



On the Road to Research Software Sustainability

Mateusz Kuzak

1 February 2020, FOSSDEM, Brussels

Mateusz Kuzak

Community Manager @ the Netherlands eScience Center
ELIXIR Software Development Best Practices WG co-lead

 @matkuzak

 mkuzak

 m.kuzak@esciencecenter.nl



EMBL (June 2013)



UK (Sept 2013)



Sweden (Sept 2013)



Switzerland (Oct 2013)



Czech Republic (Nov 2013)



Estonia (Dec 2013)



Norway (Jan 2014)



Netherlands (Feb 2014)



Denmark (Mar 2014)



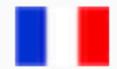
Israel (May 2014)



Portugal (July 2014)



Finland (Sept 2014)



France (Oct 2015 as Provisional Member)



Belgium (Nov 2015)



Italy (Jan 2016)



Slovenia (Feb 2016)



Luxembourg (July 2016)



Ireland (July 2016)



Germany (Aug 2016)



Hungary (Jan 2017)



Spain (Oct 2017)



Greece (Sept 2018)



Cyprus



- **Data**

Sustain core data resources

- **Tools**

Services & connectors to drive access and exploitation

- **Compute**

Access, Exchange & Compute on sensitive data

- **Interoperability**

Integration and interoperability of data and services.

- **Training**

Professional skills for managing and exploiting data





ELIXIR should adopt a policy that **encourages the Nodes to release their software under open source licenses** while respecting existing licensing restrictions and institutional policies. It is widely recognized that **transparent software development from day one improves the quality of the code** and associated documentation through community evaluation.

doi: 10.12688/f1000research.9206.1

Check for updates

OPINION ARTICLE

Top 10 metrics for life science software good practices [version 1; referees: 2 approved]

Haydee Artaza ^{1*}, Neil Chue Hong^{2*}, Manuel Corpas^{1*}, Angel Corpuz^{3*}, Rob Hooft ^{4*},
Rafael C. Jiménez ^{5*}, Brane Leskošek^{6*}, Brett G. Olivier^{7*}, Jan Stourac^{8*}, Radka Svobodová
Vařeková^{9*}, Thomas Van Parys^{10*}, Daniel Vaughan^{11*}

* Equal contributors

[Author details](#)

METRICS

7959

VIEWS

399

DOWNLOADS

[Get PDF](#)

Open Peer Review

Referee Status:

Version(s)	Invited Referees	
	1	2
Version 1 published 16 Aug 2016	 read report	 read report



version control, discoverability,
continuous Integration, testing, standards,
code review, documentation



- What if I write crappy code that nobody likes?
- Will people judge me for the code I write?
- What if someone finds a bug in my code?
- What if I get scooped?
- What are the challenges of Open Source Development?



DOI: 10.12688/f1000research.11407.1

Check for updates

OPINION ARTICLE

Four simple recommendations to encourage best practices in research software [version 1; referees: 3 approved]

[Rafael C. Jiménez](#) ¹, [Mateusz Kuzak](#)², [Monther Alhamdoosh](#) ³, [Michelle Barker](#)⁴, [Bérénice Batut](#) ⁵, [Mikael Borg](#)⁶, [Salvador Capella-Gutierrez](#) ⁷, [Neil Chue Hong](#)⁸, [Martin Cook](#)¹, [Manuel Corpas](#) ⁹, [Madison Flannery](#)¹⁰, [Leyla Garcia](#)¹¹, [Josep Ll. Gelpí](#)^{12,13}, [Simon Gladman](#)¹⁰, [Carole Goble](#)¹⁴, [Montserrat González Ferreiro](#)¹¹, [Alejandra Gonzalez-Beltran](#) ¹⁵, [Philippa C. Griffin](#)¹⁰, [Björn Grüning](#) ⁵, [Jonas Hagberg](#) ⁶, [Petr Holub](#)¹⁶, [Rob Hooft](#) ¹⁷, [Jon Ison](#)¹⁸, [Daniel S. Katz](#) ¹⁹⁻²², [Brane Leskošek](#)²³, [Federico López Gómez](#) ¹, [Luis J. Oliveira](#)²⁴, [David Mellor](#)²⁵, [Rowland Mosbergen](#)²⁶, [Nicola Mulder](#) ²⁷, [Yasset Perez-Riverol](#) ¹¹, [Robert Pergl](#)²⁸, [Horst Pichler](#)²⁹, [Bernard Pope](#)¹⁰, [Ferran Sanz](#)³⁰, [Maria V. Schneider](#)¹⁰, [Victoria Stodden](#)²⁰, [Radosław Suchecki](#)³¹, [Radka Svobodová Vařeková](#)^{32,33}, [Harry-Anton Talvik](#)³⁴, [Ilian Todorov](#)³⁵, [Andrew Treloar](#)³⁶, [Sonika Tyagi](#)^{10,37}, [Maarten van Gompel](#)³⁸, [Daniel Vaughan](#)¹¹, [Allegra Via](#)³⁹, [Xiaochuan Wang](#)⁴⁰, [Nathan S. Watson-Haigh](#)³¹, [Steve Crouch](#)⁴¹

