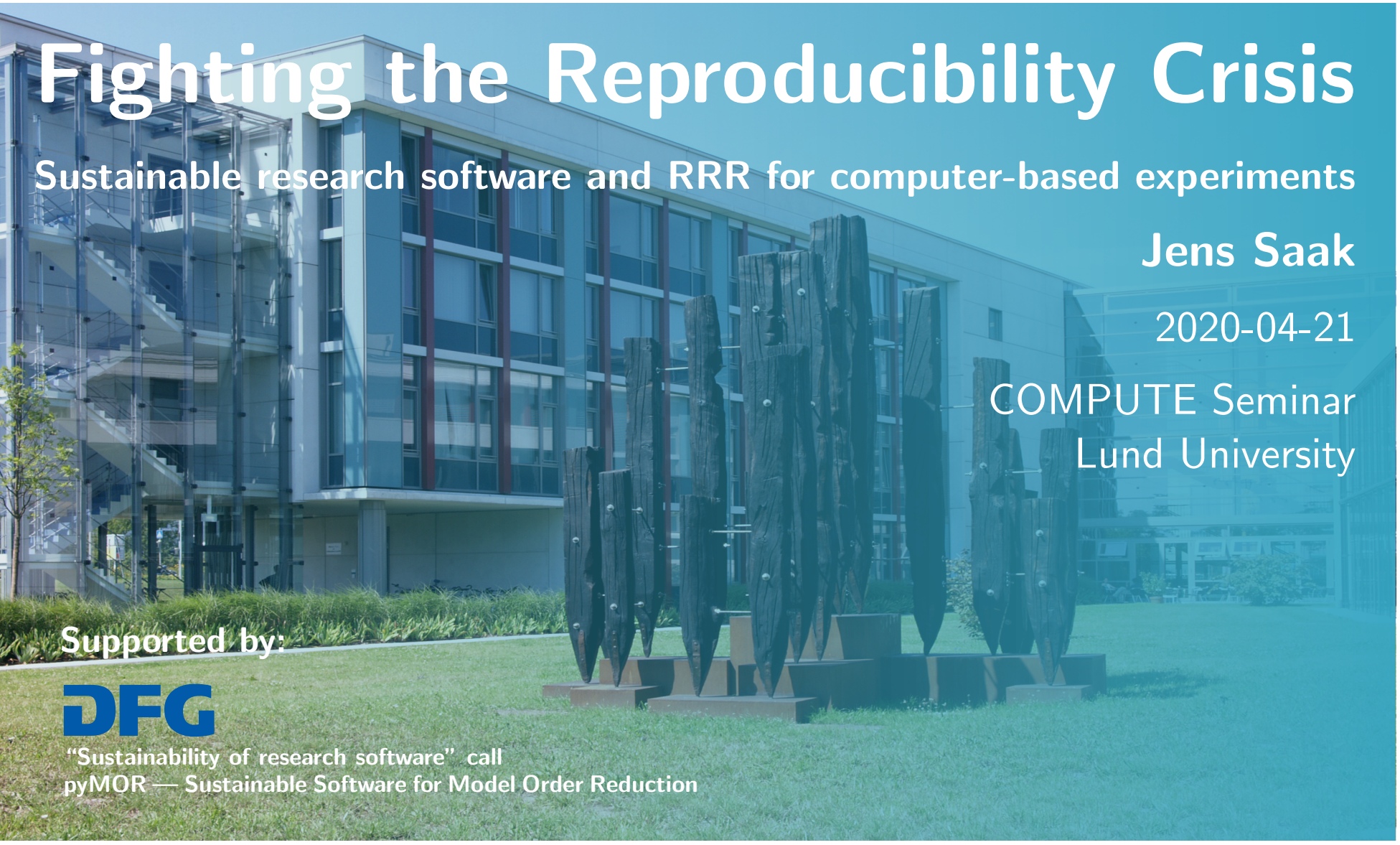


MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC
COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

# Fighting the Reproducibility Crisis

## Sustainable research software and RRR for computer-based experiments

**Jens Saak**

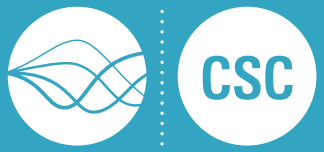
2020-04-21

COMPUTE Seminar
Lund University

**Supported by:**

DFG

"Sustainability of research software" call
pyMOR — Sustainable Software for Model Order Reduction



