



© HZDR / Detlev Müller

# Introduction to Research Software Engineering

Introduction / History / FAIR-RS

Department for Information Services and Computing · Computational Science · Dr. Guido Juckeland · [g.juckeland@hzdr.de](mailto:g.juckeland@hzdr.de) · [www.hzdr.de/fwcc](http://www.hzdr.de/fwcc)



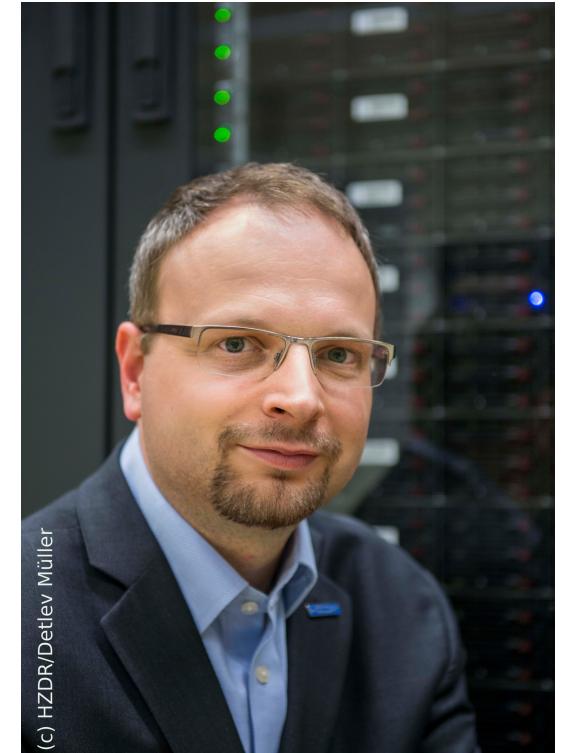
# Getting started



# Who am I?

## Dr. Guido Juckeland

- Dipl.-Ing. Informationssystemtechnik @ TUD (2005)
- 2005-2016 working at TUD ZIH as HPC systems engineer, later IT architect
- Research on usage of hardware accelerators
- 2013 Ph.D. @TUD
- Since 2016 Head of Computational Science Department @ HZDR  
→ working together with scientists via
  - Research Software Engineering
  - AI Consulting
  - Data Management and HPC Support
- Other lectures:
  - Highly parallel programming of GPUs (WS, Mon.+Tue. 3. DS)



(c) HZDR/Detlev Müller

# Who are you?

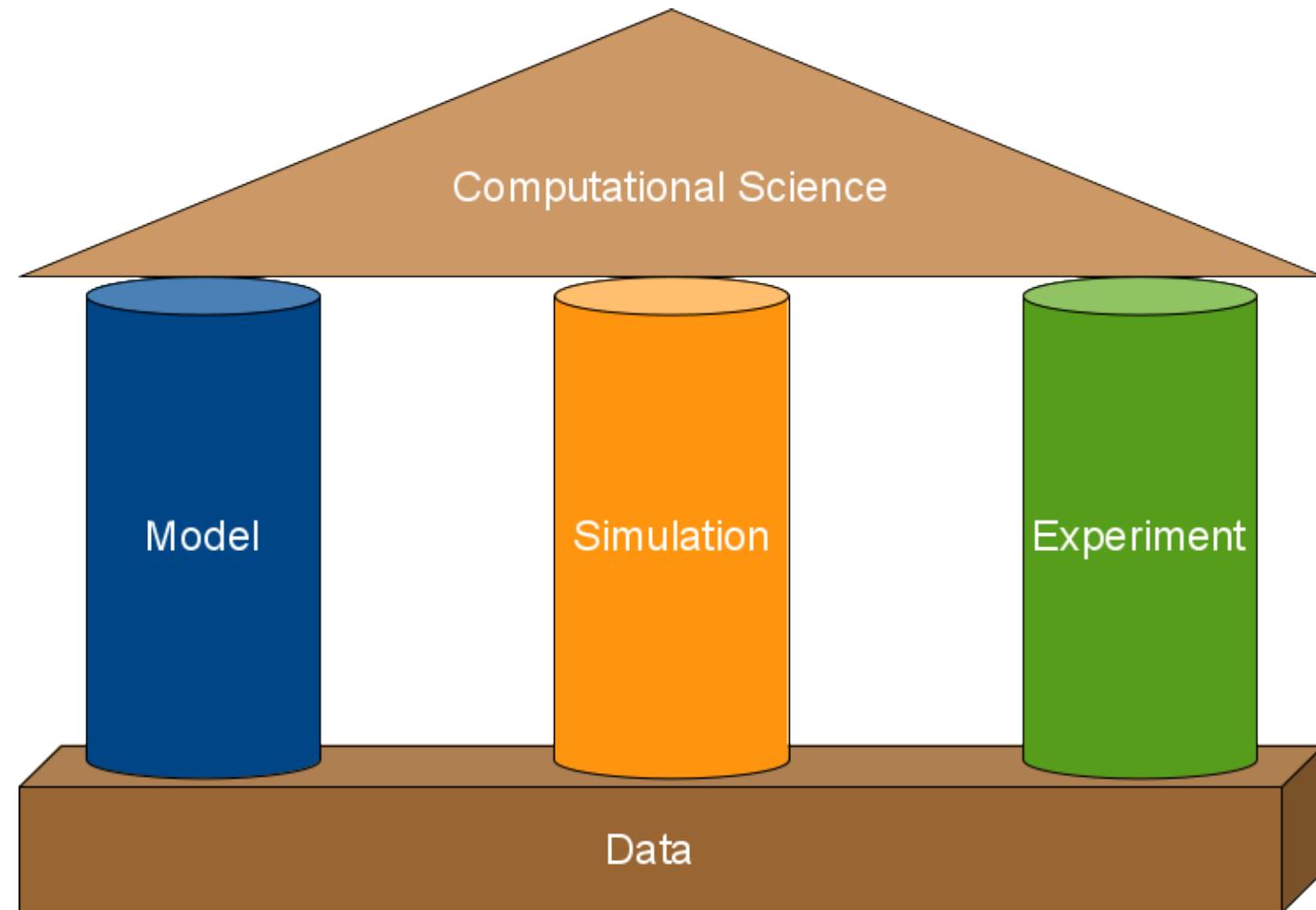
- What is your name?
- What is your branch of study?
  - Why did you choose this?
  - What do you want to do once you graduate?
- What made you come to this course?
- What will make this a successful course for you?



