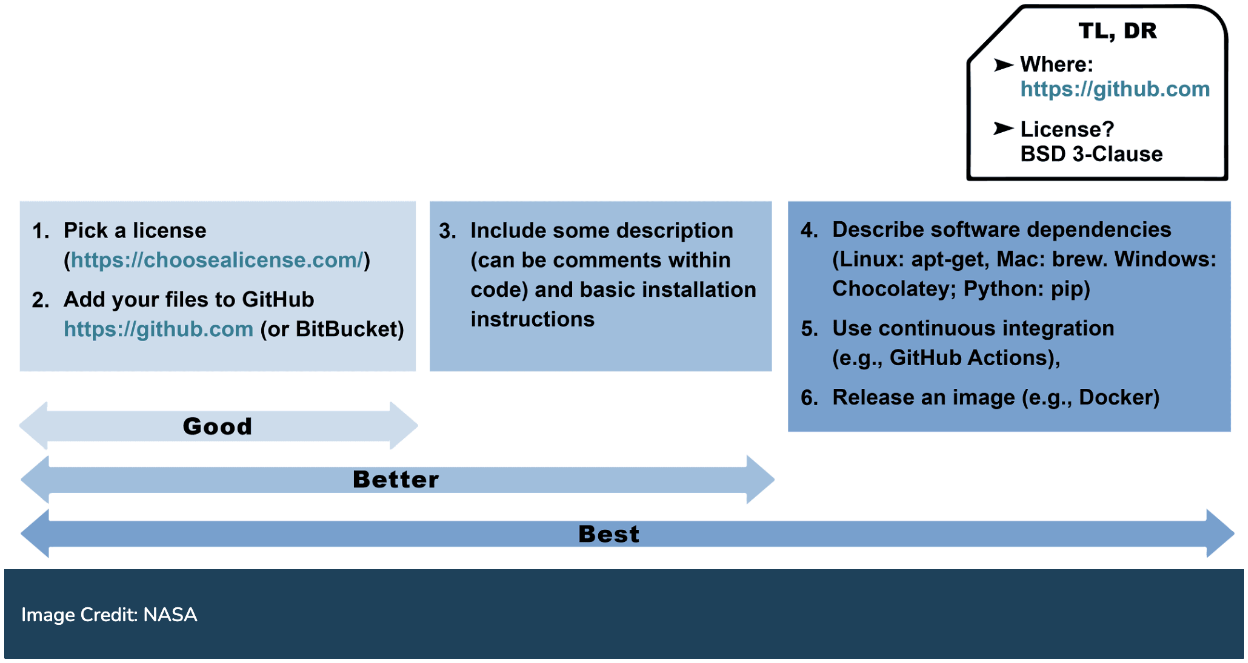
*The Process of Open Science*




How best to share code Via GitHub:

Look into Zenodo as a place to open share papers and results

General tools for open science:

- Digital persistent identifiers of PID's (Example would be an ORCID
    - "Open Researcher and Contributor IDentifier" links just to you
    -

More on *ORCID*: ORCID iDs are used to link researchers to their research and research-related outputs. It is a 16-digit number that uniquely identifies researchers and is integrated with certain organizations (like some publishers) that will add research products (such as a published paper) to an individual's ORCID profile. ORCID iDs are meant to last throughout one's career, and helps to avoid confusion when information about a researcher changes over time (e.g. career change or name change).

**Digital Object Identifiers (DOI)- Unlike dynamic transient URLs, DOIs are static pointers to documents on the internet.**

- **DOIs make citing research products easier and more useful.**
- **Resolve a DOI on a doc using https://www.doi.org/**

EX use case: "A workshop planning committee collaboratively authors a paper that summarizes the results of a workshop. They collect the ORCID iDs of everyone who participated in the workshop and include them in the paper. Finally, they publish in an academic journal that automatically assigns the paper a DOI."

"Metadata"

- Descriptive metadata can contain information about the context and content of your data, such as variable definition, data limitation, measurement/sampling description, abstract, title, and subject keywords.
- Structural metadata is used to describe the structure of the data (e.g. file format, the dataset hierarchy, and dimensions).
- Administrative metadata explains the information used to manage the data (e.g. when and how it was created, which software and the version of the software used in data creation).

Good documentation has Data, Code and Results

- "Another way to document and timestamp research plans and concepts is to share funded grant proposals publicly. This has the added benefit of making the funding process more transparent, and providing examples of successful grant proposals for other researchers, particularly those in their early career stage."