# Outline

1. Motivation
2. RRR to FAIR
3. Proposed Development Practices

# The Crew

Jörg Fehr

Uni Stuttgart

Jan Heiland

MPI Magdeburg

Christian Himpe

MPI Magdeburg

Stephan Rave

Uni Münster

Jens Saak

MPI Magdeburg

# The Crew

Jörg Fehr

Uni Stuttgart

Jan Heiland

MPI Magdeburg

Christian Himpe

MPI Magdeburg

Stephan Rave

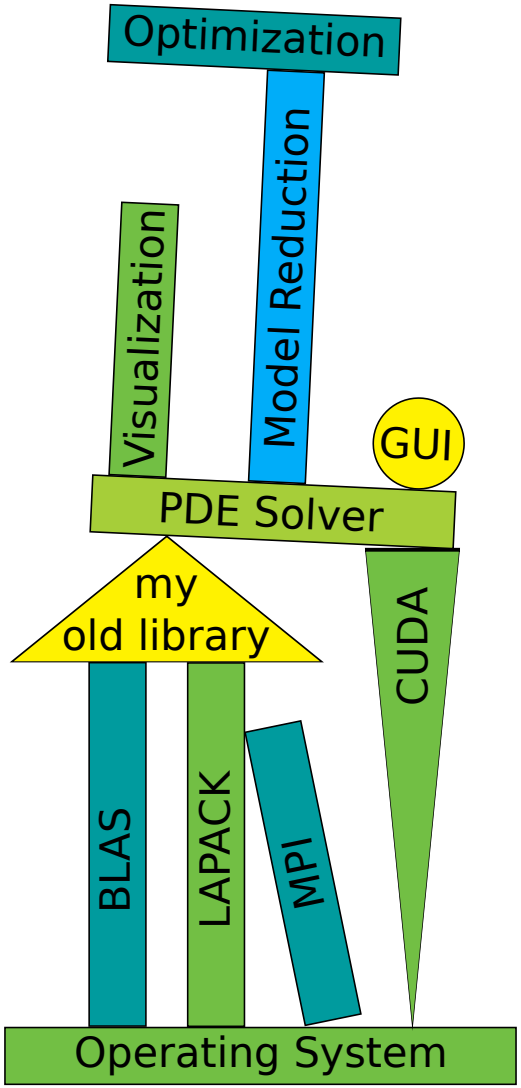
Uni Münster

Jens Saak

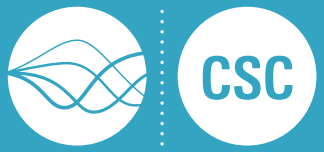
MPI Magdeburg

→ Together about one century of programming experience

# Generic Research Code 1

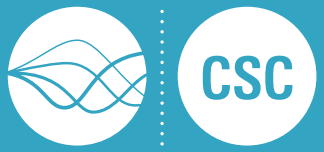



"Tower of Doom" (by: S. Rave)



# Generic Research Code 2

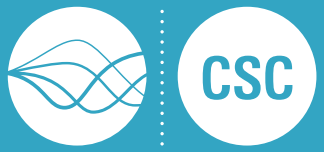
"The Void" (by: C. Himpe)



# Our Aim

**Improve Computer-Based Experiments (CBEx):**

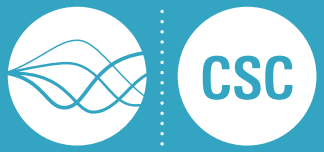
- Create problem-awareness and
- Ensure **scientificity** and **progress**



# Our Aim

**Improve Computer-Based Experiments (CBEx):**

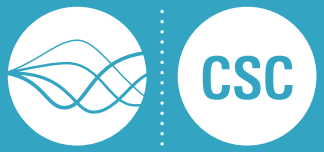
- Create problem-awareness and
- Ensure **scientificity** and **progress**
- Define terminology
- Establish best-practices
- Formulate discipline-agnostic practical guidelines



# Our Aim

**Improve Computer-Based Experiments (CBEx):**

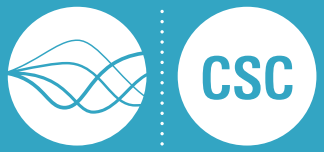
- Create problem-awareness and
- Ensure **scientificity** and **progress**
- Define terminology
- Establish best-practices
- Formulate discipline-agnostic practical guidelines
- Improve **availability** and **quality** of research **software**




# Computer-Based Experiments (CBEx)

## What is a CBEx?

## What is a scientific CBEx?




# Computer-Based Experiments (CBEx)

## What is a CBEx?

- **Any result obtained by a computer.**
- No matter if it is:
- supporting or illustrative results,
- pointwise confirmation,
- or computational proof.