Image by Katerina Limpitsouni via [undraw.co](https://undraw.co)

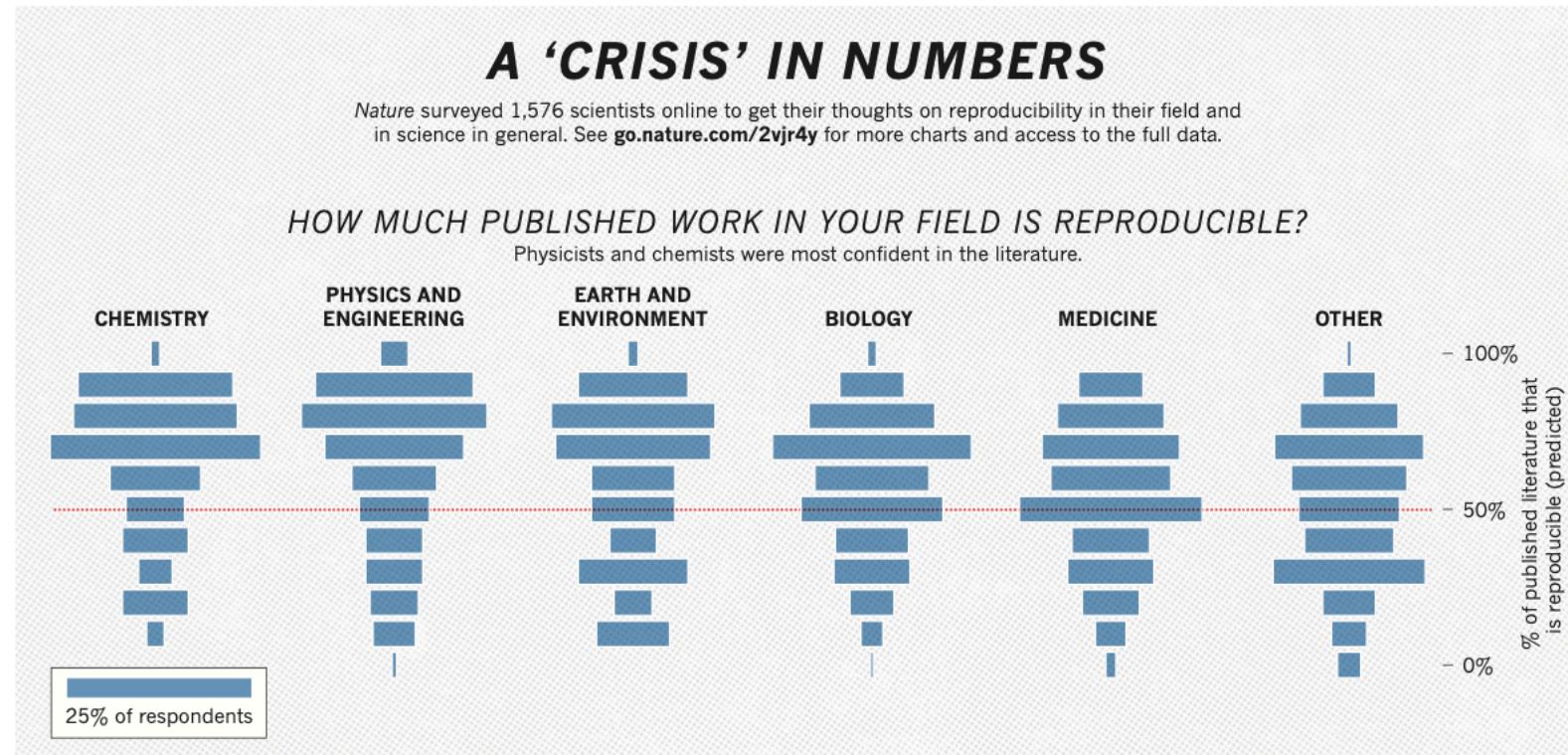
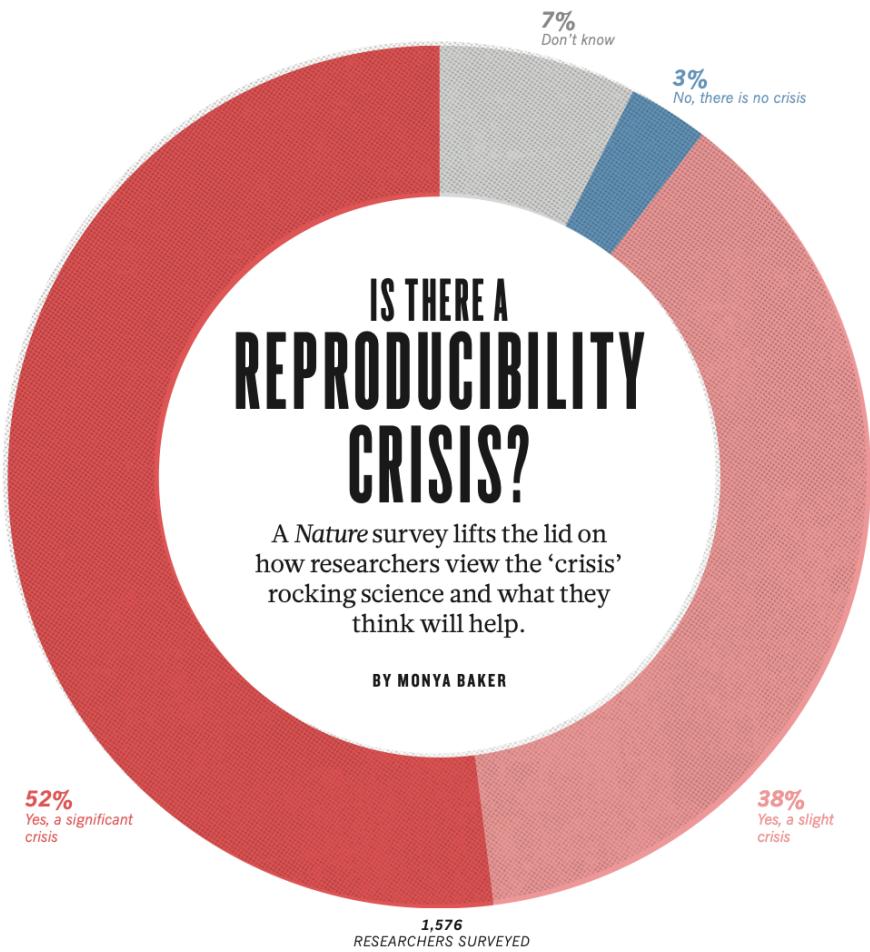
# Why research software engineering?

