Assignment 10: Project Evaluation

1 Summary

Given that one of your major objectives is to make a meaningful contribution to a free and open source software (*FOSS*) project, it is crucial that you find a project for which this is possible in the time frame that you have allocated for this task. Your objective might even be more restrictive because you want to make a contribution to a "humanitarian" FOSS project, called an *HFOSS* project, of which there are much fewer from which to choose. The number of FOSS projects is so large and varied in terms of their approachability, size, programming language base, level of activity, complexity of code, community friendliness, domain knowledge prerequisites, and complexity, that you will need some means other than an exhaustive search for choosing projects that are the optimal choices for you. This suggests, or more strongly implies, that you need a set of criteria by which you can evaluate a project for its "fit" for your purpose, and that there should be an effective means of deciding for any project how well it satisfies these criteria. Therefore, I have tried to design this assignment to give you some experience in evaluating projects in this way, using a combination of qualitative and quantitative measures.

2 Background

Though open source software existed before the *World-Wide-Web* was created, the Web and Internet connectivity have been essential for the rapid expansion of FOSS in recent years. FOSS projects need to be available on the Web in order for anybody in the world to be able to contribute to them. There are several websites that provide a home and visibility for FOSS projects (although many of the biggest projects are hosted on their own servers and websites). Some of these websites are ***source forges***, such as *SourceForge*, and some are ***full-featured development environments***, such as *GitHub* and *GitLab*. In addition, there are online directories of open source projects, such as *OpenHub*, which provide statistics about projects, such as the number of contributors, software metrics, and commit activity.

In short, different projects are hosted on different platforms. As of October 2018, *GitHub* hosted about 96 million repositories, with more than 28 million users and contributors[1] and *SourceForge* hosted about 430,000 projects with 3.7 million registered users[2]. *GitLab* is much smaller, with about 1,900 contributors and 550,000 projects[3]. *Wikipedia* has a comparison of source code hosting sites at

https://en.wikipedia.org/wiki/Comparison_of_source_code_hosting_facilities.

The point is that there are many different places to look for projects. In this assignment, you will evaluate a few projects that are hosted on different sites.

3 High Level Project Evaluation Criteria

Let us use the term "***appropriateness***" to mean how well a project fits your goal of being able to contribute to it meaningfully in the time frame of a couple of months. The appropriateness of a project is determined, for the most part, by the answers to a set of important questions. Some of these questions are more important than others in terms of whether the project is a good choice or not. Some researchers in this field have even suggested that the questions that should be asked fall into three categories related to the project: *viability*, *approachability*, and *suitability* [1], and that some should be tagged as *critical* whereas others are not as critical and are tagged as *secondary*. I will not categorize them as such. I view this as an optimization problem, in that different projects will satisfy some criteria more than others, so that there is no single metric that can be used to assess projects. It will reduce to a question of which criteria become more important in the end.

The general questions that you need to answer about a project under consideration are

[1] Source: https://github.com/about
[2] Source: https://sourceforge.net/
[3] Source:https://en.wikipedia.org/wiki/Comparison_of_source_code_hosting_facilities

- Is it open source? This is obviously the first question to answer. A negative answer for this one stops further evaluation.
- What is its license? Does the license allow for forks and modifications?
- How active is this project? Are people actively submitting issues and are people closing issues in a timely manner?
- How welcoming does the community seem? Are people friendly in the issue discussions, the discussion forum, and chat?
- How easy is it to find information about contributing to the project? Are there clear guides and documentation that can help someone who wants to contribute? Are there written guides about rules of conduct, for example? In short, is it an inviting project?
- What programming languages are used in the project?
- What is the development environment, and how hard is it to download and install it?
- How hard is it to understand the project code? Is it a large code base?
- Does the project depend on external additional software modules such as database or graphics libraries?
- How much does one need to know about the domain of the application in order to understand the code? For example, if it is a health-related application, how much medical or biological background would a contributor need?
- How complicated is the code and how large is it?
- How mature is the project? Is it very new, or has it been around for a long time? Does it have a stable release yet?
- Is the project interesting and/or exciting to you?