NASA ROSES - Funding for open projects

Funding organizations and agencies around the world are beginning to require open science plans!

**The Value of planning (Open science data management plan)**

The OSDMP describes how the scientific information that will be produced from scientific activities will be managed and made openly available. Specifically, a plan should include sections on data management, software management, and publication sharing. If your study has other types of outputs, such as physical samples, hardware, or anything else, you should include those in the plan. An OSDMP helps researchers think about the details of how they plan to share results.

A well-written OSDMP can help you win funding because it demonstrates your skills at doing open science!

Example sections to include in an OSDMP:

1. Data Management Plan (DMP)
2. Software Management Plan (SMP)
3. Publication sharing
4. Other open science activities
5. Roles and responsibilities

Successful DMPs typically include a clear terminology about FAIR and CARE Principles and how they will be applied.

- Descriptions of the data expected to be produced from the proposed activities, including types of data to be produced, the approximate amount of each data type expected, the machine-readable format of the data, data file format, and any applicable standards for the data or associated metadata.
- The repository (or repositories) that will be used to archive data and metadata arising from the activities and the schedule for making data publicly available.
- Description of data types that are subject to relevant laws, regulations, or policies that exclude them from data sharing requirements.
- Roles and responsibilities of project personnel who will ensure implementation of the data management plans

Scientific data is any type of information that is collected, observed, or created in the context of research. It can be:

- Primary - Raw from measurements or instruments.
- Secondary - Processed from secondary analysis and interpretations.
- Published - Final format available for use and reuse.
- Metadata - Data about your data.

FAIR - To be Findable Accessible Interoperable and Reusable https://fairsharing.org/

**Open code- an intro:**

Version control: (like using GitHub) has steps:

- You write code as you usually do in your code editor of choice (often I use VS Code). After you have written some code or made some updates to existing code, you then *commit* those changes to the version control system to create a sort of "checkpoint".
- Each commit requires you to add a short message which lets you briefly describe what changes were made. These messages serve as metadata that ensures collaborators, future users, and future you understand your development process at a point in time. **This is also useful for your own development, in finding bugs or critical points. Reverting to a checkpoint saves you as you increment progress and may make mistakes!**

Distributed version control systems (Git)

-Git is an open-source system that is commonly used in conjunction with web-based software hosting sites like GitHub (How use git link: https://swcarpentry.github.io/git-novice/ )

More eyes, more powerful edits: GitHub has a rich set of tools for reviewing and accepting (or denying) contributions from others or yourself, such as in-line comments and easily-viewable tracked changes to individual files.

**Main Computation,** A kernel is the core code that makes up the foundation of a computer's operating system. It manages the computer's system resources and interacts with the system's hardware. The kernel is a bridge between hardware and software.

**Computational Notebooks- A computational notebook refers to a virtual, interactive computing environment that combines code execution, documentation, and data visualization in a single interface.**

- The Laika project is developed in a Jupyter notebook, which allows for long explanations in markdown and notes. The notebook is also local web accessible

*Repository Vs Archive*

**Repos:** Open and active, use version control, hold the collaborative environment

**Archives:** Distribution of long term and stable software, tested, static environment

**Containers** are isolated environments that hold the application as well as anything needed to run the application, ensuring consistency and portability across different computing environments. Basically like a little self contained microcosm that allows everything to work. For Laika, I'm using a Docker container to house Linux ROS2 etc.

## *Tools for Open Publications*

- **Preprints** are early versions of research papers that are shared publicly before they are published in scientific journals.

    Why?:
    - Preprints enable researchers to swiftly share their findings
    - undergo a form of community review
    - Preprints may be cited when discussing ongoing research
    - Preprints are typically freely accessible to anyone with an internet connection.
    - NOT a replacement for peer reviewed papers, but very useful for feedback and open science practice!

LINK to find preprint servers: https://en.wikipedia.org/wiki/List_of_preprint_repositories
Submit or search for pre-prints!: https://osf.io/preprints#!
Directory of open access journals: https://doaj.org/

"Protocols.io (external link) is an online and secure platform for scientists affiliated with academia, industry, non-profit organizations, and agencies. It allows users to create, manage, exchange, improve, and share research methods and protocols across different disciplines. This resource can improve collaboration and record-keeping, leading to an increase in team productivity and facilitating teaching. In its free version, protocols.io supports publicly shared protocols, while paid plans enable private sharing, e.g. for industry."

**Keeping track of every paper you reference, every dataset you use, and every software library you build off of is critical. A single paper might cite dozens of references, and each new thing you produce only adds to that list.**