Software is part of all (digital) scientific aspects



# Why research software engineering?

## Reproducibility crisis in science



Baker, M. 1,500 scientists lift the lid on reproducibility. *Nature* 533, 452–454 (2016). <https://doi.org/10.1038/533452a>

# A (few) definition approaches

## What is research software?

- Script to convert data from one format to another
- Script to read data and visualize it
- Program that generates data
- Analysis script
- Set of scripts that form an analysis pipeline
- Code that is compiled
- Code that is dynamically interpreted and not compiled
- Web app
- ...

**But there is always the direct research context!**

Bast, R. (2023, August 12). Research software engineering for HPC. Zenodo. <https://doi.org/10.5281/zenodo.8242055>

# A (few) definition approaches

## Looking at literature

Barker, M., Chue Hong, N.P., Katz, D.S. et al. Introducing the FAIR Principles for research software. Sci Data 9, 622 (2022). <https://doi.org/10.1038/s41597-022-01710-x>:

**“Research software is defined by the FAIR4RS WG as including ‘source code files, algorithms, scripts, computational workflows and executables that were created during the research process or for a research purpose. Software components (e.g., operating systems, libraries, dependencies, packages, scripts, etc.) that are used for research but were not created during or with a clear research intent should be considered software in research and not Research Software. This differentiation may vary between disciplines’ “**

Taken from: Gruenpeter, M., Katz, D. S., Lamprecht, A.-L., Honeyman, T., Garijo, D., Struck, A., Niehues, A., Martinez, P. A., Castro, L. J., Rabemanantsoa, T., Chue Hong, N. P., Martinez-Ortiz, C., Sesink, L., Liffers, M., Fouilloux, A. C., Erdmann, C., Peroni, S., Martinez Lavanchy, P., Todorov, I., & Sinha, M. (2021). Defining Research Software: a controversial discussion (Version 1). Zenodo. <https://doi.org/10.5281/zenodo.5504016>

