SOFTWARE AS A FIRST-CLASS CITIZEN IN RESEARCH

LEYLA GARCIA, MICHELLE BARKER, NEIL CHUE HONG, FOTIS PSOMOPOULOS, JENNIFER HARROW, DANIEL S. KATZ, MATEUSZ KUZAK, PAULA MARTÍNEZ, ALLEGRA VIA

LJGARCIA@ZBMED.DE

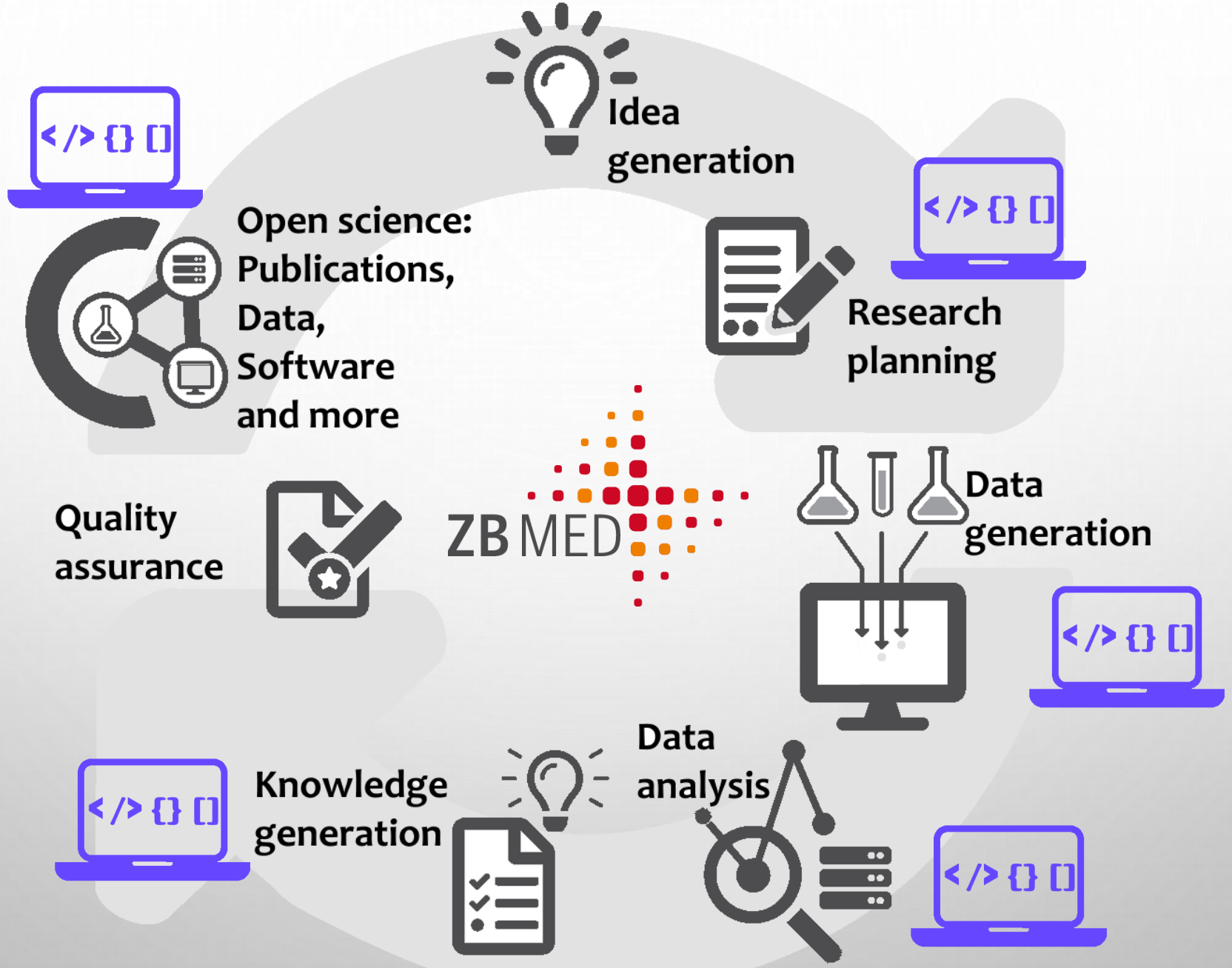
Information Centre for Life Sciences

ZB MED

www.zbmed.de

INFORMATION. KNOWLEDGE. LIFE.

SOFTWARE IN RESEARCH



RESEARCH SOFTWARE IS NOT (JUST) DATA

Why do we need FAIR principles and software management plans when they already exist for data?

