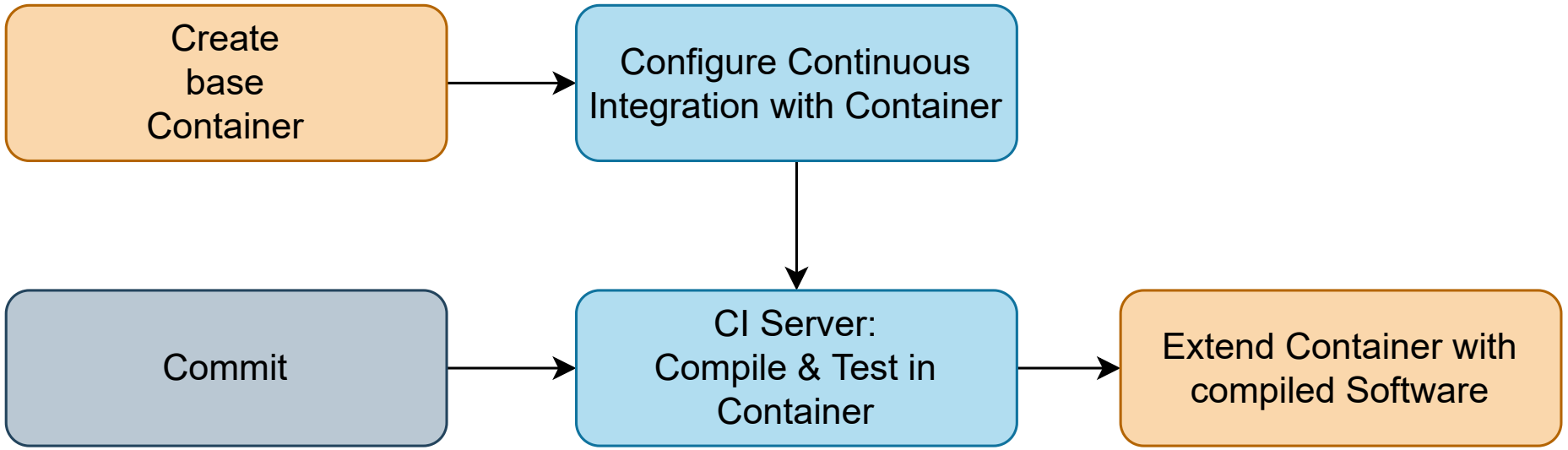


Jacamar CI: https://gitlab.com/ecp-ci/jacamar-ci

HPC-Rocket: https://github.com/SvenMarcus/hpc-rocket

https://doi.org/10.5281/zenodo.10002995

Continuous analysis

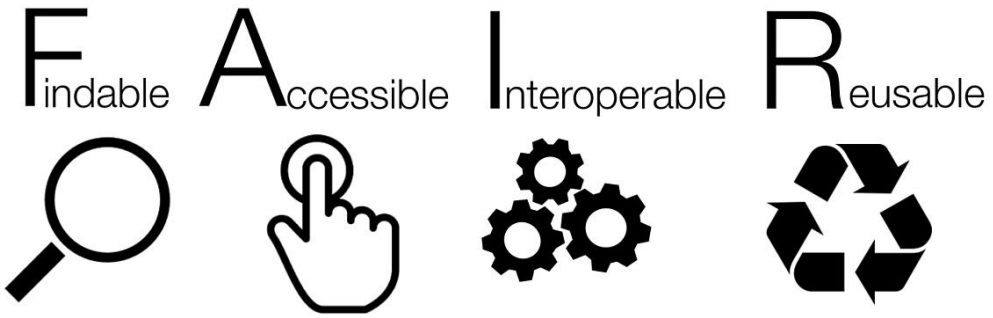


https://doi.org/10.1038/nbt.3780

Publication & Archiving