# A (few) definition approaches

Starting point for this course

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

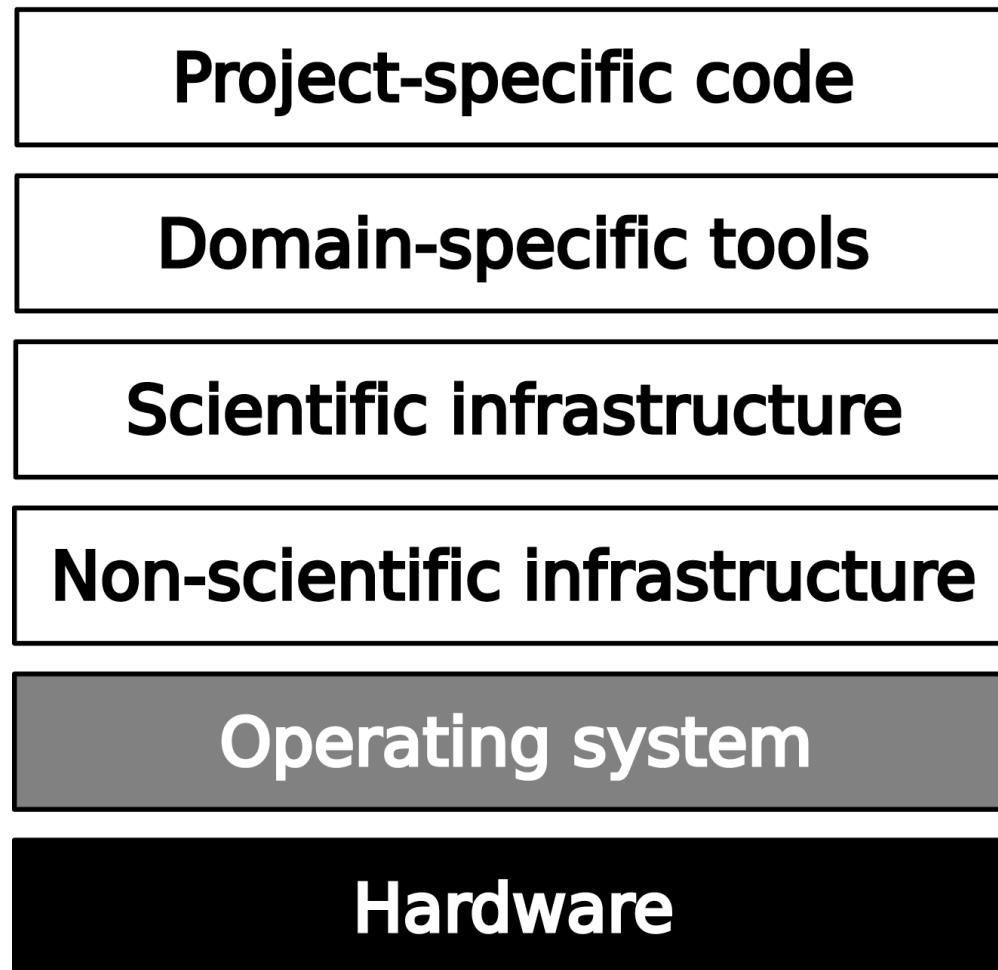
From [https://github.com/CaptainSiff/paper\\_teaching-learning-RSE](https://github.com/CaptainSiff/paper_teaching-learning-RSE) (still ongoing activity)

Also correlates with DFG definition: “**In this sense, the term “research software” refers to the software applications and software libraries specially created for scientific knowledge gain.**” (from [https://www.dfg.de/en/research\\_funding/programmes/infrastructure/lis/funding\\_opportunities/call\\_proposal\\_software/](https://www.dfg.de/en/research_funding/programmes/infrastructure/lis/funding_opportunities/call_proposal_software/))

- This excludes infrastructure software (e.g. numerical libraries etc.) which may be also written by scientists or be used for scientific purposes