- Research software is not (just) data (katz et al., 2016; lamprecht et al., 2019)

Similarities	In between	Differences
• Not commonly cited • All FAIR metadata principles apply • Multiple versions can exist	• Can be built on top of (but dependencies on software are more complex) • Depend on hardware and software (data for display and production, software for that + execution) • Licenses are different (data is not a creative work, software is)	• Software is more volatile and quickly changing, it is "alive" (evolves and requires maintenance) • Reuse comes in different flavors (re-run/execute, reuse, repeat, reproduce, extend) • Can be connected via workflows

FAIR FOR RESEARCH SOFTWARE

FINDABILITY	
F1	(Meta)data are assigned a globally unique and persistent identifier
F2	Data are described with rich metadata
F3	Metadata clearly and explicitly include the identifier of the data it describes
F4	(Meta)data are registered or indexed in a searchable resource

ACCESSIBILITY	
A1	(Meta)data are retrievable by their identifier using a standardized communications protocol
A1.1	The protocol is open, free, and universally implementable
A1.2	The protocol allows for an authentication and authorization procedure, where necessary
A2	Metadata are accessible, even when the data are no longer available

SOFTWARE CASE

- Mainly applicable to metadata
- Software identification → intrinsic and extrinsic
- Specialized registries → tailored to software
- In addition → versions and releases

- Mostly remains the same

FAIR FOR RESEARCH SOFTWARE

INTEROPERABILITY	
I1	(Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation
I2	(Meta)data use vocabularies that follow FAIR principles
I3	(Meta)data include qualified references to other (meta)data

REUSABILITY	
R1	(Meta)data are richly described with a plurality of accurate and relevant attributes
R1.1	(Meta)data are released with a clear and accessible data usage license
R1.2	(Meta)data are associated with detailed provenance
R1.3	(Meta)data meet domain-relevant community standards

SOFTWARE CASE

- Vocabularies work well for software metadata
- Metadata should target machine readability and data exchange
- And still, what is interoperable in software? Are we talking about workflows, containers or so?

- Software dependencies and their licenses should be covered
- We also need software documentation (including use case examples)
- Re-run, reuse, repeat, what are we talking about?

JOINT EFFORT → WORKING GROUP

- A DOCUMENT DEVELOPED WITH COMMUNITY SUPPORT DEFINING FAIR PRINCIPLES FOR RESEARCH SOFTWARE
- A DOCUMENT PROVIDING GUIDELINES ON HOW TO APPLY THE FAIR PRINCIPLES FOR RESEARCH SOFTWARE (BASED ON EXISTING FRAMEWORKS)
- A DOCUMENT SUMMARIZING THE DEFINITION OF THE FAIR PRINCIPLES FOR RESEARCH SOFTWARE, IMPLEMENTATION GUIDELINES AND ADOPTION EXAMPLES.

<ReSA>
Research Software Alliance

FORCE11
The Future of Research Communications and e-Scholarship

RDA
RESEARCH DATA ALLIANCE

SOFTWARE MANAGEMENT PLAN

- SIMILARLY TO DATA MANAGEMENT PLAN, AN SMP IS AN AWARENESS TOOL
  - Think in advance about the software that will be developed
  - The SMP questions help you think about most important parts
  - Think about roles and responsibilities in software project
  - Use it as a guide for everyone involved in the project

- MOTIVATION: UNIFIED APPROACH TO SOFTWARE DEVELOPMENT AND MANAGEMENT IN ELIXIR
  - Part of the tools platform: software best practices group

A SMP FOR LIFE SCIENCES

Section	Questions and options
Basic	Name, location and license
Documentation	• Documentation type → user/developer oriented, readme, release notes, comments • Purpose of the documentation • Sections of the documentation → testing, using, building, deploying, installing
Testing	• Type of testing → Unit, Integration, Regression, End-to-end • Testing methodology → Continuous Integration, Bug-Driven testing • Examples → parameters, input and output
Interoperability	• Input and output formatting and standards used
Reproducibility	• Versioning and version control → Git, mercurial, subversion • Version and releases
Recognition	• Citation information and corresponding metadata (including ORCIDs) • PIDs for releases

LINKED OPEN DATA ROLE

LOD	FAIR for Software	SMPs
☆ Data is available on the Web, in whatever format	Findability → Generic search engines Accessibility → Software and metadata	Basic → software location
☆☆ Available as machine-readable structured data, (i.e., not a scanned image).	Interoperability → metadata	Metadata behind the plan → not currently covered but part of future work
☆☆☆ Available in a non-proprietary format	Interoperability → Software	Interoperability → input and output data and standards Recognition → Citation metadata
☆☆☆☆ Published using open standards from the W3C	Findability → PIDs Interoperability → Metadata	Interoperability → input and output data and standards Recognition → Citation metadata
☆☆☆☆☆ All of the above and links to other Linked Open Data	Interoperability → Software (dependencies) and metadata (meaningful links to others)	Metadata behind the plan → not currently covered but part of future work

https://link.springer.com/chapter/10.1007/978-3-319-98192-5_60

Thank you all!