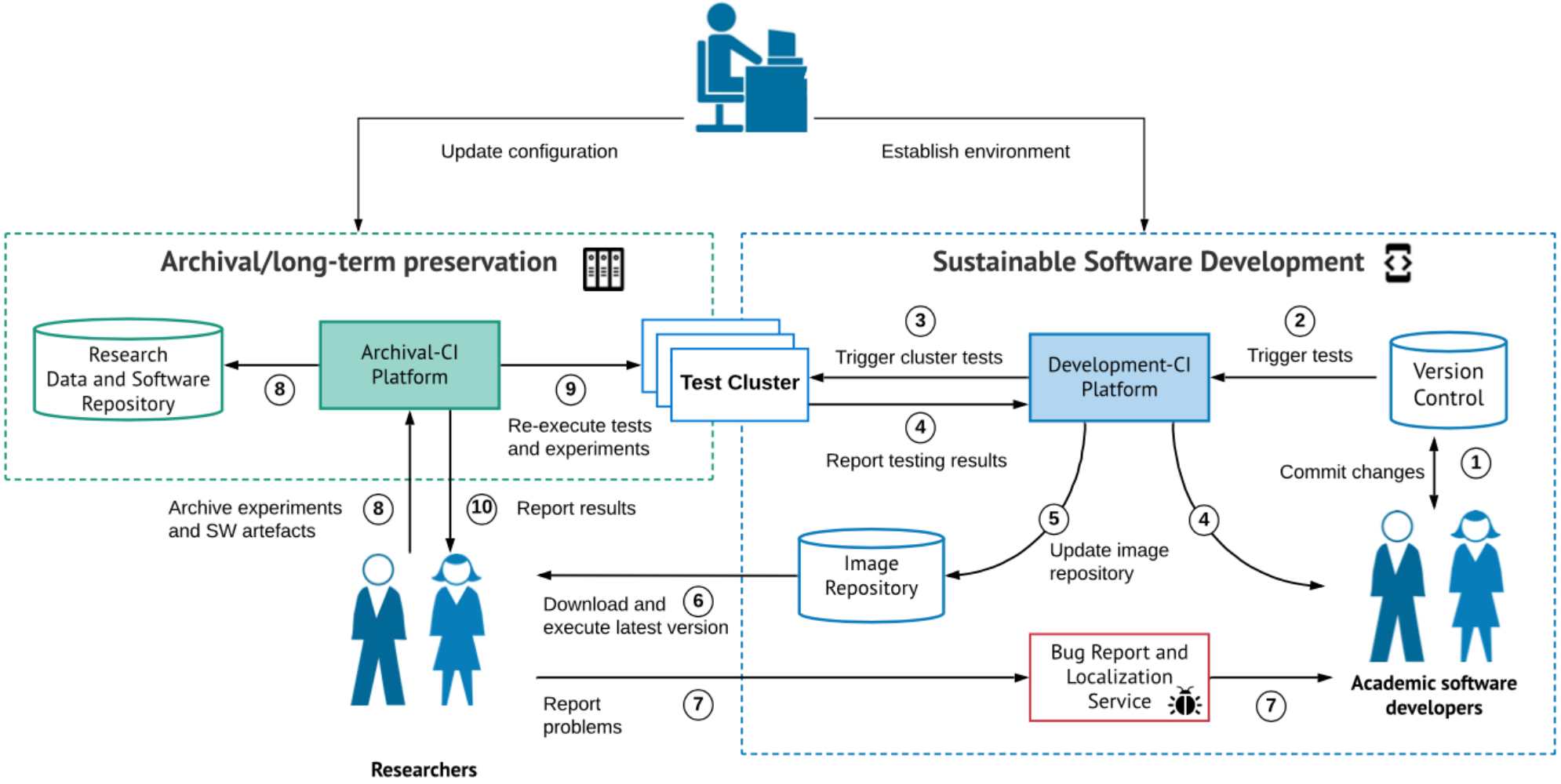
- publish & archive the source code and the compiled executable together with a complete runtime environment in an accessible repository

- provide meaningful metadata including a unique identifier (DOI) to ensure citability, findability and reusability according the FAIR principles.



