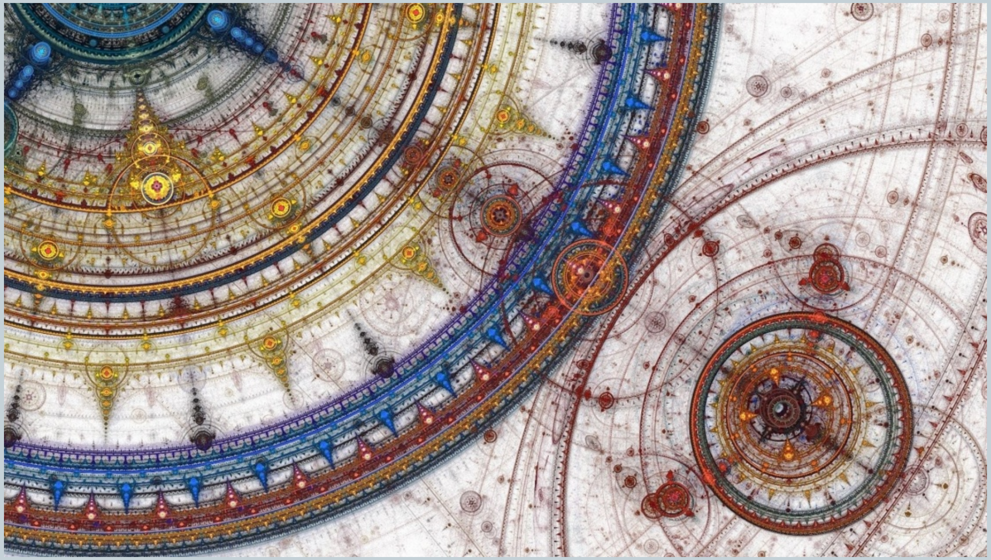
CERTH
CENTRE FOR
RESEARCH & TECHNOLOGY
HELLAS

INAB
INSTITUTE OF APPLIED BIOSCIENCES
ΙΝΣΤΙΤΟΥΤΟ ΕΦΑΡΜΟΣΜΕΝΩΝ ΒΙΟΕΠΙΣΤΗΜΩΝ
CERTH

Four simple recommendations for research software development

FOTIS E. PSOMOPOULOS



Why we need best practices?

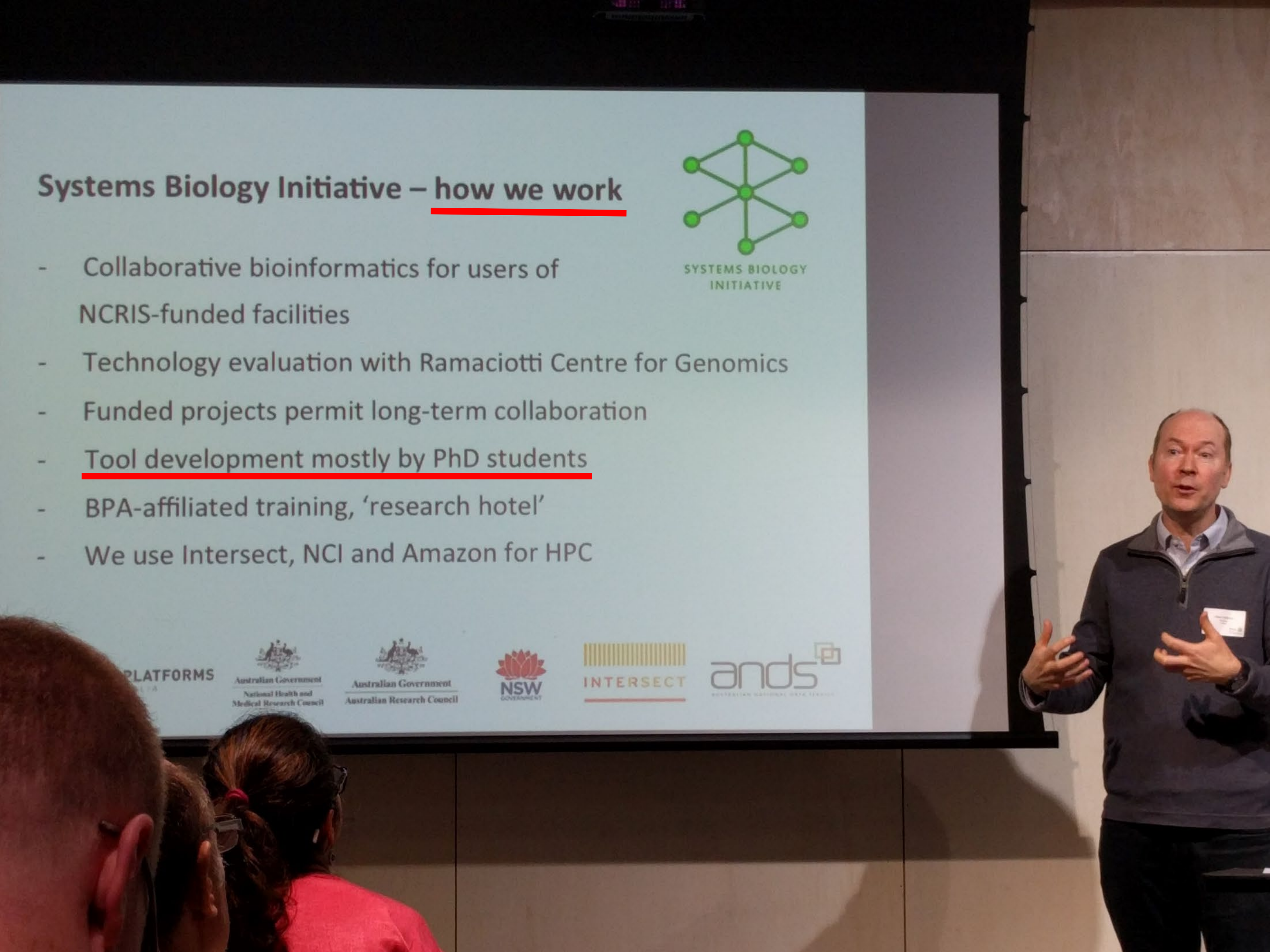
- Scientists typically develop their own software
  - Requires substantial domain-specific knowledge
  - Software engineers are scarce in the field

