4 Simple Recommendations for OSS


https://softdev4research.github.io/recommendations/

“Four Simple Recommendations to Encourage Best Practices in Research Software”
doi:10.12688/f1000research.11407.1
Software development and open source best practices in ELIXIR

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.
Researchers are afraid of putting their code out there.

- What if I write crap 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 Software Development?
The Four Recommendations

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

- Rafael C. Jiménez
- Mateusz Kuzak
- Monther Alhamdoosh
- Michelle Barker
- Bérénice Batut
- Mikaël Borg
- Salvador Capella-Gutierrez
- Neil Chue Hong
- Martin Cook
- Manuel Corpas
- Madison Flannery
- Leyla García
- Josep Ll. Gelp
- Simon Gladman
- Carole Goble
- Montserrat González Ferreiro
- Alejandra Gonzalez-Beltran
- Philippa C. Griffin
- Björn Grüning
- Jonas Hagberg
- Petr Holub
- Rob Hoofd
- Jon Ison
- Daniel S. Katz
- Brane Leskošek
- Federico López Gómez
- Luis J. Oliveira
- David Mellor
- Rowland Mosberger
- Nicola Mulder
- Yasset Perez-Riverol
- Robert Pergi
- Horst Pichler
- Bernard Pope
- Ferran Sanz
- Maria V. Schneider
- Victoria Stodden
- Radoslaw Suchecki
- Radka Svobodová Vařeková
- Harry-Anton Talvitie
- Ilian Todorov
- Andrew Treloar
- Sonika Tyagi
- Maarten van Gompel
- Daniel Vaughan
- Allegra Via
- Xiaochuan Wang
- Nathan S. Watson-Haigh
- Steve Crouch

**Open Peer Review**

Referee Status: ✔️ ✔️ ✔️

<table>
<thead>
<tr>
<th>Invited Referees</th>
<th>Version(s)</th>
<th>Published</th>
<th>Invited</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Version 1</td>
<td>13 Jun 2017</td>
<td>✔️ ✔️ ✔️</td>
</tr>
</tbody>
</table>

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
4OSS recommendations

Recommendations to encourage best practices in research software

1. Develop publicly accessible open source code from day one

Start a project as open source from the very first day, in a publicly accessible, version controlled repository (e.g., github.com and bitbucket.org). The longer a project is run in a closed manner, the harder it is to open source it later.

2. Make software easy to discover by providing software metadata via a popular community registry

Facilitate the discoverability of the open source software projects by registering metadata related to the software in a popular community registry, making your source code more discoverable. Metadata might, include information like the source code location, contributors, license, references and how to cite the software.

3. Adopt a license and comply with the licence of third-party dependencies
1. OPEN SOURCE YOUR CODE FROM DAY ONE

Make your source code publicly accessible in a version-controlled repository (e.g. github.com, GitLab 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.
Support Faster Scientific Discoveries with Four Simple Recommendations

1. **OPEN SOURCE YOUR CODE FROM DAY ONE**
   Make your source code publicly accessible in a version-controlled repository (e.g. github.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.

---

ENDORSE:
https://softdev4research.github.io/recommendations/endorse/

---

Read more:
“Four Simple Recommendations to Encourage Best Practices in Research Software”
doi: 10.12688/f1000research.11407.1
The goal of this project is to create open and reusable training materials to teach researchers and developers how to adopt the 4OSS as development practices.
The lesson is being developed in a 4-stage process

1. CarpentryCon: defining lesson learning outcomes, structure and contents

Dublin June 1st
14 participants
In 4 teams
One per each recommendation
The lesson is being developed in a 4-stage process

2. Hackathon in Utrecht produced the first lesson draft

2 days
15 in person participants
3 online participants
The lesson is being developed in a 4-stage process

3. Online lesson refinement through GitHub issues and pull requests (Aug-Oct 2018)
The lesson is being developed in a 4-stage process

4. NETTAB (Oct 2018) satellite meeting to release the first version of the material.

Future:
BioHackathon, Paris, November
Timeline

- Today 3rd Aug
  - 10th Sept: ECCB 11th Tool WS
    - Metadata Episode "ready"
  - 21st Sept: Approve content
  - 24/25 Sept: BBQ
  - 3rd Oct: ISSUE BONANZA
- 22nd Oct: NETLAB
  - DB RUN: TUTORIAL
  - WSSSPE: RELEASE
- 29th Oct: BIOHACKATHON
  - Notebook Binder GitLab
- 12-16th Nov: CLEANUP
<table>
<thead>
<tr>
<th>Schedule</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setup</strong></td>
<td>Get ready, create a repository and create accounts if needed</td>
</tr>
<tr>
<td>00:00</td>
<td>1. <strong>Introduction</strong>&lt;br&gt;Why are best practices necessary in research software?&lt;br&gt;How Open Source can help with better quality of software?</td>
</tr>
<tr>
<td>00:10</td>
<td>2. <strong>Make source code publicly accessible from day one</strong>&lt;br&gt;What are the benefits of making my software project public from the beginning?&lt;br&gt;How do I make my project publicly accessible?&lt;br&gt;What resources are available to help me document my software?&lt;br&gt;What are the best practices in open software development?&lt;br&gt;How do I publish my open source software?</td>
</tr>
<tr>
<td>00:10</td>
<td>3. <strong>Adopt a licence and comply with the licence of third-party dependencies</strong>&lt;br&gt;What a licence does?&lt;br&gt;What is an open source licence?&lt;br&gt;What is the importance of your licence for third-party dependencies?</td>
</tr>
<tr>
<td>00:10</td>
<td>4. <strong>Define clear and transparent contribution, governance and communication processes</strong>&lt;br&gt;How does someone start contributing to my project?&lt;br&gt;What do I need to consider about project design and governance?&lt;br&gt;How do people communicate within the project?</td>
</tr>
<tr>
<td>01:25</td>
<td>5. <strong>Make software easy to discover by providing software metadata via a popular community registry</strong>&lt;br&gt;Why are metadata important in research software?&lt;br&gt;What are good metadata?&lt;br&gt;Which are the most commonly used platforms for registering research software data?</td>
</tr>
<tr>
<td>03:40</td>
<td>Finish</td>
</tr>
</tbody>
</table>
Get involved, talk to us!

Mateusz Kuzak
@matkuzak
mateusz.kuzak@dtls.nl

Paula Martinez
@orchid00
orchid00@gmail.com

Allegra Via
@elixir_ita
allegra.via@gmail.com

Fotis Psomopoulos
@fopsom
fpsom@issel.ee.auth.gr

/SoftDev4Research/4OSS-lesson