## What is a scientific CBEx?

# Computer-Based Experiments (CBEx)

## What is a CBEx?

- **Any result obtained by a computer.**
- No matter if it is:
- supporting or illustrative results,
- pointwise confirmation,
- or computational proof.

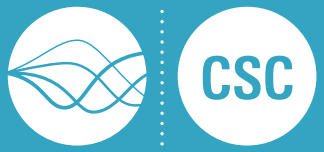
## What is a scientific CBEx?

- Any **CBEx** by which the authors' claim is **verifiable**.

# CBEx Problems

**Sorted by increasing commonality:**

- Hardware not available
- Software stack not available
- Reporting not sufficient
- Archiving not stable
- Provisioning not sufficient
- **Lack of education**



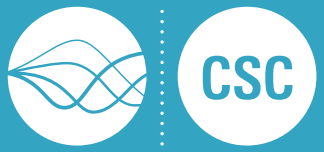

# Disclaimer

- The following is not a strict set of rules.
- View it as a collection of best-practices.
- Adapt these ideas to your use-case.

# RRR to FAIR

**based on**

J. Fehr, J. Heiland, C. H., J. Saak. **Best Practices for Replicability, Reproducibility and Reusability of Computer-Based Experiments Exemplified by Model Reduction Software**. AIMS Mathematics 1(3): 261–281, 2016. https://doi.org/bsb2



# RRR to FAIR: The Three Rs

1. **R**eplicability
2. **R**eproducibility
3. **R**eusability

Each **R** has:

- Minimal requirements
- Optional recommendations

# RRR to FAIR: Replicability

**Definition**

The attribute **Replicability** describes the ability to repeat a CBEx and to come to the same (in a numerical sense) results. Sometimes the equivalent term **Repeatability** is used for this experimental property.

- Replicability is a basic requirement of reliable software as well as of its result as it shows a certain robustness of the procedure against
  - statistical influences
  - and bias of the observer.
- Also, **only replicable CBEx can serve as a benchmark** to which new methods can be compared, cf. [Vitek & Kalibera '11].

# RRR to FAIR: Replicability

## The Essence of Replicability (aka Repeatability)

- **You** are able
- to **repeat**
- **your experiment**
- on **your computer**.

# RRR to FAIR: Replicability

## The Essence of Replicability (aka Repeatability)

- **You** are able
- to **repeat**
- **your experiment**
- on **your computer**.

## Minimal Requirements

**Basic Documentation:**

- Recipe to obtain (numerical) results
- Recipe for post-processing of data
- Recipe for creating visualizations

# RRR to FAIR: Replicability

## The Essence of Replicability (aka Repeatability)

- **You** are able
- to **repeat**
- **your experiment**
- on **your computer**.

## Minimal Requirements

**Basic Documentation:**

- Recipe to obtain (numerical) results
- Recipe for post-processing of data
- Recipe for creating visualizations

## Optional Recommendations

**Automation and Testing:**

- Machine-readable recipes
- For example (shell) scripts
- Sanity tests

# RRR to FAIR: Reproducibility

**Definition**

**Reproducibility** of a CBEx means that it can be repeated by a different researcher in a different computer environment.

This is an adaption of the general concept of Reproducibility

- that is key in any science that relies on experiments,
- that is a subject in the theory of science, and
- which **absence** in a significant fraction of publications in many research areas has **shaped the term Reproducibility crisis** in recent years [Marcus '13]; cf. also [Collberg, Proebsting, & Warren '04] on Reproducibility in computer science.

(`https://en.wikipedia.org/wiki/Replication_crisis` collects $>$ 100 references across the sciences.)

# RRR to FAIR: Reproducibility

## The Essence of Reproducibility

- **Someone else** is able
- to **repeat**
- **your experiment**
- on **their computer**.

# RRR to FAIR: Reproducibility

## The Essence of Reproducibility

- **Someone else** is able
- to **repeat**
- **your experiment**
- on **their computer**.

## Minimal Requirements

**Detailed Documentation:**

- Environment description
- Versions of system and dependencies
- Building instructions (if applicable)

# RRR to FAIR: Reproducibility

## The Essence of Reproducibility

- **Someone else** is able
- to **repeat**
- **your experiment**
- on **their computer**.