- 90% or more of scientist are primarily self-taught

Low software quality and sustainability

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

Slide from "Tool platform: Software development best practices" presentation by Rafael C Jimenez, 08/02/2018



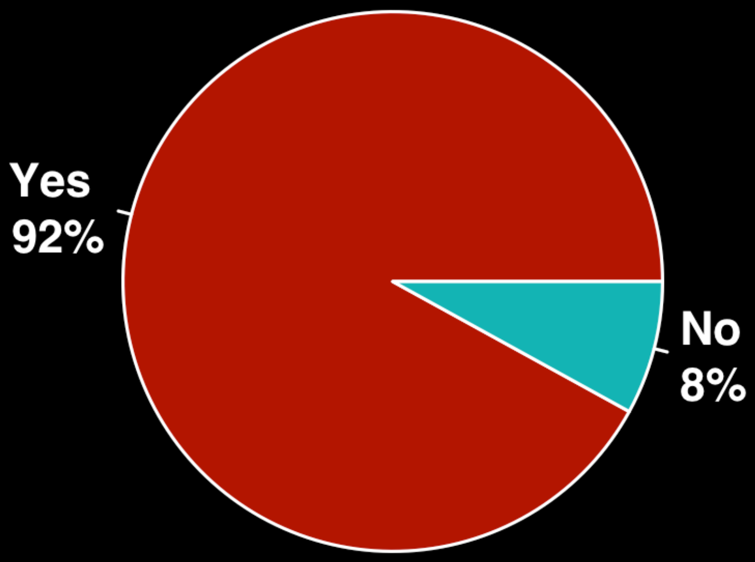
Systems Biology Initiative – how we work

- Collaborative bioinformatics for users of NCRIS-funded facilities
- Technology evaluation with Ramaciotti Centre for Genomics
- Funded projects permit long-term collaboration
- Tool development mostly by PhD students
- BPA-affiliated training, ‘research hotel’
- We use Intersect, NCI and Amazon for HPC

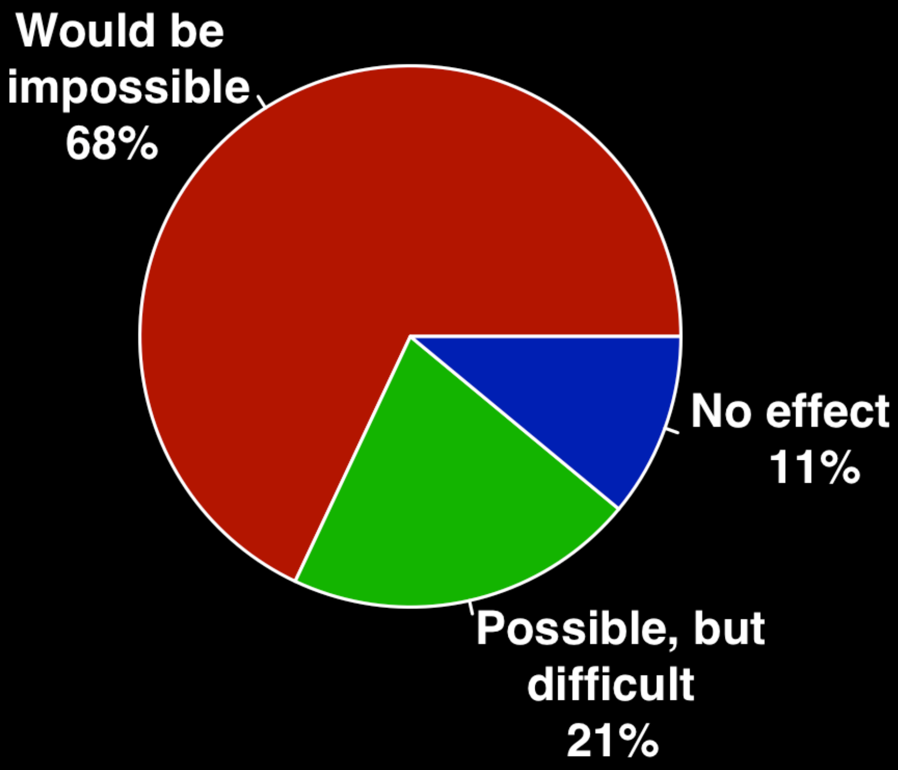
SYSTEMS BIOLOGY INITIATIVE
PLATFORMS
Australian Government National Health and Medical Research Council
Australian Government Australian Research Council
NSW GOVERNMENT
INTERSECT
ands

The research community relies on software

Do you use research software?



