# Organizing Collaboration in Open Source Teams

Gregor von Laszewski, Fidel Leal, Shannon Kerr, Erin Seliger, Cooper Young, Agness Lungu

**Abstract**

To organize a research team, we need to establish simple communication pathways. This includes ways to conduct text chat video conferencing, code sharing, editing, and task management. In the following sections, we will list some useful tools that can be used by the research team while keeping the learning curve to a minimum. For more information and improvements, please contact: laszewski@gmail.com

## Contents

> **Note:** Do not edit this document, only edit the documents in the chapters directory

## 1 Organizing Collaborative Research Teams

To organize a research team, it is of utmost importance to establish simple collaboration pathways. This includes ways to conduct video conferencing, text chat code sharing, editing, and task management.

In the following sections, we will list some useful tools that can be used by the research team while keeping the learning curve to a minimum. In general, it is good to ask the participants if they already use particular tools in a category and if all in the team use them to adopt them. However, this may limit the general availability in case the team grows into the open-source community. Hence, it is important to consider licensing issues and if possible adopt free tools for the research team.

## 1.1   Communication

We start by identifying tools for voice and text communications.

### 1.1.1   Video Conferencing

Videoconferencing has undoubtfully become a major component of research teams. It allows face meetings without the need for traveling. Thus you can spend the time saved on travel. Also, it allows researchers to continue if unexpected events take place that does not allow in-person meetings such as the recent COVID epidemic.

There are many conferencing tools available that can be used. You may even use multiple dependent on the particular meeting or preferences by the subgroup. To keep things simple it is however recommended to just use one tool.

**1.1.1.1   Google Meet**   Google Meet is an online service that facilitates meetings as video and audio conference calls. It has evolved from google hangout.

**What are good features?**

Some good features of Google meet include the ability to use closed captions allowing the integration of participants having trouble hearing the speaker. It is compatible across devices and typically its sound quality is very good.

**What are not so good features?**

1. Goggle meet does not provide an easy way to have others take control of a remote desktop. However, it is possible to use Google Remote Desktop for it.

   - https://remotedesktop.google.com/?pli=1

2. Google meet cannot share each other's desktops at the same time. This feature was available in hangout but is no longer available as far as we can tell.

**Why you may consider choosing Google meet and not Zoom?**

Google offers many services that are useful for collaboration. This includes Google drive, docs, presentations, mail, calendar, and groups. features such as google drive. As they can be accessed through a single account, it is obvious that Google meet provides a valuable set of services to any research team.

**1.1.1.2   Zoom**   Zoom is a cloud-based communications platform that provides one-on-one, group meetings, and webinars.

**What are good features?** (Fidel, Gregor)

Some of Zoom's features include live chat, screen sharing, a whiteboard, and virtual reactions for meeting participants. Additionally, it can record meetings to the cloud or personal devices, create breakout rooms, and allows participants to seamlessly moving between them. A very important feature is that the meeting owner can remotely control another participant's screen. Zoom allows users to join a session through an established meeting URL. Participants do not need to be signed in or even have a Zoom account. Additionally, people joining from places with limited Internet access can call into the meeting's audio channel using dedicated telephone numbers.

**What are not so good features?**

Free account holders can host unlimited one-on-one meetings (meeting durations up to 24 hrs). In contrast to Google meet, there is a duration limit of 40 mins for meetings with three or more participants. We observed that video quality can be unstable, and the overall platform performance can quickly deteriorate over limited bandwidth connections. In such cases, we recommend switching the camera from the participant that has issues. Furthermore, if you do at the same time a lot of calculations on your machine it may affect the quality of the call. This applies to older machines and should allow you to give a beautiful argument to get a new

computer. In some cases, you may have a second computer and can use one for sharing your session, while the other one is used for sound, or you use a cell phone for the later

**How can someone take control of a remote desktop?** (Fidel)

To take control of a remote desktop, the remote user must activate screen sharing. Once the screen sharing is activated, we need to click the `View Options` dropdown menu (usually at the top of the screen) and click on `Request remote control`. The remote user will then get a prompt to approve the remote control request.

Institutional accounts may have the remote control functionality disabled by their account administrator. For further details, refer to the Zoom support pages.

