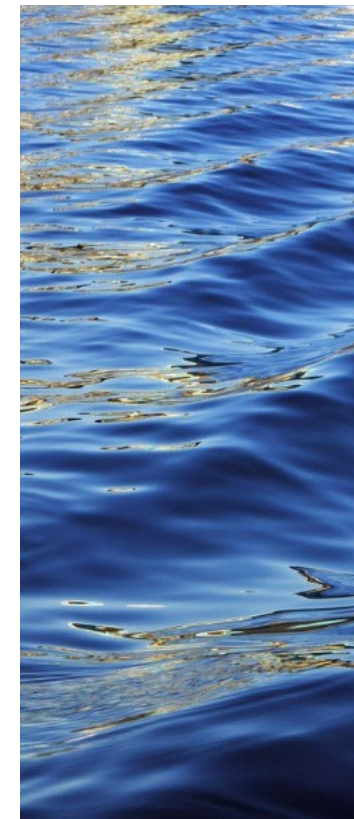




# Τέσσερις καλές πρακτικές για την ανάπτυξη λογισμικού στην Ανοιχτή Επιστήμη



Φώτης Ε. Ψωμόπουλος, Ερευνητής Γ' INEB|EKETA

# Γιατί χρειαζόμαστε καλές πρακτικές;

- Οι επιστήμονες συνήθως αναπτύσσουν το δικό τους λογισμικό
  - Απαιτεί συγκεκριμένες γνώσεις πάνω στο αντικείμενο
  - Μηχανικοί λογισμικού (software engineers) είναι δυσεύρετοι
- Περισσότερο από 90% των ερευνητών είναι αυτοδίδακτοι



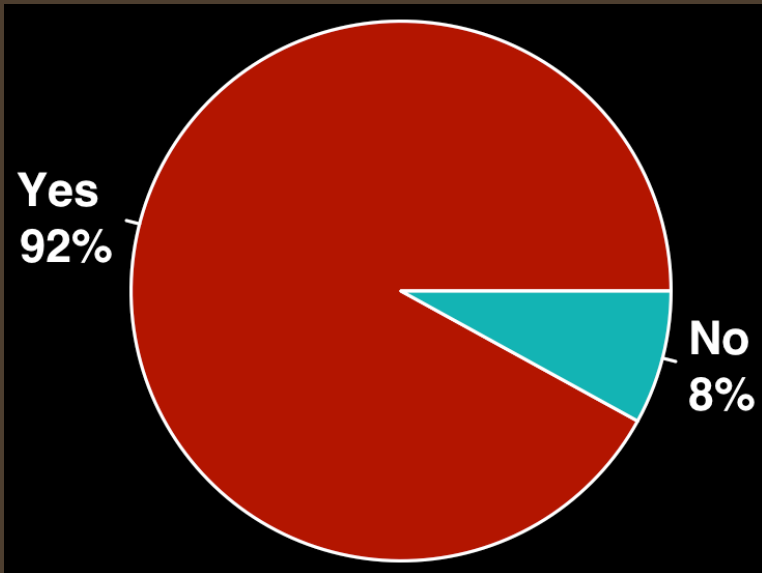
Χαμηλή ποιότητα και βιωσιμότητα του λογισμικού

Greg Wilson. *Best Practices for Scientific Computing*. <https://doi.org/10.1371/journal.pbio.1001745>

