

Background

Not all projects are equally good for someone wanting to become a contributor. Some projects just don't welcome new contributors, or are not well organized to support getting new people up to speed. Other projects are welcoming to new contributors and provide clear pathways to join the community. Anyone considering becoming a contributor to a project should have some idea what to look for in evaluating whether a project is a good choice for becoming a contributor. While these evaluation criteria are not foolproof, they at least provide a starting point and framework of things to consider.

Directions

Walk through of an evaluation of the OpenMRS project

There are many criteria that should be looked at when determining if a project is appropriate to use in your class. These criteria are prioritized and explored below. The [Project Evaluation Rubric](#) contains instructions for each criterion and directions on scoring criteria. Place a copy of the Project Evaluation Rubric in your documentation. Include your findings (notes and the answers to the below questions) in your copy of the rubric, along with your scores for each.

Licensing - An important first step is to identify the license used by the project. An open source project must specify that others are free to use it, redistribute it, change it, and redistribute modified versions too. An extensive list of open source licenses can be found at <https://opensource.org/licenses/alphabeticall>. A list of free software licenses can be found at <https://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses>

- Go to the OpenMRS core repository (<https://github.com/openmrs/openmrs-core>). On the repository page, click on the "< > Code" tab below the repository name. Look at the information line below the tabs for a license name (see below image). Note: if the license does not appear here, or if a project is not on GitHub, look at the top level files in the repository for a license file.
- Does OpenMRS use an OSI approved open source license? Enter your findings in the rubric.

The screenshot shows the GitHub repository page for 'allready' by tonydurma. The repository description states: 'This repo contains the code for allReady, an open-source solution focused on increasing awareness, efficiency and impact of preparedness campaigns as they are delivered by humanitarian and disaster response organizations in local communities. <http://www.htbox.org/projects/allready>'. The repository statistics bar shows: 2,627 commits, 2 branches, 104 contributors, and MIT license. Below this is a list of files and their commit history:

File	Commit Message	Time Ago
AllReadyApp	Merge pull request #1662 from mgmccarthy/AbstractHttpClientForGoogleO...	an hour ago
SampleData	adding sanitized sample data	3 months ago
docs	added steps for adding Android SDK dirs to path	20 days ago
.deployment	move deployment script to the root of the repo	a year ago
.gitignore	Ignore lower_components, and give a test a little bit more room for m...	4 months ago
CONTRIBUTING.md	Repaired a link	3 months ago
LICENSE	Initial commit	2 years ago
README.md	update list of upcoming events	15 days ago



Language - The language(s) used in the project is an essential consideration for your students. If the project is written in a language(s) that your students are already familiar with, or better yet, well versed in, this is one less hurdle to overcome.

- Click on the language details bar (see above image). Record the top three languages used and the percentage for each.

Activity - To support student participation, a project should be reasonably active. Number of commits can be used as an indicator of activity. Little to no activity over a year, for example, may indicate that the project is dead, or mature and not being actively developed.

- Click the "Graphs" tab then select "Commits"
- If we define "Active" as meaning that a majority of the weeks in a given quarter have commits, determine whether each quarter was active over the last year and place your findings in the rubric. Note: since the definition of "active" is approximate, assess each quarter at a glance rather than by actual count of commits.

Number of contributors - A common fossism states that "It's all about community," so a suitable project should have an active user community. The community members are great resources for both faculty and students as they begin to learn about a new project, its culture, and norms.

- Click the "< > Code" tab. The number of contributors is listed above the language details bar. Determine how many contributors there are to the OpenMRS core project and enter your findings in the rubric.

Size - The size of the project is likely to be a factor depending on the level of your students. A large project that is built using many various technologies is likely to seem overwhelming to a CS2 student, for example, but may be a perfect fit for a senior capstone course. A simple first step is to determine how large the project is, additional research could be done to ascertain complexity. By default, Github does not provide information about how many lines of code there are in a repository or its size. You can however install an extension for Google Chrome that will display the size. Follow the instructions below to install the extension.

1 Click the "Customize and control Google Chrome" button (see below image)

2 Click "More tools"

3 Click "Extensions"

4 Scroll down and click "Get more extensions"

5 Search for "GitHub Repository Size"

- You should now see the repository size next to license type. Record the size in the rubric.



Issue tracker - The issue tracker can provide insight into the health of a project. An active issue tracker should highlight issues that clients/developers have logged as

well as an indication that these issues are being addressed.

- Click "Issues" (note: this should appear next to "< > Code"; if you do not see this tab, then there are no issues logged in Github). OpenMRS uses a third-party issue tracker - click the link to openmrs.org located near the top of the repository page, scroll to the bottom and click the "OpenMRS Issue Tracking" link. Scroll to the table labeled "Two Dimensional Filter Statistics: All JIRA Tickets" located near the bottom of the page. Provide answers to the following in the rubric.

1 How many open (for OpenMRS look at "ready for work") issues are there?

2 How many closed issues are there?

3 When was the fifth issue opened (for OpenMRS look at the "ready for work" issues)?

4 Based on browsing the table, and clicking on some of the table cells, assess whether issues are actively being added and resolved

New contributor - The project should appear welcoming to new contributors. Some clear examples of this would be links to getting started pages or information on ways to become involved. These pages, in turn, should include additional detail about **how** to become involved, as well as information about **how** to connect with the community.

- Browse the repository and associated links, is there any indication that the project welcomes new contributors? Indicate which of the following are present and provide links in the rubric. Note: for OpenMRS you will find that the link at the top to openmrs.org and the link toward the bottom of the repository page to the OpenMRS wiki quite useful!

1 Are there instructions for downloading and installing the development environment?

2 Are communication mechanisms, such as IRC, list serves, you can join, meeting notices, etc. apparent?

3 Is there a discussion platform? If so, how recent are the responses?

4 Is there Web presence? This might include information about the project, how to get started as a developer, links to blogs, links to IRC logs, links to pages that contain information about coding standards and the code submission process.

Community norms - The way in which community members interact with one another is equally important, especially for student involvement. You do not want to point students to a project that advocates or permits lewd and unprofessional behavior.

- Some projects provide a "Code of Conduct", yet others do not. It is quite possible that you will find the code of conduct more quickly by doing a Google search. For OpenMRS you should look in the "Developer Guide" (link along with left side in the OpenMRS wiki) and then choose "Conventions"
- You should also review some actual communications to determine if there are any indications of rude or inappropriate behavior. This could be quite time consuming since you would first have to determine the type of communication typically used by community members and then locate and review the appropriate artifacts. For OpenMRS, click the "TALK" link at the top of the OpenMRS wiki page and review the communications that occurred for two of

the topics. Choose two that have at least 5 members and 15 or more replies.

- Record the following in the rubric.

- 1 Provide three observations about the OpenMRS Code of Conduct.
- 2 Provide three observations about the type of communication that occurred between community members on TALK. Is there any indication of rude or inappropriate behavior?

User base - A project will not thrive without a core user-base. The user-base consists of clients, people who use the product on a day-to-day basis. They provide the development team with necessary feedback about the project, what works, what doesn't and what new features they might like to see. If no one is using the product then developers are likely to abandon it. Browse the repository and related links.

- In the rubric record your answers to the following.

- 1 Does there appear to be a user base?
- 2 Are there instructions for downloading and setting up the software for use by clients?
- 3 Are there instructions for how to use the software?