### 1.1.2  Realtime nad Offline Text Messaging

In many research projects, participants may be in different timezones or have schedules that do not provide overlapping times for video-conferencing. For this reason, it is important to support a chat-like feature, that allows the researchers to catch up with activities that took place they were not available for. Tools such as e-mail have filled this demand for quite some time. Recently additional tools such as Slack have appeared that enhance the e-mail activity while also allowing real-time text messaging,

**1.1.2.1  Slack (Gregor, Cooper)**  Slack is a communication software that is used for groups to send and receive text messages. Additional 3rd party services that can be added to slack that can send messages to it even automatically.

**What are good features?**

Slack offers a GUI that focuses on a real-time message stream. It can be used on computers, phones, and tablets. It is easy to send photos, which may be useful in case the device you need to discuss is not on the internet but you need to share the content for example of its screen.

**What are not so good features?**

Slack is stream-based and does not provide a good mechanism for organizing messages once they have been sent. The thread feature is far inferior to that of even a simple e-mail client. If one is involved in many slack workspaces, it becomes difficult to manage them. Most importantly Slack comes in its free version with only a limited number of free messages. This means you will have to pay once you exceed the limit. Thus even the integration of useful services such as GitHub notifications is not recommended as you will too quickly exceed the limits. A posting policy needs to be established. Those that are not using slack frequently may be out of touch quickly. Although there is an unread threads feature, it may be filled with messages if you do not use slack daily just to keep up.

To support the separation of topics it is advisable to create several channels such as "general" or a channel for a particular topic. However, it is also important to limit the number of channels so it does not become too confusing

## 1.2  Creating Text

As part of your project, you will need to develop documents such as manuals or reports hence you must have the means to easily add text to your project. This includes the creation of text via editors but also through dictation in case this is useful and works for you.

### 1.2.1  Dictation

Sometimes it is convenient to directly dictate the text for a manual or tutorial into an editor. On macOS and Windows, you will find useful tools for this. A voice-to-text recorder may also help you in case you have a recoded video of yourself to generate transcripts. A disadvantage for a lot of non-native English speakers is that the accuracy may be limited and that not using them leads to unacceptable results. Some of them can be trained.

Hence before you integrate it into your toolset, we recommend trying it out. Different participants may be more successful than others. However, errors will happen and you will have to clean up the document after dictation.

Examples include:

- Apple Dictation (free for Apple devices): You can directly dictate into various applications that help you improve your text such as Grammarly and MS Word if you have installed them.

- Windows 10 Voice Recognition (free for Windows users). You can directly record into MS Word so you get a free grammar checker. **Note: We received reports from some of our participants that they could not get this to work. More investigation is needed.**

- Google Docs offers build-in voice typing. Its recognition quality is very good as well as fast. It does have difficulties recognizing some names and acronyms. **Note: that we were unable to get it to work in Chromium on Ubuntu 21.04. We have not tried other versions.**

### 1.2.2 Grammar Checkers

When developing content for tutorials and documentation it is important to check their correctness with a grammar checker. We have made the best experience with Grammarly followed by MS word. The best way to use them is to copy and paste small sections into them from your document and then check them. After you are satisfied copy the contents back to the original one while overwriting the old text.

#### 1.2.2.1 Grammarly    What are good features?

Grammarly works well, is available for free, and the free version is good enough for most.

**What are not so good features?**

As with any grammar checker, not everything is corrected properly. In some text, it shows false errors, but it's still very good.

#### 1.2.2.2 Word    What are good features?

The word grammar checker is built into word and has a high accuracy.

**What are not so good features?**

In practice, we observed that Grammarly performs better. Copying text back and forth can introduce errors when it comes to using quotes and other symbols. Thus you must check all symbols after you copy them into a markdown document. IN case you have the choice we currently recommend you use Grammarly.

### 1.2.3 Editors

You will need likely multiple two editors as part of your research activities. This is motivated by the fact that we do lots of development on your local machine, but also do remote development via terminal access to a remote computer that does not have a GUI. In case you only want to learn one editor to do all of this, just use emacs. We have listed below some editors and you may want to choose

- `emacs` vs vi/vim for terminal editing
- `pycharm` vs MS code for fancyful python code development

There is a third option that we will use and that is Jupiter which allows us to interactively develop python code. Jupyter is important as it is used by many data scientists due to its ad hoc interactive mode while programming. However, also pyCharm and VScode provide options to view and run Jupyter notebooks. However, not everything that works in Jupyter is working on these platforms.