# A (few) definition approaches

Visualizing this relationship



*Scripts, notebooks,  
workflows, ...*

**GROMACS, MMTK, ...**  
*(domain: biomolecular simulation)*

**BLAS, HDF5, SciPy, ...**

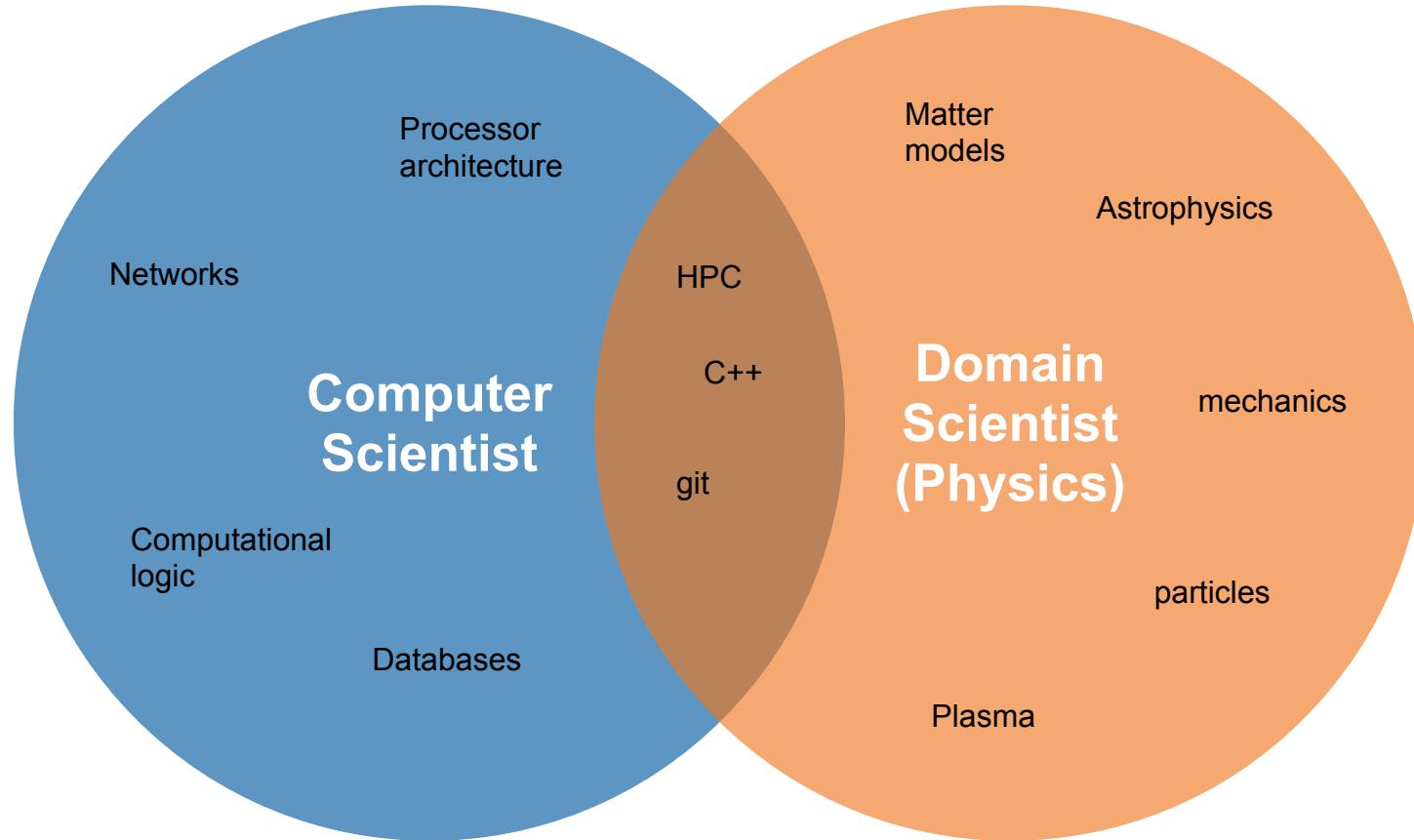
*gcc, Python, ...*

**GNU/Linux, ...**

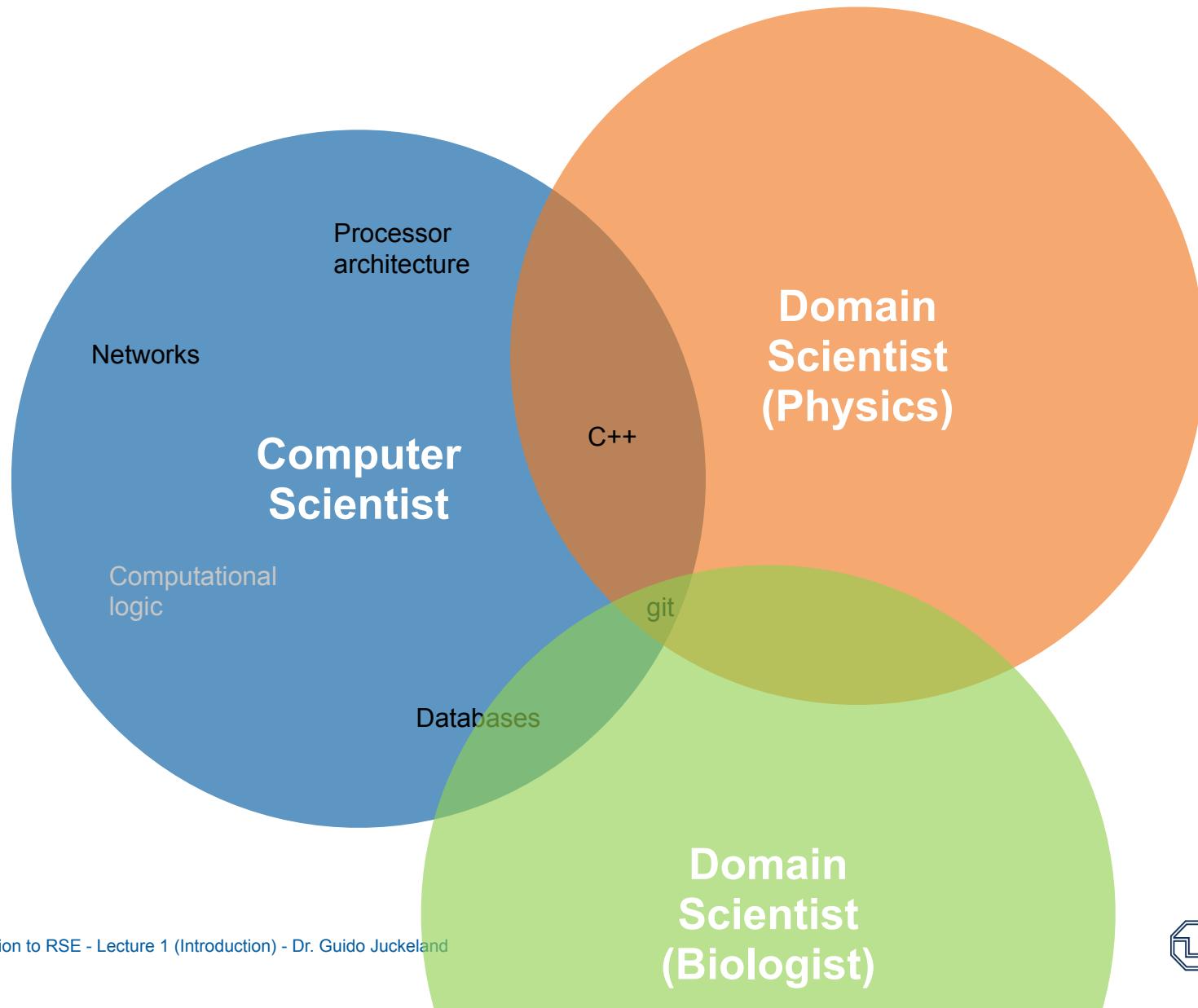
**x86 processor ...**

Figure: <https://doi.org/10.7717/peerjcs.158/fig-1> referenced by K. Hinsen, "Dealing With Software Collapse," in *Computing in Science & Engineering*, vol. 21, no. 3, pp. 104-108, 1 May-June 2019, doi: 10.1109/MCSE.2019.2900945.