What would happen to your research without software



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

- Goal
  - Raise Quality and Sustainability in software development in the life sciences
- Objectives
  1. Promote adoption of best practices on software development
  2. Develop Open Source Software (OSS) recommendations
  3. Develop metrics for the assessment of principles and best practices

Journal List › F1000Res › v.5; 2016 › PMC5007752.1

Version 1. F1000Res. 2016; 5: ELIXIR-2000.
Published online 2016 Aug 16. doi: 10.12688/f1000research.9206.1

PMCID: PMC5007752

Top 10 metrics for life science software good practices

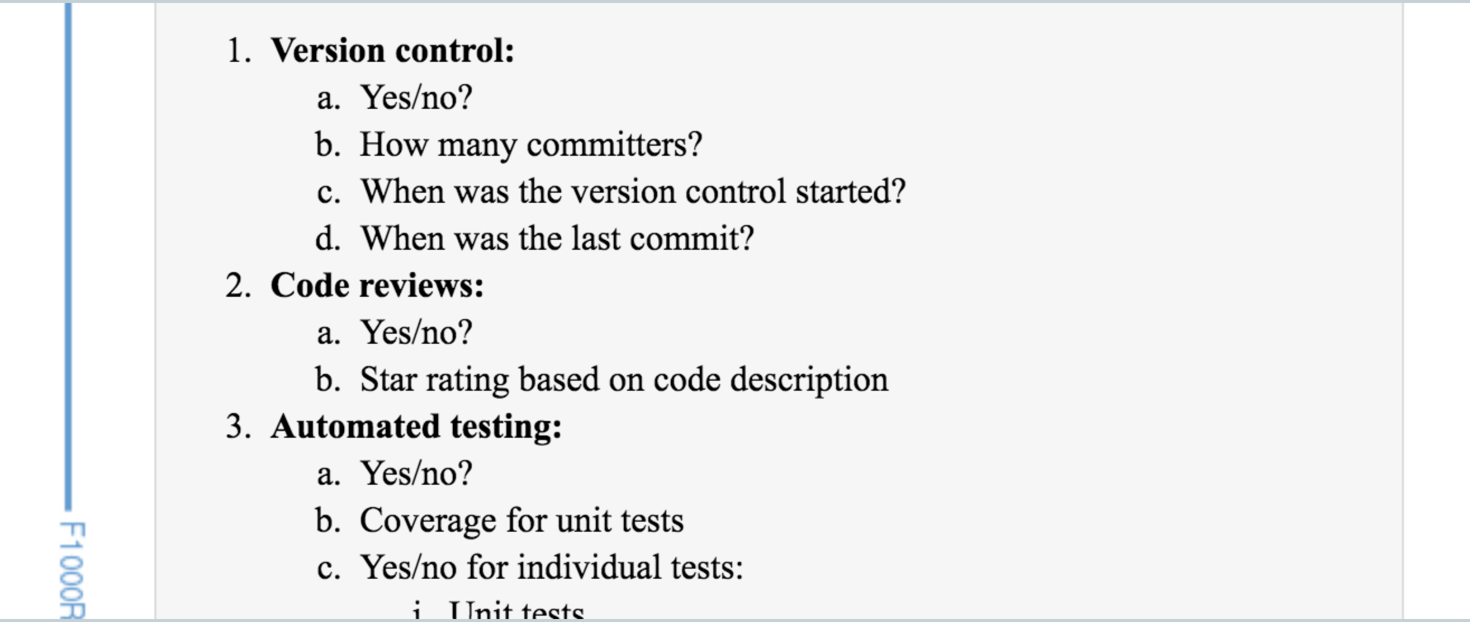
Haydee Artaza,[#1] Neil Chue Hong,[#2] Manuel Corpas,[#a,1] Angel Corpuz,[#3] Rob Hooft,[#4] Rafael C. Jimenez,[#5] Brane Leskošek,[#6] Brett G. Olivier,[#7] Jan Stourac,[#8] Radka Svobodová Vařeková,[#b,9] Thomas Van Parys,[#10] and Daniel Vaughan[#11]