## Minimal Requirements

**Detailed Documentation:**

- Environment description
- Versions of system and dependencies
- Building instructions (if applicable)

## Optional Recommendations

**Availability:**

- Location with long-term storage
- Storage is not bound to author
- persistent identifier is provided

# RRR to FAIR: Reusability

**Definition**

In the sphere of CBEx, **Reusability** refers to the possibility to reuse the software or parts thereof for different purposes, in different environments, and by researchers other than the original authors.

- In particular, Reusability enables the utilization of the test setup or parts of it for other experiments or related applications.
- Although theoretically, any bit of a software can be reused for different purposes, here, Reusability applies only for reproducible parts.

# RRR to FAIR: Reusability

## The Essence of Reusability

- **Someone else** is able
- to **use your experiment**
- on **their computer**.
- for **something else**.

# RRR to FAIR: Reusability

## The Essence of Reusability

- **Someone else** is able
- to **use your experiment**
- on **their computer**.
- for **something else**.

## Minimal Requirements

**Accessibility:**

- Availability (Code, Howto)
- Remote access required
- Binaries available (if applicable)

# RRR to FAIR: Reusability

## The Essence of Reusability

- **Someone else** is able
- to **use your experiment**
- on **their computer**.
- for **something else**.

## Minimal Requirements

**Accessibility:**

- Availability (Code, Howto)
- Remote access required
- Binaries available (if applicable)

## Optional Recommendations

**Modularity, Software Management and Licensing:**

- Modular design
- Project management facilities
- License considerations

# RRR to FAIR: RRR Summary

- **Replicability**

  Required: Basic Documentation
  Recommended: Automation & Testing

- **Reproducibility**

  Required: Extensive Documentation
  Recommended: Availability

- **Reusability**

  Required: Accessibility
  Recommended: Software Management, Modularity & Licensing

# The Road to Sustainability

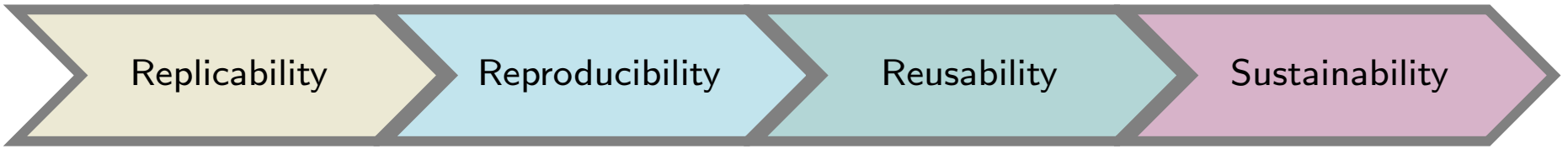
- Replicability ← Verifies your findings

# The Road to Sustainability

- Replicability ← Verifies your findings
- Reproducibility ← Ensures it is science

# The Road to Sustainability

- Replicability ← Verifies your findings
- Reproducibility ← Ensures it is science
- Reusability ← **Enables scientific progress**



# The Road to Sustainability

- Replicability ← Verifies your findings
- Reproducibility ← Ensures it is science
- Reusability ← **Enables scientific progress**


Sustainable software is:

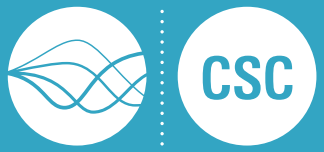
**F**indable, **A**ccessible, **I**nteroperable, **R**eusable

more

# Proposed Development Practices

**based on**

J. Fehr, C. Himpe, S. Rave, J. S. **Sustainable Research Software Hand-Over**. arXiv, cs.GL: 1909.09469, 2019.
https://arxiv.org/abs/1909.09469



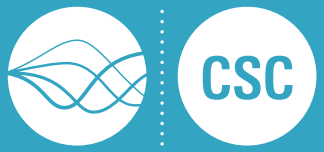
# Proposed Development Practices

- small project

  paper code, thesis project code

- large project

  groups in-house tool, community code, . . .



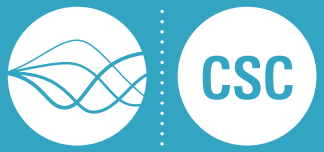
# Proposed Development Practices

- small project ← often single developer and user

  paper code, thesis project code

- large project

  groups in-house tool, community code, . . .