# Who does research software engineering?



# Who does research software engineering?



# A short look into history



# Historic origins of the term “RSE”

- Term mentioned first 2012 at <https://www.software.ac.uk/cw12>
- In 2010 newly founded institution in UK: Software Sustainability Institute (SSI)
  - Expected a lot of software engineering work
- Main challenge:
  - A lot of academic work revolves around software, but no recognition
  - (no difference if you do good or bad software)
  - However, good software → sustainable
- Communities followed
  1. uk-RSE
  2. First conferences
  3. Then more worldwide groups: de-RSE, us-RSE

<https://www.software.ac.uk/blog/2016-08-17-not-so-brief-history-research-software-engineers-0>

# A new career path: RSE

**HZDR**  
HELMHOLTZ ZENTRUM  
DRESDEN ROSENDORF

[HZDR](#) [Institutes](#) [Research](#) [Press & News](#) [Career](#) [Technology Transfer](#)

Search  Custom search 

 Deutsch

[Institutes](#) [HZDR](#) [Institutes](#) [Department of Information Services and Computing](#) [Departments](#) Computational Science

## Computational Science

### Mission statement

We are the method-mentoring partner of all scientific institutions in their efforts of digitalizing in their scientific areas. We provide expertise and develop solutions for managing scientific data, research software engineering as well as artificial intelligence. We also develop and maintain software packages for enabling easier access to compute resources for all sciences. In all things we do, we utilize the software engineering methods we advocate and always work together with a partner/customer outside of our department.

### Working groups of the department

 **The Working Group "Data Management and HPC"**  
We develop a technical

 **HIFIS - Helmholtz Federated IT Services**  
HIFIS aims to ensure an excellent information environment for outstanding research in all

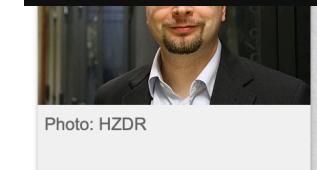
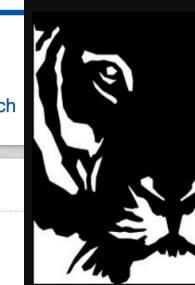