https://doi.org/10.5281/zenodo.10002995

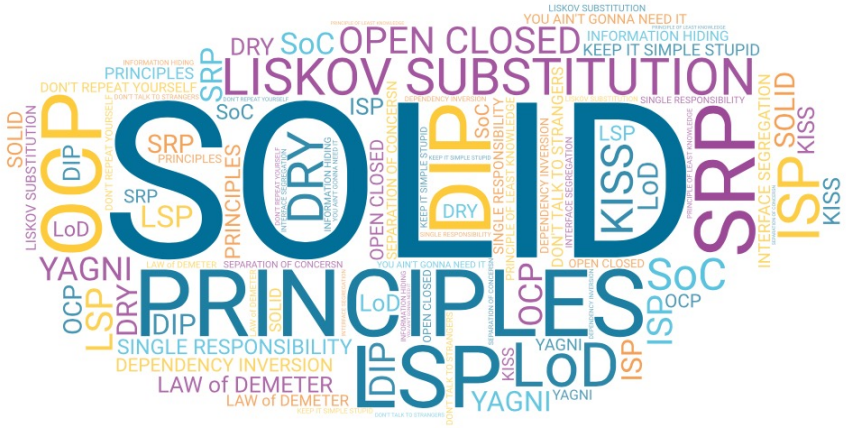
SURESOFT workflow

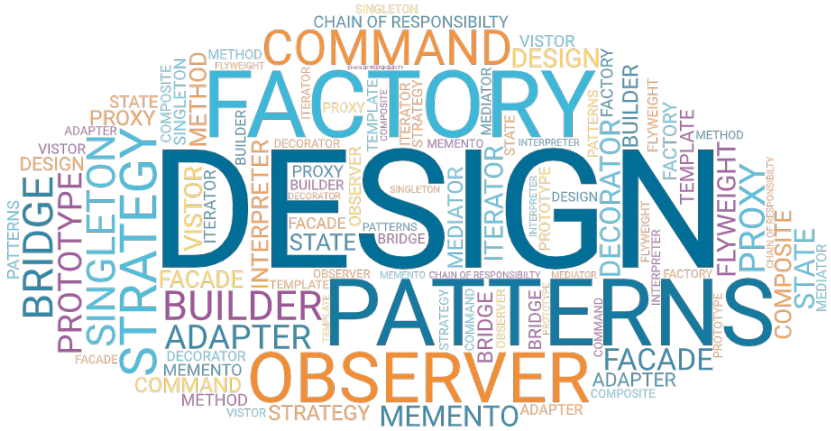


https://doi.org/10.5281/zenodo.10002995

Education & Support

- Documentation
- Software Engineering Principles:
  - SOLID
  - Design Patterns
- Testing, TDD
- Version Control & CI
- Containerization
- Research Data Management & Long-term archiving
- Software Licensing





https://doi.org/10.5281/zenodo.10002995

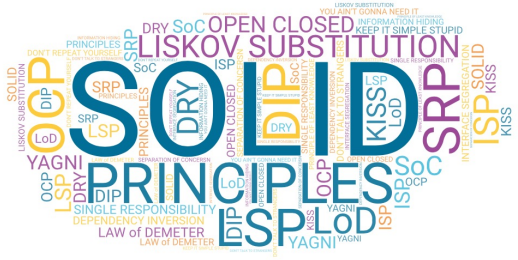
Workshop Series

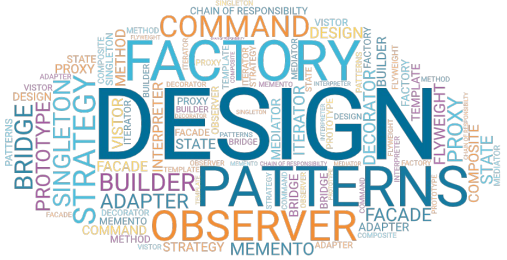
- SURESOFT Workshop Series 2022: 9 Workshops
- deRSE 2023
- RSE UK 2023
- HefDi Summer School 2023: 4 Workshops
- https://suresoft.dev

https://zenodo.org/communities/suresoft

New Workshop-Series

HeFDI Code School 'Sustainable Research Software'





4 half-day workshops from 9 am – 1 pm

- 27 October 2023
- 10 November 2023
- 24 November 2023
- 8 December 2023

https://lnk.tu-bs.de/8oyp07

Get in touch!

website online https://suresoft.dev

chat 19 user https://matrix.to/#/#suresoft-general:matrix.org

Zenodo Community https://zenodo.org/communities/suresoft

Mailinglist https://lists.tu-braunschweig.de/sympa/info/musen-rse

https://git.rz.tu-bs.de/suresoft

https://doi.org/10.5281/zenodo.10002995

ACKNOWLEDGMENTS

The SURESOFT project is funded by the German Research Foundation (DFG) as part of the “e-Research Technologies” funding programme under grants: EG 404/1-1, JA 2329/7-1, KA 3171/12-1, KU 2333/17-1, LA 1403/12-1, LI 2970/1-1 and STU 530/6-1.

Bibliograhpy

[1] Monya Baker. “1,500 scientists lift the lid on reproducibility”. In: *Nature* 533.7604 (Mai 2016), S. 452–454. ISSN: 0028-0836. DOI: 10.1038/533452a.

[2] Mark De Rond und Alan N Miller. “Publish or perish: bane or boon of academic life?” In: *Journal of management inquiry* 14.4 (2005), S. 321–329.

[3] Jo Hannay u. a. “How Do Scientists Develop and Use Scientific Software?” In: *Software Engineering for Computational Science and Engineering, ICSE Workshop on* 0 (Mai 2009), S. 1–8. DOI: 10.1109/SECSE.2009.5069155.

[4] Lucas Joppa u. a. “Troubling Trends in Scientific Software Use”. In: *Science (New York, N.Y.)* 340 (Mai 2013), S. 814–815. DOI: 10.1126/science.1231535.

[5] Zeeya Merali. “Computational science: Error, why scientific programming does not compute”. In: *Nature* 467.7317 (Okt. 2010), S. 775–777. ISSN: 0028-0836. DOI: 10.1038/467775a.

[6] Mark D Wilkinson u. a. “The FAIR Guiding Principles for scientific data management and stewardship”. In: *Scientific data* 3.1 (2016), S. 1–9.