Here is a list of popular editors for python:

- emacs

- Pro: terminal, established, very good markdown support, block format with ESC-q, keyboard shortcuts also used in bash, and their shells have a python and markdown mode

- Cons: some users have a hard time remembering the keyboard shortcuts, the editor may get stuck in some unknown mode that you activated by accident. ALL this is easy to fix by remembering CRTL-X-s (save) and CTRL-G (get out of a strange mode)

- vi

  - use vim instead, vi is available on all Linux machines, but has rather archaic editing controls.

- vim

  - Pros: like vi, but with cursor control
  - Cons: often awkward to use

- Pycharm

  - Pro: best Python editor
  - Cons: needs lots of resources, the steep learning curve

- VS Code from Microsoft

  - Pro: often used on Raspberry
  - Cons: Pycharm seems to have more features from the start

## 1.3 Documentation with Markdown

We have developed a special booklet that helps you to feel the many features of markdown that can be used to create scientific documentation please follow the link to this document to find out more. For open-source projects, you often need to develop documents that are portable across platforms and do not depend on editors provided by a specific vendor. They exist a great number of formats that could be used but most recently markdown has become one of the easier-to-use document formats For developing portable documentation.

- Scientific Writing with Markdown, Gregor von Laszewski

There are many editors for editing markdown locally on your computer or and services that are offered remotely. For groups one of the services that we have successfully used is Hackermd.io. On macOS one of the very useful editors at macdown. However, emacs already provides a built-in markdown editor for you if. If you are using PyCharm it also comes with a MacDown editor but you cannot enable it if it does not come with your version.

For complex documents, the management of the document with Github is superior to those of online editors. This is especially the case if the documents have to be split up into multiple documents due to their size.

## 1.4 ToDo Lists

It is important to communicate quickly some tasks in documents that we write as a team. To do this we use the keyword TODO, followed usually by an explanation what needs to be done. As a TODO can be hopefully resolved quickly it should be able to complete them in 1-2 hours. Any TODO that may take longer we also add to our GitHub project for it to be recorded and if we identify delays in its execution we can assign additional team members to help on these larger tasks.

Once a team member has identified a TODO item, the team member can simply put his name behind it, as well as the date and time of teh anticipated completion so others know you work on it. Also, it is recmmended to communicate on slack about the task you do if you run into issues or have questions.

In case a TODO has not yet been assign, a team member can simply tak it and complete it. In practivce we have seen in some groups that the project lead needs to assign tasks to speed up the development and avid length times of inactivity by team members.

## 1.5   Git and GitHub

Git is a distributed version control system to support working on project in teams while allowing different team members to contribute and to curate the contribution through reviews.

GitHub is a service offered for free with the limitation that the repositories should not be larger than 1GB and the individual files must be smaller than 200MB. Github is very popular for OpenSource projects and through its free offering allows community building around OpenSource Projects.

### 1.5.1   Git from the command line

Git can easily be installed on all platforms including

- macOS: You will need to install Xcode which includes not only git but user Linux programs such as Makefile
- Ubuntu: You can install it via `apt install git`
- Windows: You can install it via *Git Bash* which is distributed from https://git-scm.com/downloads. When installing read carefully the available options. We recommend you install a desktop shortcut.

## 1.6   Git form IDEs

Pycharm is one of the best editors for Python. It does provide build-in support to interact with GitHub. For beginners, we do recommend to get started with PyCharms github features.

## 1.7   GitHub from a GUI

Some may fancy using a Graphical user interface to interact with GitHub. However, in many cases, the terminal access is simpler. However, if you like to browse the repositories and see the commit tree, these GUI interfaces are useful. Several such interfaces are available at:

- https://git-scm.com/downloads/guis

### 1.7.1   GitHub Commands

Many tutorials introduced get up in its full details however many of the contributing participants in open source project may not need them. That is it important to identify which commands are the most useful ones to get team members started

What are the most important commands?

`git pull` retrives the latest content from the shared remote repository

`git add FILENAME` adds a filename to the local repository. Also, do never use the git command `git add .` as that adds all files and you could have files that you do not want to commit. instead **always** use `git add FILENAME`, where FILENAME is the file you like to add.