METRICS

2887
VIEWS

567
DOWNLOADS

Get PDF

Get XML

Cite

Export

Open Peer Review

Referee Status:

Version(s)	Invited Referees		
	1	2	3
Version 1 published 13 Jun 2017	read report	read report	read report

1 [Roberto Di Cosmo](#) , Inria, France; Paris Diderot University, France

2 [Greg \(Gregory V.\) Wilson](#), Rangle.io, Canada

3 [Stefanie Betz](#), Karlsruhe Institute of Technology (KIT), Germany



1. OPEN SOURCE YOUR CODE FROM DAY ONE



Make your source code publicly accessible in a version-controlled repository (e.g. github.com, [GitLab](https://gitlab.com) and bitbucket.org) and increase reproducibility, reusability and collaboration.



2. MAKE YOUR SOFTWARE DISCOVERABLE



Register your software metadata in a popular community registry (e.g. bio.tools) and increase your project's visibility.



3. MIND THE LICENSE



Adopt a license that specifies how others can use and distribute your software. Ensure that the software fits with the license of third-party dependencies.



4. DEFINE RESPONSIBILITIES



Let people know how they can contribute to your project and contact you.



<https://softdev4research.github.io/4OSS-lesson/>



THE
CARPENTRIES

4 Simple recommendations for Open Source Software

The aim of this lesson is to provide practical suggestions that contribute to making research software and its source code more discoverable, reusable and transparent. After the introduction, the following episodes of this lesson are structured in the form of one episode per recommendation. Hence the name four open source software recommendations.

Note: This lesson materials are being developed in the open and are in current improvement.

☀ Prerequisites

It is recommended that participants have some familiarity with Github, to create a public repository. Follow the [Setup](#) for instructions or partner with someone who can help you work on this part.

Schedule

	Setup	Get ready, create a repository and create accounts if needed
00:00	1. Introduction	Why are best practices necessary in research software? How Open Source can help with better quality of software?
00:10	2. Make source code publicly accessible from day one	What are the benefits of making my software project public from the beginning? How do I make my project publicly accessible? What resources are available to help me document my software? What are the best practices in open software development? How do I publish my open source software?
00:10	3. Adopt a licence and comply with the licence of third-party dependencies	What a licence does? What is an open source licence? What is the importance of your licence for third-party dependencies?
00:10	4. Define clear and transparent contribution, governance and communication processes	How does someone start contributing to my project? What do I need to consider about project design and governance? How do people communicate within the project?
01:25	5. Make software easy to discover by providing software metadata via a popular community registry	Why are metadata important in research software? What are good metadata? Which are the most commonly used platforms for registering research software data?
03:40	Finish	

The actual schedule may vary slightly depending on the topics and exercises chosen by the instructor.

<https://github.com/NLeSC/awesome-research-software-registries>

Awesome Research Software Registries

A list of research software registries (also known as catalog, index, warehouse, repository, hub, platform, and other terms).

The Turing Way

- 1. Introduction
- 2. Reproducibility
- 3. Open Research
- 4. Version Control
- 5. Collaborating on GitHub/GitLab

← TOGGLE SIDEBAR

Welcome to the Turing Way

The Turing Way is a lightly opinionated guide to reproducible data science.

Our goal is to provide all the information that researchers need at the start of their projects to ensure that they are easy to reproduce at the end.

This also means making sure PhD students, postdocs, PIs, and funding teams know which parts of the “responsibility of reproducibility” they can affect, and what they should do to nudge data science to being more efficient, effective, and understandable.

ON THIS PAGE

- A LITTLE MORE BACKGROUND
- THE BOOK ITSELF
- THE TURING WAY COMMUNITY
- CITING THE TURING WAY



<https://the-turing-way.netlify.com>

 @matkuzak

FIVE RECOMMENDATIONS FOR FAIR SOFTWARE

ENDORSE

LET'S GO! →



<https://fair-software.eu>