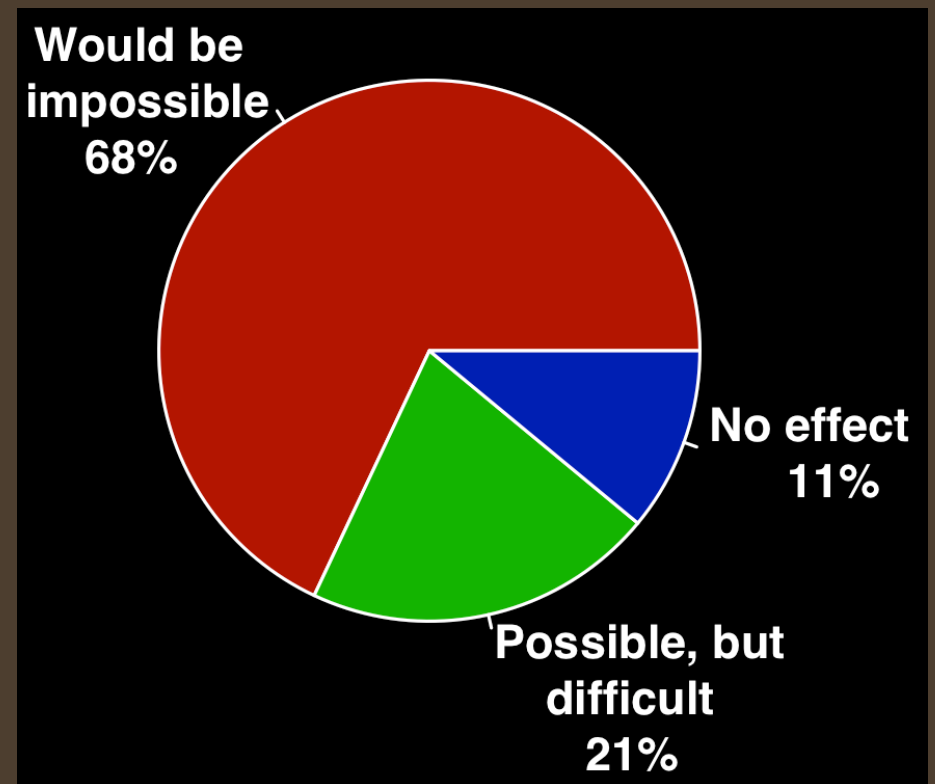
*Slides από την παρουσίαση "Tool platform: Software development best practices" presentation by Rafael C Jimenez, 08/02/2018*

# Η ερευνητική κοινότητα βασίζεται στο λογισμικό

Χρησιμοποιείτε ερευνητικό λογισμικό?



Τι επίδραση θα είχε στην έρευνά σας η απουσία ερευνητικού λογισμικού?



*Survey of researchers from 15 UK Russell Group universities conducted by SSI between August - October 2014. 406 respondents covering representative range of funders, discipline and seniority.*

## ELIXIR Software development best practices group

- Σκοπός
  - Βελτίωση της ποιότητας και της βιωσιμότητας του λογισμικού στις επιστήμες ζωής
  
- Επιμέρους στόχοι
  1. Προώθηση της υιοθέτησης καλών πρακτικών για την ανάπτυξη λογισμικού
  2. Ανάπτυξη συστάσεων για το λογισμικό ανοικτού κώδικα (OSS)
  3. Ανάπτυξη μετρικών για την αξιολόγηση των συστάσεων και των καλών πρακτικών





US National Library of Medicine  
National Institutes of Health

PMC



[Advanced](#) [Journal list](#)

[Journal List](#) > [F1000Res](#) > [v.5; 2016](#) > [PMC5007752.1](#)

Version 1. [F1000Res](#). 2016; 5: ELIXIR-2000.

PMCID: PMC5007752

Published online 2016 Aug 16. doi: [10.12688/f1000research.9206.1](https://doi.org/10.12688/f1000research.9206.1)

## Top 10 metrics for life science software good practices

[Haydee Artaza](#),<sup>#1</sup> [Neil Chue Hong](#),<sup>#2</sup> [Manuel Corpas](#),<sup>#a,1</sup> [Angel Corpuz](#),<sup>#3</sup> [Rob Hooft](#),<sup>#4</sup> [Rafael C. Jimenez](#),<sup>#5</sup> [Brane Leskošek](#),<sup>#6</sup> [Brett G. Olivier](#),<sup>#7</sup> [Jan Stourac](#),<sup>#8</sup> [Radka Svobodová Vařeková](#),<sup>#b,9</sup> [Thomas Van Parys](#),<sup>#10</sup> and [Daniel Vaughan](#)<sup>#11</sup>

[Author information](#) ▶ [Article notes](#) ▶ [Copyright and License information](#) ▶

### Peer Review Summary

Go to:

F1000Res

# Αντιστοίχιση των μετρικών στις καλές πρακτικές

- 17 καλές πρακτικές
- 43 μετρικές
- Ποσοτικές και ποιοτικές
- Προτεραιότητα των 10 μετρικών σε ένα πίνακα αντίκτυπου / προσπάθειας

## 1. **Version control:**

- a. Yes/no?
- b. How many committers?
- c. When was the version control started?
- d. When was the last commit?