Author information ► Article notes ► Copyright and License information ►

Peer Review Summary

Mapping metrics to good enough practices

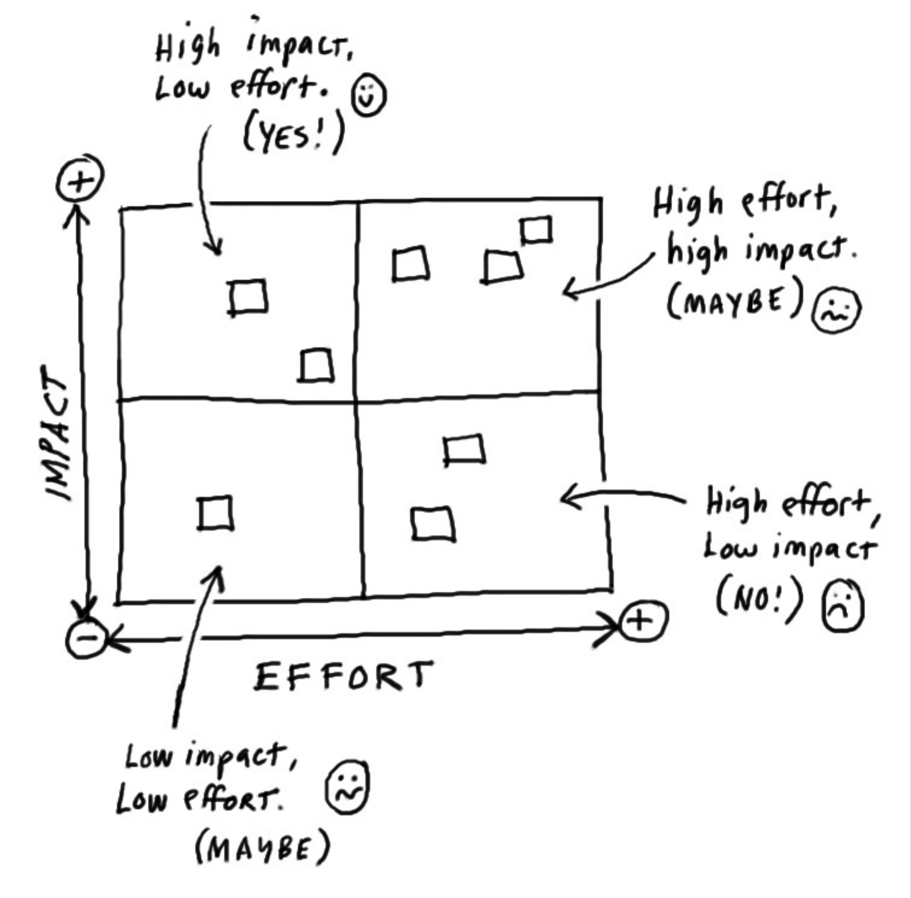
- 17 good enough practices
- 43 metrics
- Quantitative as well as qualitative
- 10 metrics Prioritized by impact/effort matrix



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

Top 10 metrics (from 43)

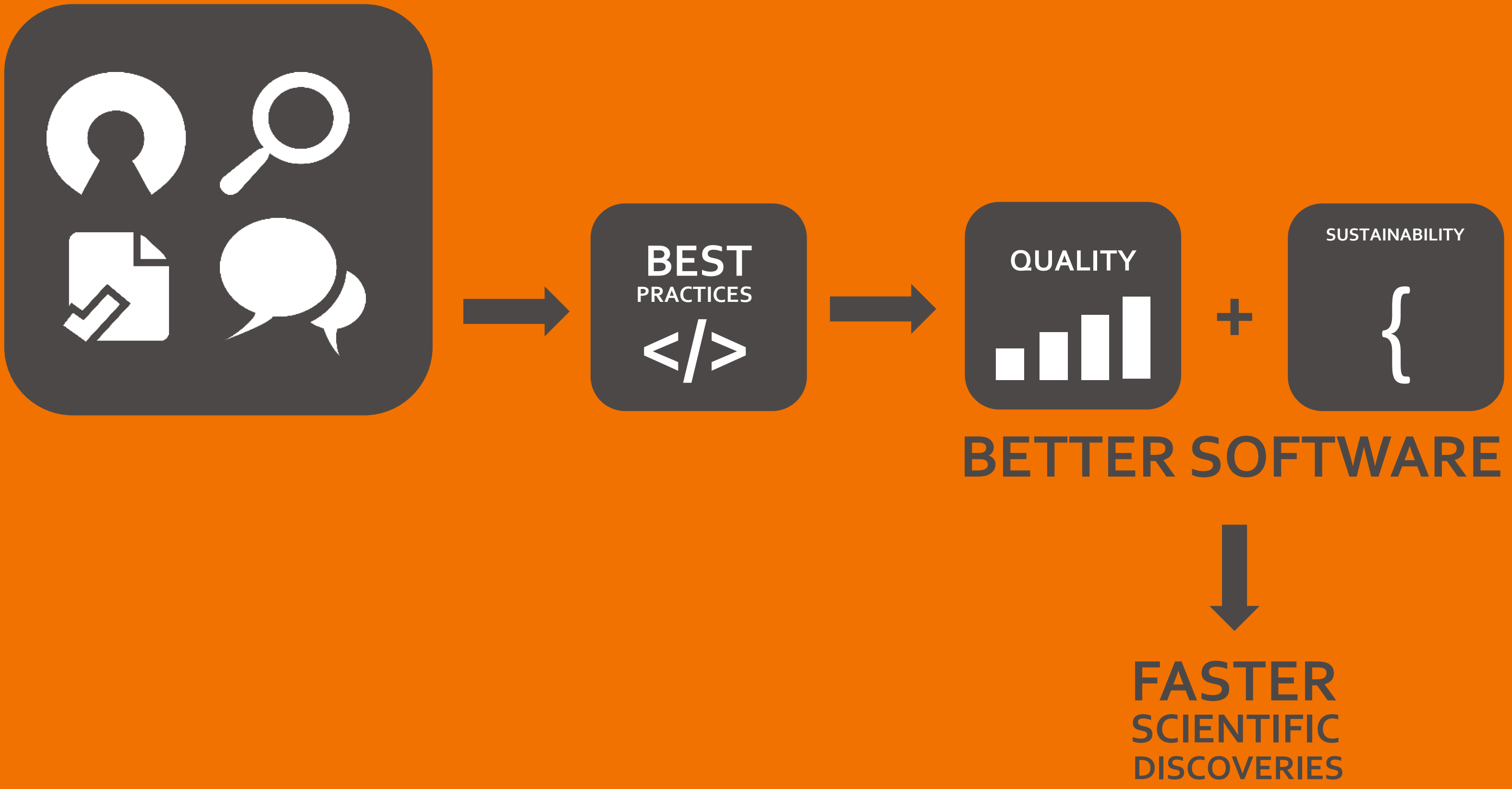
1. Is version control used?
2. Is the software discoverable?
3. Is an automated build system used?
4. Are test data available?
5. Does software contain parts that reimplement existing technology?
6. Is the software compliant with community standards?
7. Are code reviews performed?
8. Is automated testing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