Photo: HZDR

Dr.  
He  
g.ju  
Pho

## Princeton Research Computing

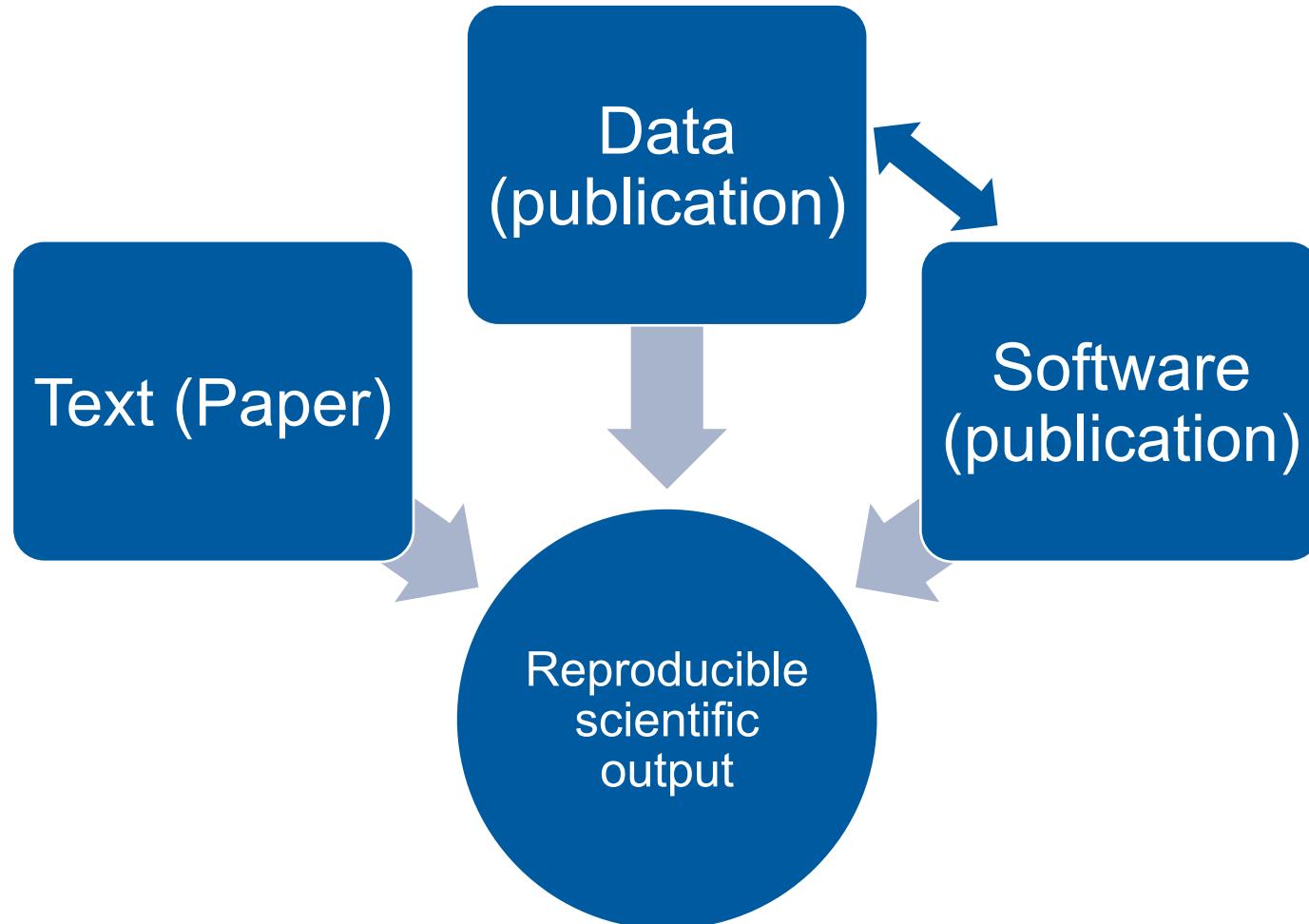
A R D C

## Australian Research Data Commons

# Software as part of the scientific environment

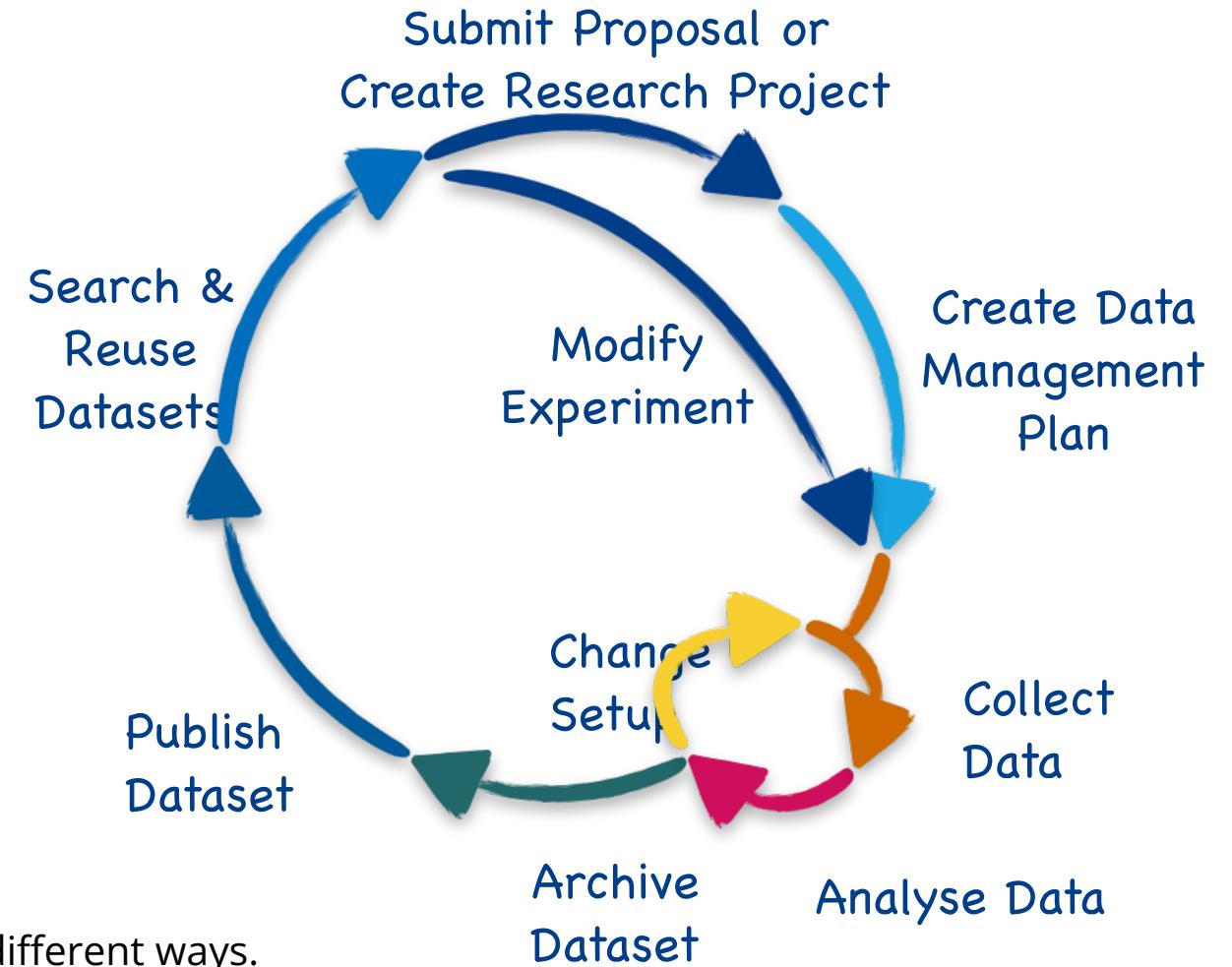


# Scientific output

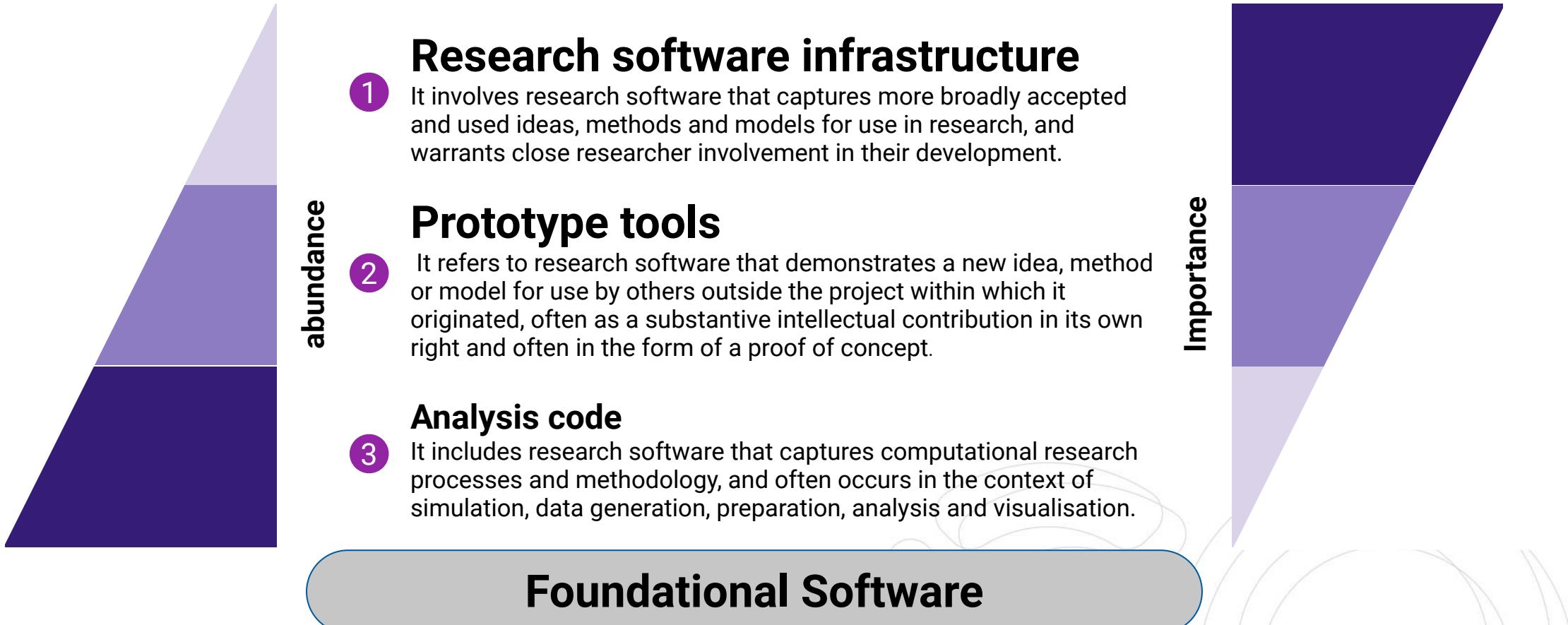


# Scientific data and project lifecycle at HZDR

- In a typical project lifecycle we have multiple kinds of systems and datasets or data products:
  - Proposal (Title, Authors, ...),
  - Data Management Plan (Datasets, ...),
  - Documentation (Experiment specific, ...),
  - Workflows (Source Code, Files, ...),
  - Source Code (Authors, License, ...),
  - Instrument and protocol information,
  - Sample or test object,
  - Environment,
  - Raw data from the experiment,
  - Post-processed Data and
  - Licenses.
- All of these datasets are referred to each other in different ways.



Not all software has the same level of importance ...



# Summary

# Key messages

- You don't need to be a "proper software engineer" to produce research software  
Bast, R. (2023, August 12). Research software engineering for HPC. Zenodo. <https://doi.org/10.5281/zenodo.8242055>
- Highly interdisciplinary work
- Know your audience/target group!
- They need to be able to continue without you!
- **“Developers are drawn to complexity like moths to a flame, often with the same outcome.”**  
(from Neil Ford. The Productive Programmer, ISBN: 0-596-51978-8)

# Most important RSE topics

How can you make others better