## 2. **Code reviews:**

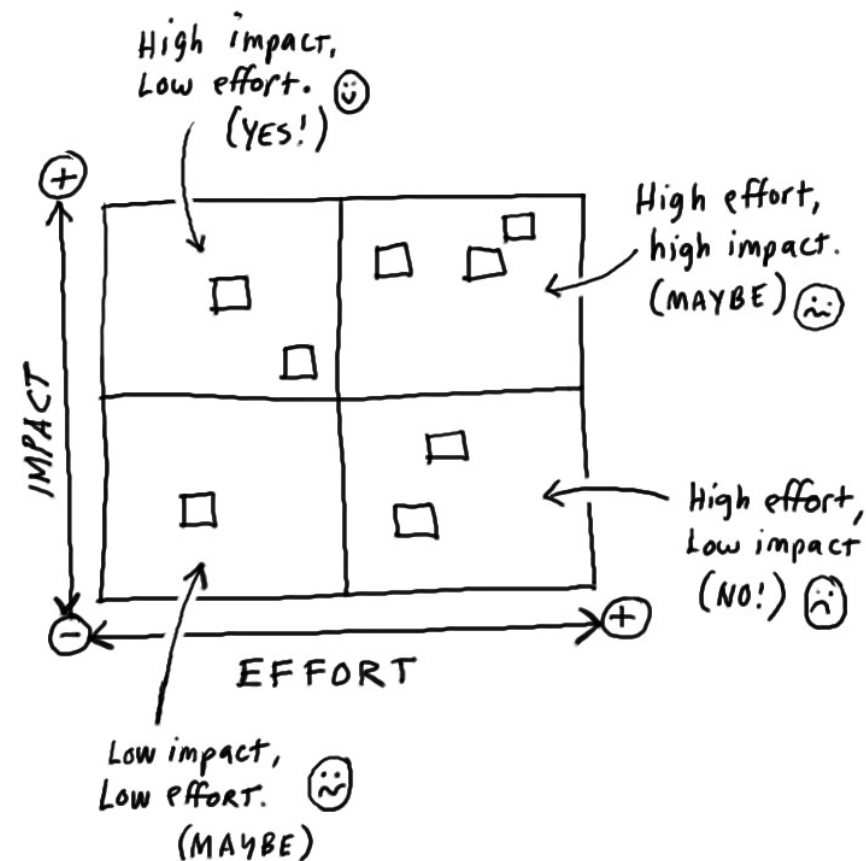
- a. Yes/no?
- b. Star rating based on code description

## 3. **Automated testing:**

- a. Yes/no?
- b. Coverage for unit tests
- c. Yes/no for individual tests:
  - i. Unit tests

# Οι πρώτες 10 μετρικές (από τις 43)

1. Is version control used?
2. Is the software discoverable?
3. Is an automated build system used?
4. Are test data available?
5. Does software contain parts that reimplement existing technology?
6. Is the software compliant with community standards?
7. Are code reviews performed?
8. Is automated testing performed?
9. Is the code documented?
10. How high is the code complexity?



# Για την υποστήριξη της ταχύτερης έρευνας

## 4OSS

πρακτικές



Καλύτερο λογισμικό



Ταχύτερη  
Έρευνα και  
Ανακάλυψη



# Για την υποστήριξη της ταχύτερης έρευνας

## 4OSS

πρακτικές



Μεγάλη προσπάθεια,  
δύσκολο να μετρηθεί

Καλύτερο λογισμικό

↓  
Ταχύτερη  
Έρευνα και  
Ανακάλυψη

# Για την υποστήριξη της ταχύτερης έρευνας





Check for updates

OPINION ARTICLE

## Four simple recommendations to encourage best practices in research software [version 1; referees: awaiting peer review]

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

+ [Author details](#)

+ [Grant information](#)



METRICS

491

VIEWES

62

DOWNLOADS

 [Get PDF](#)

 [Get XML](#)

“ [Cite](#)

 [Export](#)

 [Track](#)

 [Email](#)

 [Share](#)

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.