SUPPORT FASTER SCIENTIFIC DISCOVERIES

4OSS

recommendations



BEST PRACTICES

</>

QUALITY

+

SUSTAINABILITY

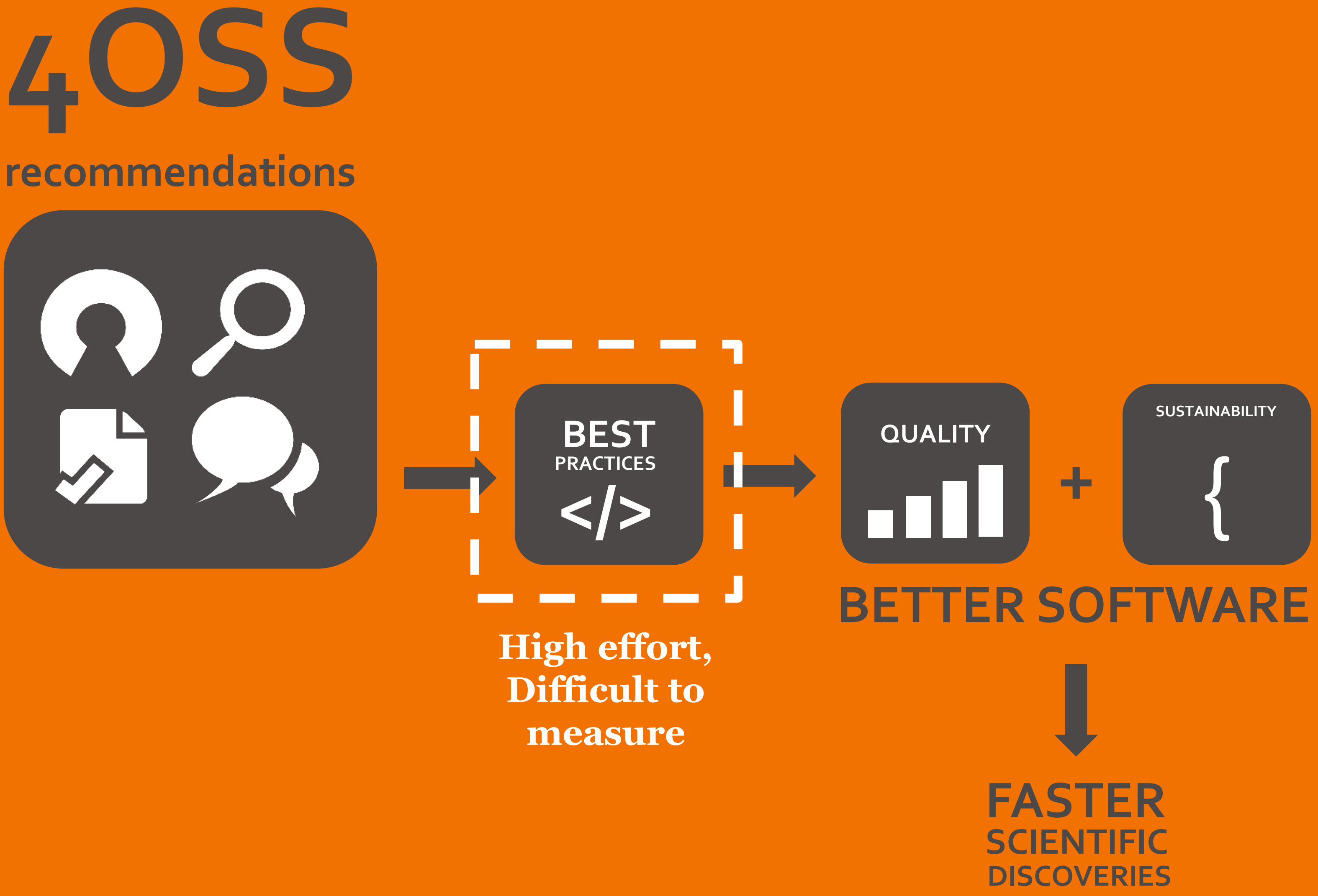
{

BETTER SOFTWARE

FASTER SCIENTIFIC DISCOVERIES

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

SUPPORT FASTER SCIENTIFIC DISCOVERIES



4OSS

recommendations

BEST

PRACTICES

</>

High effort, Difficult to measure

QUALITY

+

SUSTAINABILITY

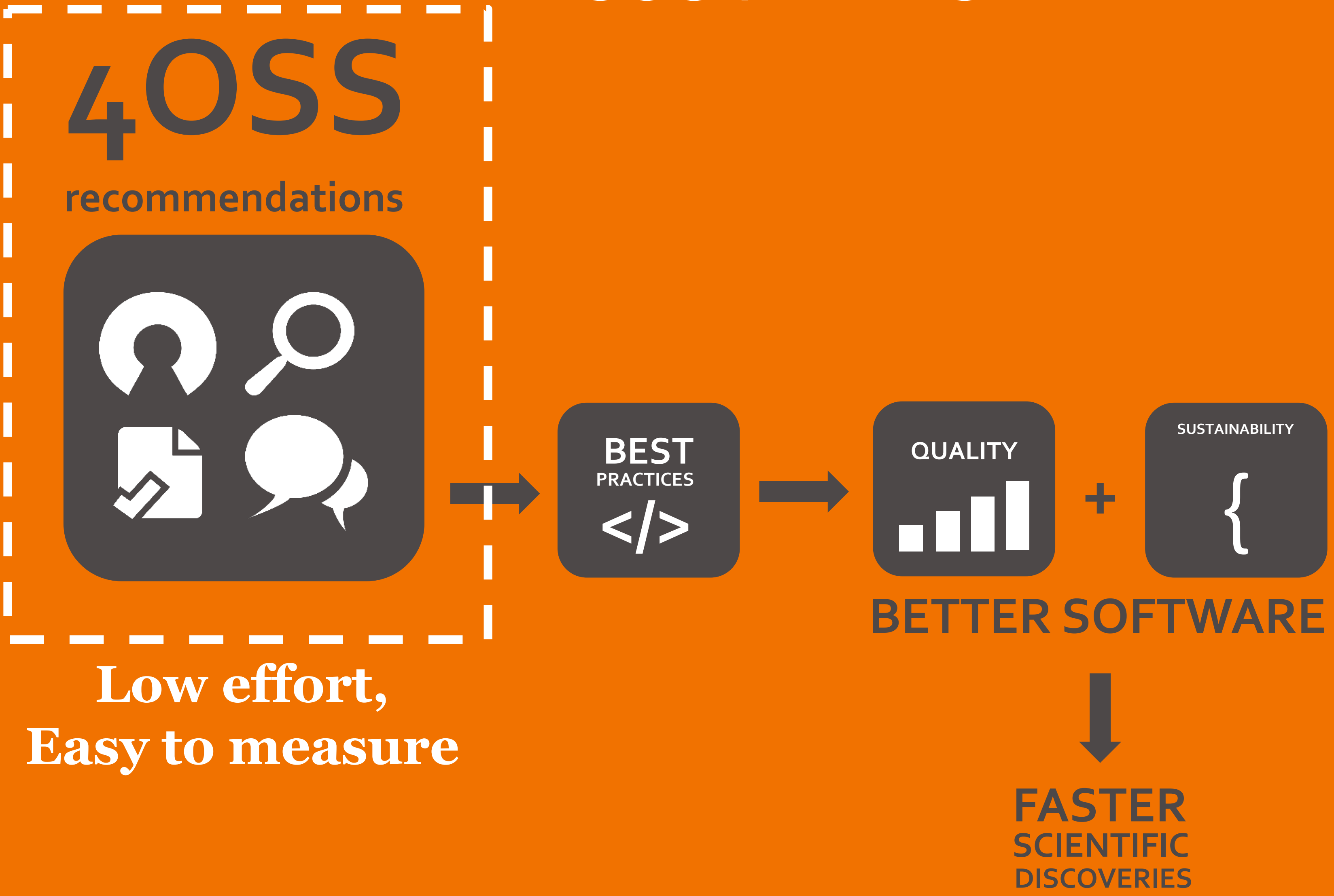
{

BETTER SOFTWARE