# Proposed Development Practices

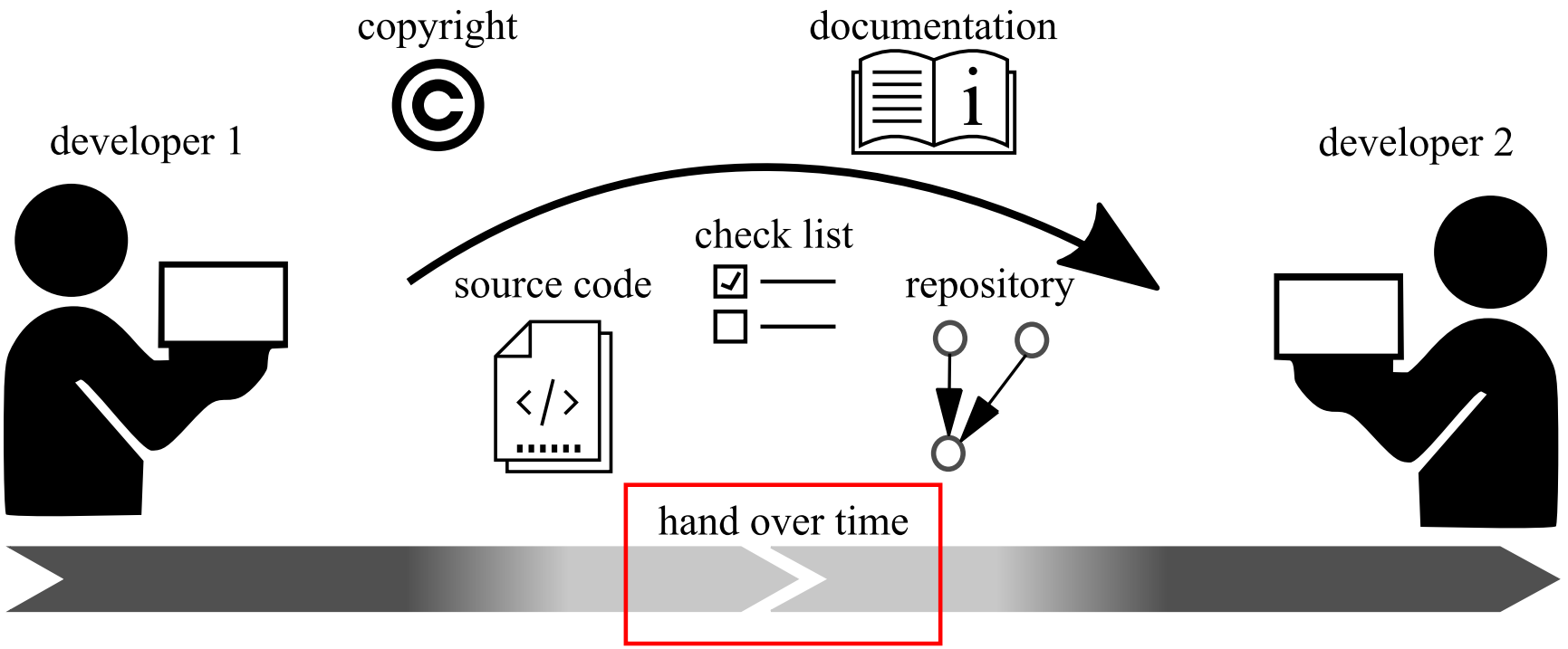
- small project ← often single developer and user

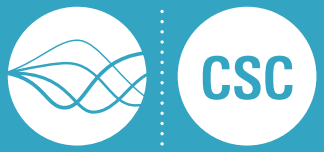
  paper code, thesis project code

- large project ← separate developer and user groups

  groups in-house tool, community code, . . .