# 1. Make source code publicly accessible from day one

## Questions

- 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?

## Objectives

1. create a new software project on GitHub
2. list three things that should be included in 'bare minimum' software documentation
3. explain the difference between user- and developer-oriented documentation
4. describe three steps that can be taken to increase the (re)usability of software
5. describe how external resources can be used at the stage of publishing OS software

## 2. Adopt a license

### Questions

- What a license does?
- What is an open source license?
- What is the importance of your license for third-party dependencies?

### Objectives

1. Tell what is a copyright and what a license does
2. tell why is important that a product/code has a license
3. tell what is the importance of third-party dependencies on your product/code
4. Choose a license for your code
5. Add a LICENCE file to a repository

# 3. Define clear and transparent contribution, governance and communication processes

## Questions

- 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?

## Objectives

1. Design the minimum guidelines for people to contribute and engage with the project
2. Define how you would support your contributors
3. Establish roles, expectations and processes around contributions to the project
4. Create communication strategies among people in the project

## 4. Make software easy to discover by providing software metadata

### Questions

- Why are metadata important in research software?
- What are good metadata?
- Which are the most commonly used platforms for registering research software data?

### Objectives

1. Understand the importance of metadata
2. Understand why metadata are necessary for software discoverability
3. Have a clear concept of what good metadata entail
4. Use one of the existing platforms to upload metadata

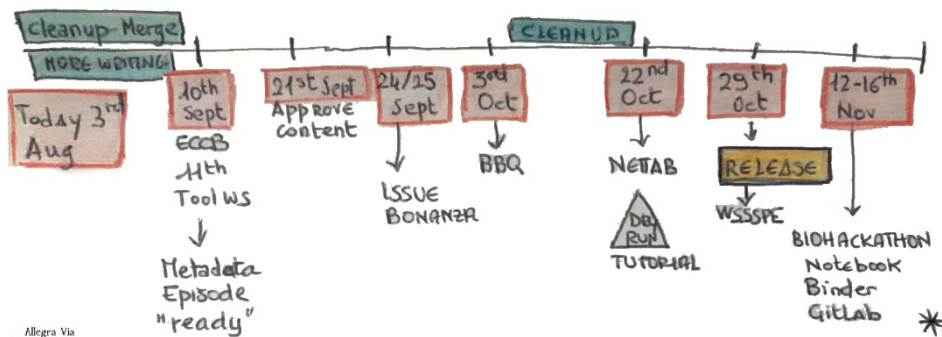
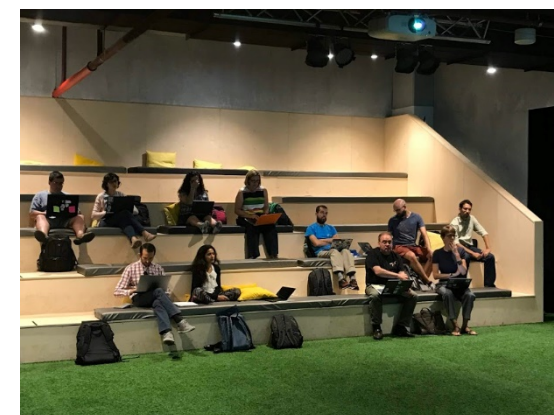


# Ανάπτυξη του εκπαιδευτικού υλικού



Έγινε ένα workshop ώστε να προσδιορισθεί η γενικότερη δομή και το περιεχόμενο που θα πρέπει να καλύπτει το εκπαιδευτικό υλικό.