- Version control
- Documentation
- Reproducibility and containers
- Building code with CMake (HPC-specific part)
- Automated testing
- Sharing and reusing

Also offering a “Praktikum” (practical lab), every Monday 13:00-14:30 as a separate course to take in addition if you like.

# How this class will run?

Week	Date	Lecture	Exercise	Lab Assignment
1	7.4.	<i>History of RSE / RSE Qualifications</i>	-	<i>Introduction</i>
2	14.4.	-	<i>Shell scripts</i>	<i>Find your topic</i>
3	21.4.	(Easter Monday)		
4	28.4.	<i>Developing in Teams</i>	-	<i>Fix code style, plan data analysis</i>
5	5.5.	-	<i>Version Control, GitLab Workflows</i>	<i>Setup the basic project structure, Fundamental implementation</i>
6	12.5.	<i>Licensing (interactive lecture)</i>		<i>Add metadata</i>
7	19.5.	<i>CI/CD/Testing</i>	-	<i>Project submission</i>
8	26.5.	-	<i>CI/CD/Testing</i>	<i>Build Teams</i>
9	2.6.	<i>Build systems</i>	-	<i>Find topic, initial design</i>
10	9.6.	(Pentecost break)		
11	16.6.	-	<i>Build systems</i>	<i>Add testing, CI/CD</i>
12	23.6.	<i>Software publication</i>	-	<i>Improve and release</i>
13	30.6.	-	<i>Software Publication</i>	<i>Add build tools, metadata</i>
14	7.7.	<i>Case Studies</i>	-	<i>Improve project</i>
15	14.7.	-	<i>Reproducing results of others</i>	<i>Project submission</i>

# Homework to prepare for first lab

## Working with the shell / shell scripts

- Please work through the material at <https://swcarpentry.github.io/shell-novice/>
- Make sure that you are familiar with automating repetitive tasks using the shell