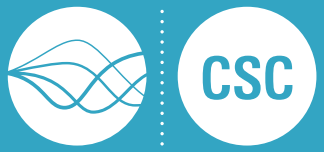
# Small Project





# Small Project: Requirements

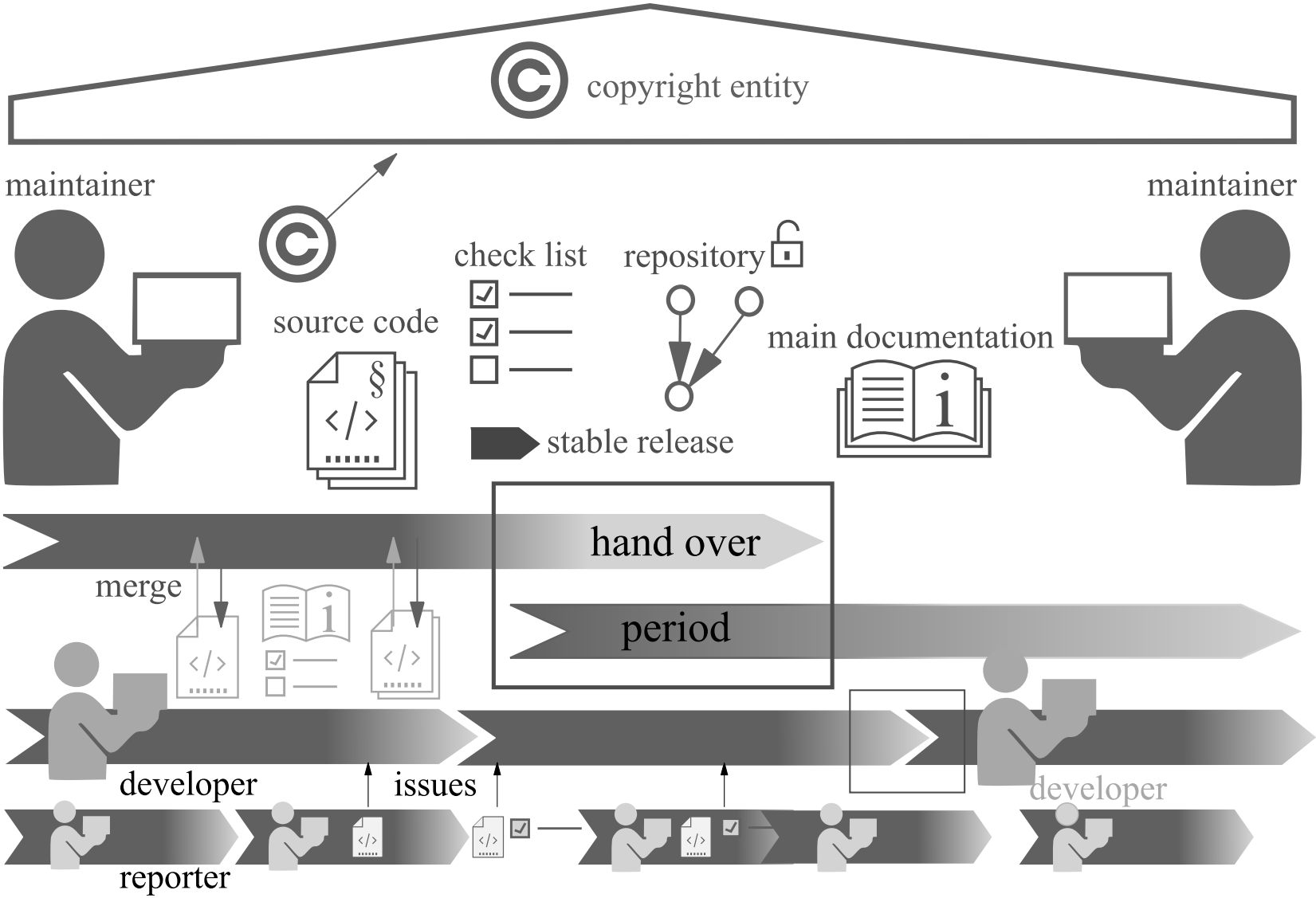
- Code availability
  (recoverable from central institute repository)
- Working example(s)
  (RUNME, easier handover, usable for testing)
- Code ownership
  (institution? supervisor? developer?)
- Execution environment
  (documentation of soft- and hardware for compilation and execution)
- Minimal documentation
  (README)



# Small Project: Recommendations

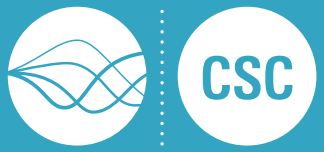
- Public release
  (License? Find community repositories: `https://re3data.org/`)
- Version control
  (track changes, named revisions, BACKUP!)
- Basic code cleanup
  (obscure constants, dead code, hard-paths)
- Reproducible execution environment
  (virtual machine, container, step-by-step guide, . . . )
- Integration into larger project
  (e.g. in-house or community code / modularity? interfaces?)