`git commit -m "commit comment" FILENAME` commits the current content of the FILE to the local repository

`git push` pushes the content from the local repostory to the shared remote repository

More advanced feature off get is the use of branches. With branches we can coordinate individual contributions that I merged into the main remote repository. This is done With a branch name as well as a git pull request.

`git checkout BRANCHANME` creates a new branch in your local repository

`git push -u origin BRANCHNAME` pushes the branch with the name BRANCHNAME to the remote repository for all to see

After your branch may be reviewed by the team it may be decided to merge it into the main branch. This has to be done carefully and multiple people should review such a merge before It is executed and pushed. You can try out a branch by checking it out with the command

`git checkout BBRANCHNAME`

After the check out you have the contents of the branch in your local repository and you can verify if it works. Usually one person or team members are dedicated to merging branches. A branch can be merged with the commands

```
git checkout main
git merge main BRANCHNAME
```

To push the branch into the remote repository simply used to the push command.

```
git push
```

# 2 Appendix

## 2.1 Hardware of current students

- Fidel Leal,
  - Equipment
    * MacBook Pro 2015, 16GB RAM i7, SSD 512GB
    * PC, 64-bit, 8GB RAM, i5, SSD <240GB, speed>
      · Windows 10 Education
    * Editor: Pycharm, vim
- Shannon Kerr,
  - Equipment
    * Dell Inspiron, i5, 8GB, 1.6GHz 5482,
    * HDD 1TB 5.4K 6GB/s
    * Windows 64bit Home
    * Editor: Vim
- Erin Seliger
  - Equipment
    * Windows hp 2020, 16GB RAM, i7, 64-bit operating system
    * Windows Pro 64bit
    * Editor: Vim
- Cooper Young
  - Equipment
    * Dell Inspiron 7000, i7 2 Ghz, 16GB RAM, Intel Optane 512GB SSD
    * Windows 10 Education 64bit
    * Vim, Pycharm, Pico
- Agness Lungu
  - Equipment
    * Lenovo V570, 16GB RAM, intel(R) core(TM) i5-2430M, 64-bit operating system,
    * Windows 10 Education
    * editor: Vim, Pycharm

# 3 Make on Windows

Make fires provide a good feature to organize workflow while assembling programs or documents to create an integrated document. Within makefiles defined targets that you can call and are executed. Through its mechanism, it can easily deal with complex workflows that require a multitude of actions to be executed. Makefiles are executed by the program make that is available on all platforms.

On machines such as on machines that run Linux, it is likely to be pre-installed while on MacOS you can install it with Xcode. On Windows, you have to explicitly install it and we recommend that you use get bashed the window so that you can call make from us and get bash. The following instructions will provide you with a guide to install make under windows.

## 3.1   Installation

Please visit

- https://sourceforge.net/projects/ezwinports/files/

and download the file

- 'make-4.3-without-guile-w32-bin.zip'

After the download you have to extract and unzip the file as follows:

```
$ cp make-4.3-without-guile-w32-bin.zip /usr
$ cd /usr
$ unzip make-4.3-without-guile-w32-bin.zip
```

Now start a new terminal and type the command

```
which make
```

It will provide you the location if the installation was successful

"'bash /usr/bin/make

to make sure it is properly installed and in the correct directory. >

## 3.2   Installing WSL on Windows 10

WSL is a layer that allows the running of Linux executables on a Windows machine. This broadens the number of commands able to be run and creates more flexibility.

To install WSL2 your computer must have Hyper-V support enabled. THis doe snot work on Windows HOmew and you need to upgrade to Windows Pro, Edu or some other Windows 10 version that supports it. Windows Edu is typically free for educational institutions. The HYper-V must be enabled from your Bios and you need to change your settings if it is not enabled.

More information about WSL is provided at

- https://docs.microsoft.com/en-us/windows/wsl/install-win10 for further detail

To install WSL2 you can follow these directions while using Powershell as an administrative user and run

```
ps$ dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
ps$ dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
ps$ wsl --set-default-version 2
```

Next, Download Ubuntu 20.04 LTS from the Microsoft store

- https://www.microsoft.com/en-us/p/ubuntu/9nblggh4msv6?activetab=pivot:overviewtab

Run Ubuntu and create a username and passphrase.

Make sur enot to just give an empty passphrase but chose a secure one.

☐ TODO: Cooper, how to start it the next time