FASTER SCIENTIFIC DISCOVERIES

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

SUPPORT FASTER SCIENTIFIC DISCOVERIES



4OSS

recommendations

BEST PRACTICES

</>

QUALITY

+

SUSTAINABILITY

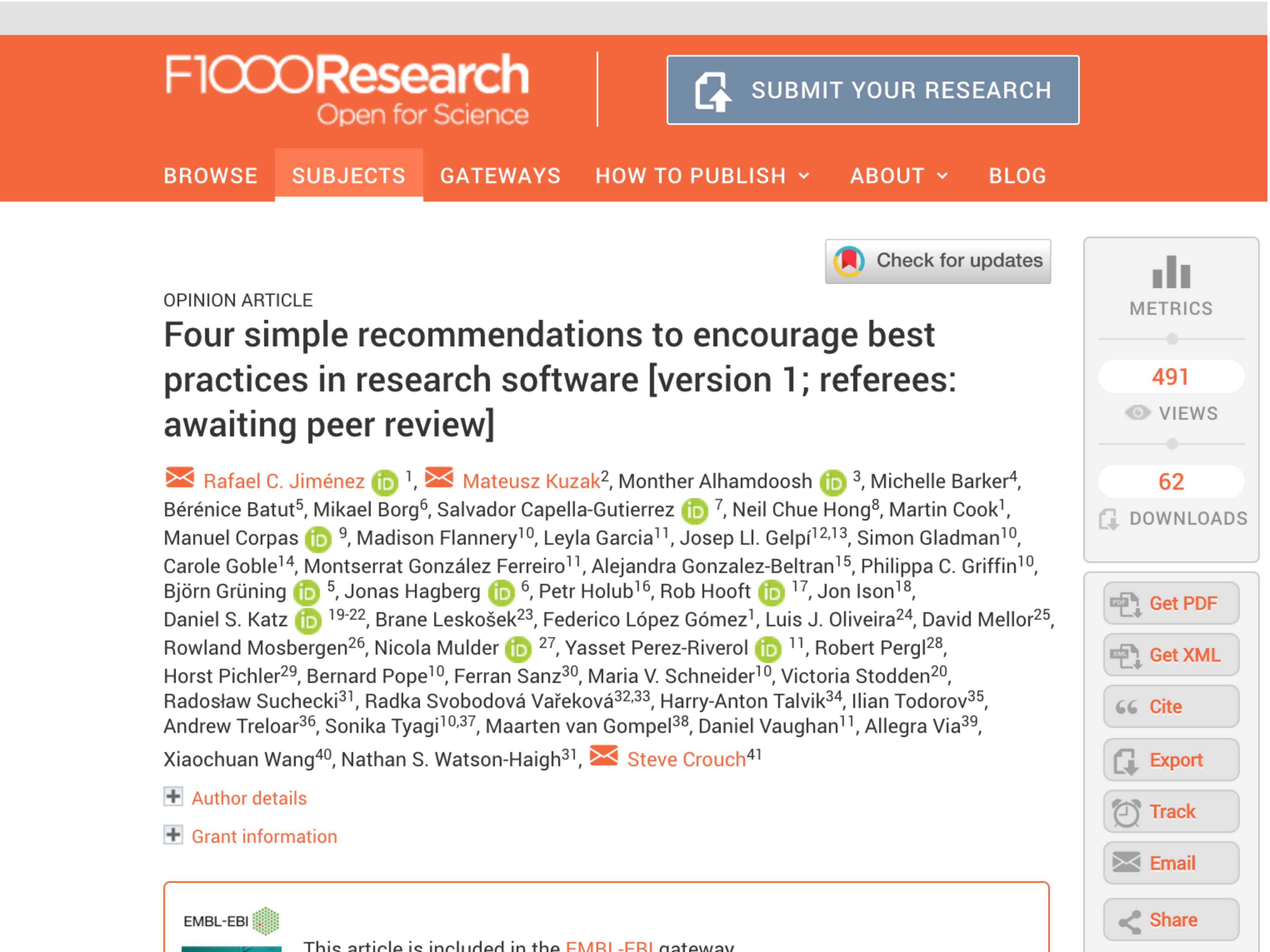
{

BETTER SOFTWARE

FASTER SCIENTIFIC DISCOVERIES

**Low effort,
Easy to measure**

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



OPINION ARTICLE

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

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]