# Large Project



# Large Project: Requirements

- Software license
  (license compatibility? `https://ufal.github.io/public-license-selector/`)
- Code ownership of contributions
  (re-licensing, availability of copyright holders, . . . )
- Access to project resources
  (website, code repo, mailing list, support desk,. . . )
  (developer hierarchy, responsibilities)
- Development in branches
  (stable master, management of branches, . . . )
- Changelog
  (compressed history for smooth handover)



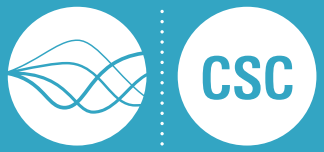

# Large Project: Recommendations

- Code maintainability
  (Code reviews, automatic testing and deployment (CI))
- Code of Conduct
  (handover guidelines, new and leaving maintainers, . . . )
- Contribution Policy
  (who? how? required skills?)
- Citation Policy
  (Do developers/authors get the credits?)

# Wrap-up!

## As an author make your . . .

- . . . CBEx replicable, reproducible, reusable.
- . . . scientific software sustainable and FAIR.



# Wrap-up!

**As an author make your . . .**

- . . . CBEx replicable, reproducible, reusable.
- . . . scientific software sustainable and FAIR.

**As a reviewer/editor ask the authors to do so.**

# Wrap-up!

**As an author make your . . .**

- . . . CBEx replicable, reproducible, reusable.
- . . . scientific software sustainable and FAIR.

**As a reviewer/editor ask the authors to do so.**

**Questions? Remarks? Suggestions?**

# FAIR principles [12]

https://www.go-fair.org/fair-principles/

**F** indable

*"... Metadata and data should be easy to find for both humans and computers. Machine-readable metadata are essential for automatic discovery of datasets and services, ..."* persistent identifier, rich & clear metadata, searchable resource

# FAIR principles [12]

https://www.go-fair.org/fair-principles/

**F** indable

*"... Metadata and data should be easy to find for both humans and computers. Machine-readable metadata are essential for automatic discovery of datasets and services, ..."* persistent identifier, rich & clear metadata, searchable resource

**A** ccessible

*"Once the user finds the required data, she/he needs to know how can they be accessed, possibly including authentication and authorisation."*

open, free and universal protocol with authentication where necessary

# FAIR principles [12]

https://www.go-fair.org/fair-principles/

**F** indable

*"... Metadata and data should be easy to find for both humans and computers. Machine-readable metadata are essential for automatic discovery of datasets and services, ..."* persistent identifier, rich & clear metadata, searchable resource

**A** ccessible

*"Once the user finds the required data, she/he needs to know how can they be accessed, possibly including authentication and authorisation."*

open, free and universal protocol with authentication where necessary

**I** nteroperable

*"The data usually need to be integrated with other data. In addition, the data need to interoperate with applications or workflows for analysis, storage, and processing."* (meta)data in common language and fair vocabulary with qualified cross-references

# FAIR principles [12]

https://www.go-fair.org/fair-principles/

**F** indable

*"... Metadata and data should be easy to find for both humans and computers. Machine-readable metadata are essential for automatic discovery of datasets and services, ..."* persistent identifier, rich & clear metadata, searchable resource

**A** ccessible

*"Once the user finds the required data, she/he needs to know how can they be accessed, possibly including authentication and authorisation."*

open, free and universal protocol with authentication where necessary

**I** nteroperable

*"The data usually need to be integrated with other data. In addition, the data need to interoperate with applications or workflows for analysis, storage, and processing."* (meta)data in common language and fair vocabulary with qualified cross-references

**R** eusable

*"The ultimate goal of FAIR is to optimise the reuse of data. To achieve this, metadata and data should be well-described so that they can be replicated and/or combined in different settings."*

(meta)data in community standard repreentation follows clear and accessible license

back

# Related Material

- Software deposit guidance for researchers [10]
  (The Software Sustainability Institute)
- Recommendations on the development, use and provision of Research Software [9]
  (Alliance of German Science Organizations)
- Criteria fo Software Self-Assessment [6]
  (INRIA Evaluation Committee)
- Open Source Guides [5]
  (GitHub and friends)
- Code of Conduct
  (Your favorite research organization or funding agency)
- ...

# Further Reading I


[1] W. Bangerth and T. Heister, *Quo vadis, scientific software?*, SIAM News, 47 (2014), https://sinews.siam.org/Details-Page/quo-vadis-scientific-software-1. Accessed: 2020-04-16.

[2] C. Collberg, T. Proebsten, and A. M. Warren, *Repeatability and benefaction in computer systems research*, tech. report, University of Arizona, 2014, http://reproducibility.cs.arizona.edu/v2/RepeatabilityTR.pdf. Accessed: 2016-09-22.