Hackathon στο DTL / Utrecht, ετοιμάζοντας το υλικό - 2 ημέρες, 4 ενότητες, 25 συμμετέχοντες από όλο τον κόσμο



Τελικό χρονοδιάγραμμα

# Το τελικό υλικό

## From Design to DOI: Good Practices in Research 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

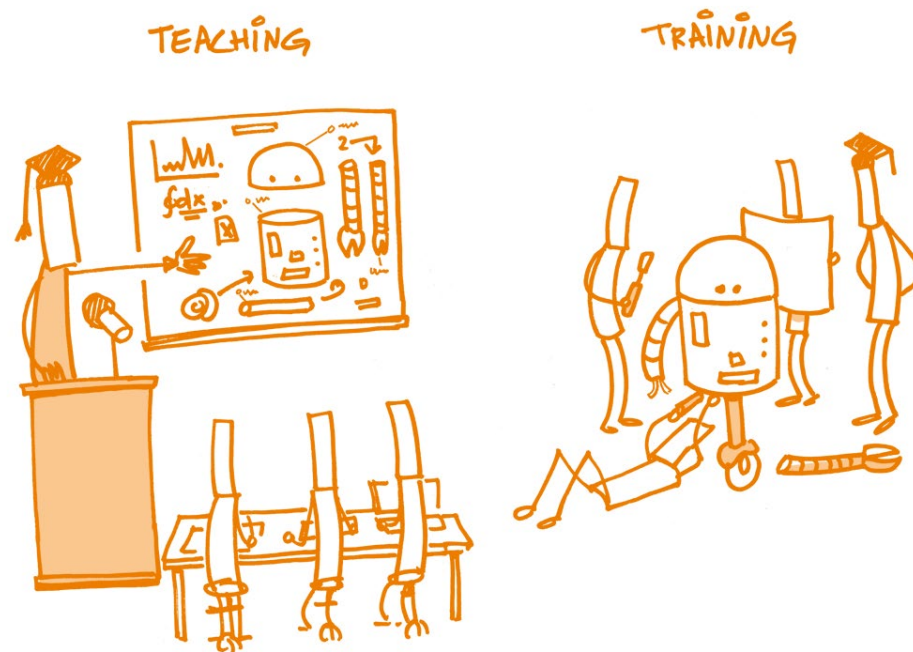
	Setup	Get ready, create a repository and create accounts if needed
00:00	1. Introduction	Why are best practices necessary in research software? How Open Source can help with better quality of software?
00:10	2. Make source code publicly accessible from day one	What are the benefits of making my software 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	



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

# Τα επόμενα βήματα: Εκπαίδευση

- Η κατάρτιση υψηλής ποιότητας και το αντίστοιχο εκπαιδευτικό υλικό αποτελούν τα **θεμέλια** όταν στοχεύουν σε μια πολιτιστική αλλαγή προς την κατεύθυνση της εφαρμογής των αρχών της ανοιχτής επιστήμης.



# Τα επόμενα βήματα: Οδηγίες / Προδιαγραφές

- Χρηματοδοτικοί Φορείς / Οργανισμοί:
  - συνήθως ζητούν από τους αιτούντες να υποβάλουν σχέδιο διαχείρισης δεδομένων
  - δεν παρέχουν οδηγίες σχετικά με τον τρόπο ανάπτυξης ή διαχείρισης του λογισμικού κατά τη διάρκεια του χρηματοδοτούμενου έργου.
  - Στο καλύτερο σενάριο αναφέρουν κάτι της μορφής:
    - "το λογισμικό πρέπει να είναι ελεύθερα διαθέσιμο" ή
    - "το εργαλείο θα είναι ανοικτού κώδικα"
  
- Χρειαζόμαστε:
  - συγκεκριμένες κατευθυντήριες γραμμές λογισμικού που οι οργανισμοί και οι χρηματοδότες θα μπορούσαν να ενσωματώσουν ως μέρος της πολιτικής τους για την ανάπτυξη λογισμικού