⊞ Author details

⊞ Grant information

EMBL-EBI

This article is included in the EMBL-EBI gateway.

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

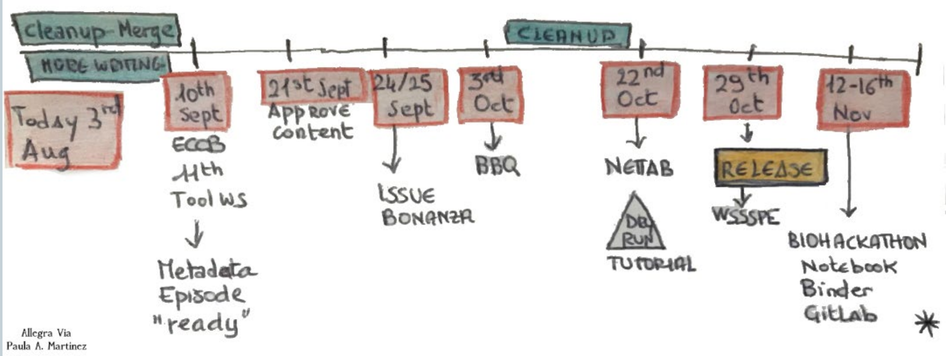
DEFINE RESPONSIBILITIES

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

Towards developing training material

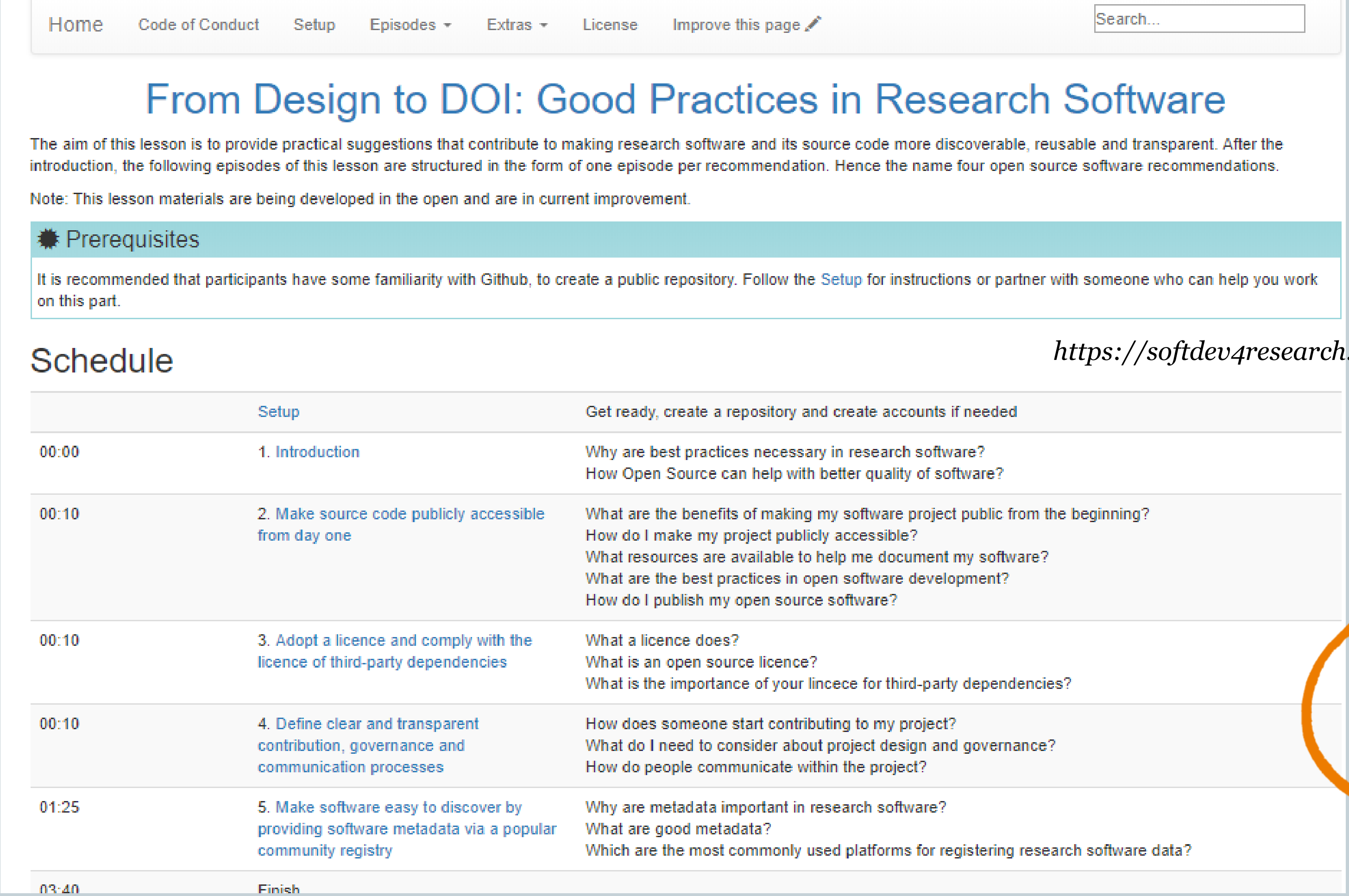
Workshop dedicated to outline the content for the hackathon to be further developed by the community