[3] J. Fehr, J. Heiland, C. Himpe, and J. Saak, *Best practices for replicability, reproducibility and reusability of computer-based experiments exemplified by model reduction software*, AIMS Mathematics, 1 (2016), pp. 261–281, https://doi.org/10.3934/Math.2016.3.261.

[4] J. Fehr, C. Himpe, S. Rave, and J. Saak, *Sustainable research software hand-over*, e-print arXiv:1909.09469, arXiv cs.GL, Sept. 2019, https://arxiv.org/abs/1909.09469.

# Further Reading II


[5] GitHub and Friends, *Open Source Guides*, GitHub, https://opensource.guide/.
Accessed 2019-02-17.

[6] INRIA Evaluation Committee, *Criteria for Software Self-Assessment*, INRIA, Aug. 2011, https://www.inria.fr/content/download/12702/427946/version/2/file/softwarecriteria-ce_2011-08-01.pdf.

[7] R. J. LeVeque, *Top ten reasons to not share your code (and why you should anyway)*, SIAM News, 46 (2013), http://archive.is/eAr7z.

[8] G. Marcus, *The crisis in social psychology that isn't*, 2013, https://www.newyorker.com/tech/elements/the-crisis-in-social-psychology-that-isnt.

# Further Reading III


[9] Research Software Working Group in the Priority Initiative Digital Information of the Alliance of German Science Organisations, *Recommendations on the development, use and provision of research software*, Mar. 2018, https://doi.org/10.5281/zenodo.1172988.

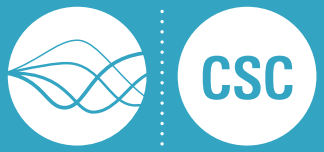
version 1.0.

[10] The Software Sustainability Institute, *Software deposit guidance for researchers*, Aug. 2018, https://softwaresaved.github.io/software-deposit-guidance/.

edited by Michael Jackson.

[11] J. Vitek and T. Kalibera, *Repeatability, reproducibility, and rigor in systems research*, in Proceedings of the 9th ACM International Conference on Embedded Software, 2011, pp. 33–38, https://doi.org/10.1145/2038642.2038650.

[12] M. D. Wilkinson, et al., *The FAIR Guiding Principles for scientific data management and stewardship*, Science Data, 3 (2016), https://doi.org/10.1038/sdata.2016.18.

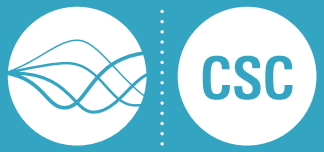


# Compute Environment

**Useful Minimal Information** (MATLAB, Octave, Python, R, Julia):

- Runtime interpreter name and version
- Operating system name, version and architecture / word-width
- *Processor name and exact identifier*
- Required amount of random access memory
- BLAS / LAPACK library implementation name and version

back



# Timing Results

**Pitfalls:**

- CPU time vs wall time
- Parallelization (implicit / explicit)
- Efficient memory access (NUMA)
- Overhead (actual compute-time)
- Statistics (i.e. means of repeated runs)

# Code Availability Section

## Numerical Results

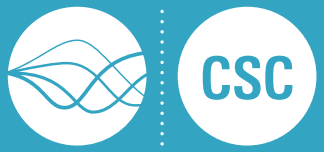
...

## Code Availability Section

> The source code of the implementations used to compute the presented results can be obtained from:
>
> `https://my.stable.url`
>
> and is authored by: X. Y., A. B.

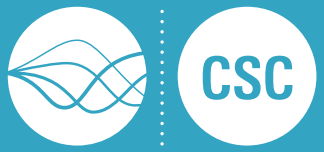
(if available use supplemental material!)

back



# Standard Paper Files

| | |
|---|---|
| `README` | Read this to get started! |
| `RUNME` | Run this to get started! |
| `CODE` | Machine readable code meta-data |
| `CITATION` | How to cite the software? |
| | ... |



# Standard Project Files

| | |
|---|---|
| `AUTHORS` | Who wrote it |
| `LICENSE` | The license text |
| `INSTALL` | How to install |
| `CHANGELOG` | What changed |
| `DEPENDENCIES` | What are the dependencies |
| `VERSION` | The version number |
| `TODO` | Open problems |
| `FAQ` | Frequently Asked Questions |
| | ... |