# Οδηγίες FAIR

- Κατευθυντήριες γραμμές που:
    - δεν είναι δύσκολο να εφαρμοστούν, αλλά εξακολουθούν να οδηγούν καλές πρακτικές στην ανάπτυξη λογισμικού
    - είναι εύκολο να εκτιμηθούν ώστε να επιτραπεί η αξιολόγηση της συμμόρφωσης του λογισμικού.
1. Ποιο θα πρέπει να είναι το περιεχόμενο αυτών των οδηγιών;
  2. Πώς πρέπει να αξιολογείται το λογισμικό σύμφωνα με τις κατευθυντήριες γραμμές;
  3. Πώς πρέπει να υποστηρίζει η κοινότητα αυτές οι οδηγίες;
  4. Πώς πρέπει να εμπλακούν χρηματοδότες και οργανώσεις για την υιοθέτηση των κατευθυντήριων γραμμών;



# Αντί επιλόγου

Πρέπει να έχουμε συγκεκριμένες κατευθυντήριες γραμμές για το λογισμικό οι οποίες θα μπορούσαν να ενσωματωθούν στην πολιτική των χρηματοδοτικών οργανισμών για την ανάπτυξη λογισμικού.

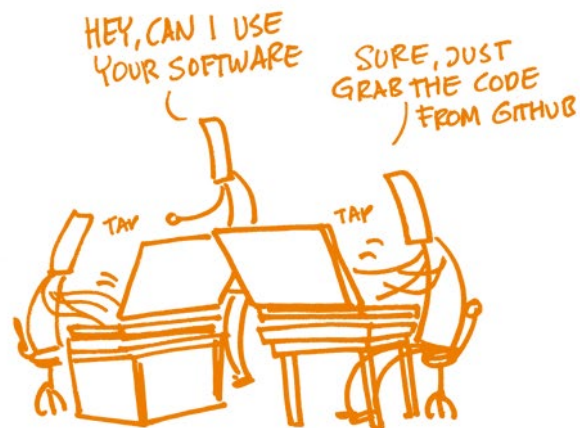
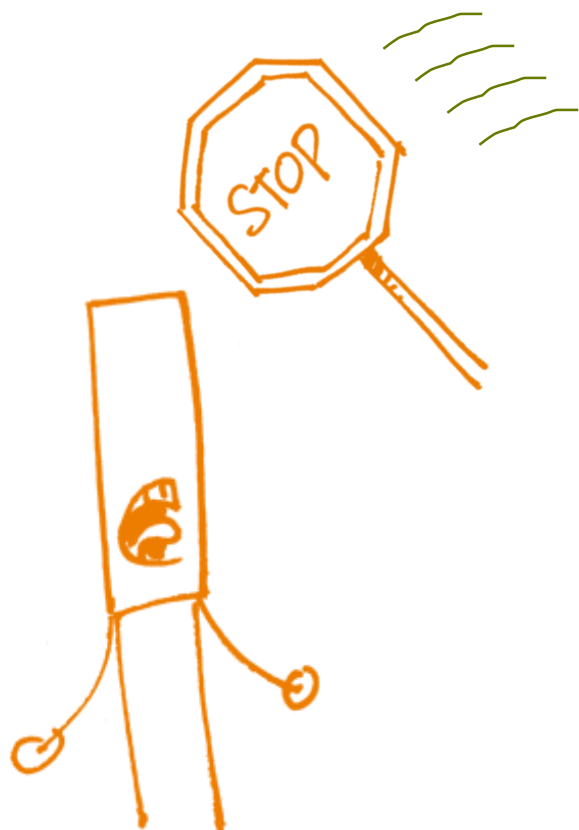
Όσο πιο σύντομα βρούμε την απάντηση σε αυτές τις ερωτήσεις και συμφωνήσουμε σε μια κοινή προσέγγιση, τόσο πιο εύκολο θα είναι να εργαστούμε για τη βελτίωση της ποιότητας και της βιωσιμότητας του ερευνητικού λογισμικού που θα αναπτυχθεί στα πλαίσια νέων ερευνητικών προγραμμάτων.



OPEN SOURCE

VS

CLOSED SOURCE



Ευχαριστώ για την προσοχή σας!



@fopsom

<https://fopsom.github.io/>