Hackathon at DTL / Utrecht, putting together the material - 2 days, 4 episodes, 25 participants from around the world



Final timeline

The Final lesson



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.

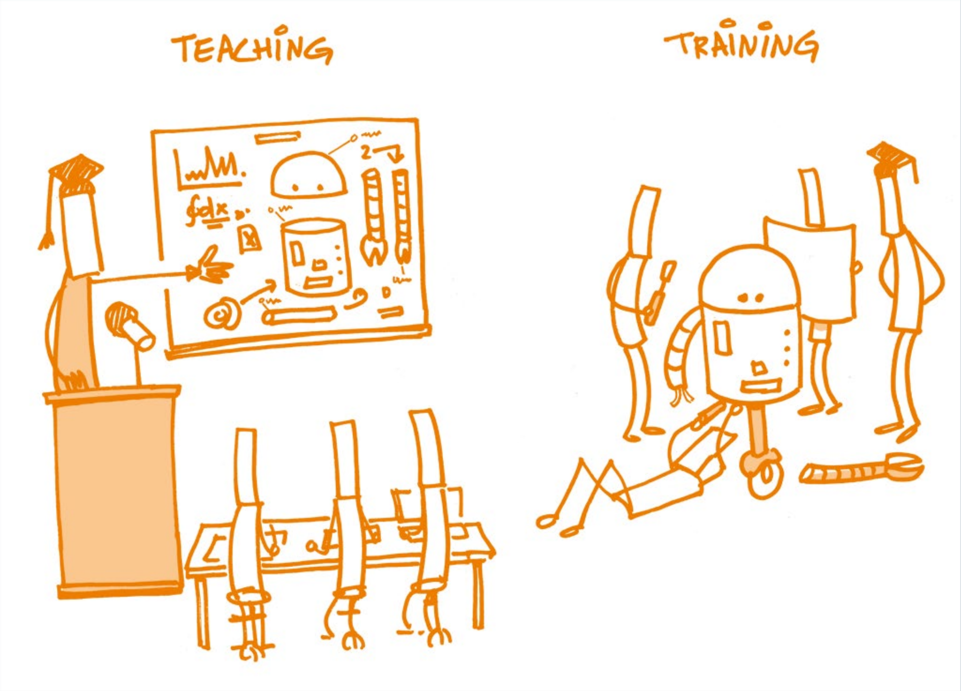
Schedule

	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 lincece 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/

The road ahead: Training

- High-quality trainings and training materials are **fundamental** when aiming at a cultural change towards the implementation of Open Science principles.



The road ahead: Guidelines

- Research grants:
  - usually request applicants to submit a data management plan
  - neither provide guidelines about how software should be developed or managed during the grant.
    - At most: "software should be freely available" or "tool will be open-source"
- We need:
  - concrete software guidelines that organisations and funders could embed as part of their policy for the development of software

FAIR Guidelines

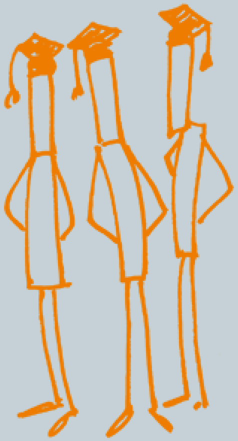
- Guidelines that:
  - are not hard to implement but still drive good practices in software development
  - are easy to assess to evaluate software compliance.

- What should be the content of these guidelines?
- How should software be assessed against the guidelines?
- How should the community support these guidelines?
- How should funders and organisations be engaged for the adoption of the guidelines?

- To re-iterate:

We need to have concrete software guidelines that organisations and funders could embed as part of their policy for the development of software.

The sooner we answer these questions and agree on a solution the easier will be to work towards improving the quality and sustainability of the research software to be developed in new research grants.



Thank you for your attention!