It is unlikely that you will find a project for which the answers to all questions are the "good" ones. You will probably have to make some hard decisions. But

- if the community is not friendly,
- if finding the answers to questions is hard,
- if the project is not very active,
- if the code is hard to understand, or
- if it is very difficult to download and install the development environment,

then it is not a good first project.

So how do you answer these questions? How can you evaluate a project based on these criteria? What follows are specific questions for which you should find the answers for any project. Some of this material comes from the *GitHub Open Source Guide*[4],

`https://opensource.guide/how-to-contribute/#finding-a-project-to-contribute-to`

and some of it is based on materials developed elsewhere.

4 Specific Questions

These are the specific questions that you should answer for each project listed in section 5 below.

[4] Content based on github.com/github/opensource.guide used under the CC-BY-4.0 license.

4.1 Finding the Project License

- What is the project's license? There might be more than one. If so, what are some of the licenses that you found? On most source forges there will be a file named `LICENSE` or something similar in the root level of the repository.

4.2 Assessing Activity

- When was the last commit?
- How many contributors does the project have?
- How often do people commit?

On *GitHub* you can find this information in the repository's root directory. For example, you can find commit activity by clicking `"Commits"` in the top bar. For projects that have their own sites, you have to find the repository and dig around. You can also look on *OpenHub* for this information.

- How many open issues are there?
- Do maintainers respond quickly to issues when they are opened?
- Is there active discussion on the issues?
- Are the issues recent?
- Are issues getting closed?

On *GitHu*b, a project might have an "`Issues`" tab and you can view them. You can click the `"closed"` tab on the Issues page to see closed issues, for example. Some projects maintain their own issue trackers and you will have to find them by "following your nose" on their websites (e.g., *Mozilla*). Projects listed on *SourceForge* usually have repositories hosted on some other site, such as *GitHub*, where they will have their issue tracker.

- How many open pull requests are there?
- Do maintainers respond quickly to pull requests when they are opened?
- Is there active discussion on the pull requests?
- Are the pull requests recent?
- How recently were any pull requests merged?

On *GitHub*, you can click on the "closed" tab on the `Pull Requests` page to see closed pull requests.

4.3 Assessing the Welcomeness

- Do the maintainers respond helpfully to questions in issues? Are responses generally constructive?
- Are people friendly in the issues, discussion forum, and chat?
- Do pull requests get reviewed?
- Do maintainers thank people for their contributions?

5 Projects To Evaluate

There are five projects listed below. Evaluate *Quantum GIS* and two others of your choice following the instructions below. See Section 6 for what to submit and due dates.

5.1 Quantum GIS

Quantum GIS, also known as *QGIS*, is a free and open source geographic information system mostly written in C++. It might be a potential project to which you could contribute. Its code is hosted on *GitHub*, but much of the development resources are on its own website, QGIS homepage. Start on the home page and navigate elsewhere as needed. You will see that the issue tracker is hosted by them but the code is not.

5.2 Epiphany

Epiphany, also known as *Gnome Web* is the web browser for the *GNOME* desktop. Start on *OpenHub* to explore this project. The page for *Epiphany* is https://www.openhub.net/p/epiphany. Many of the questions listed in Section 4 can be answered by browsing this page. In particular, answer these questions first:

- What is the main programming language used in *Epiphany*?
- How many lines of code does *Epiphany* have?
- What percentage of the code is comments?
- How many commits were made in the last 30 days, based on *OpenHub* statistics?
- Click on "`User & Contributor Locations`" (lower right side of screen). List some of the locations of the developers.

The code is hosted on *GitLab*. In order to access the repository, you will need a *GitLab* account and will need to authorize Gnome to access it. You can refuse to do this, but if you choose to evaluate this project, you will need to do this.

5.3 LibreOffice

LibreOffice is an integrated office suite based on copyleft licenses and compatible with most document formats and standards. It is mostly written in C++. Start on *OpenHub* to explore this project. The page for *LibreOffice* is https://www.openhub.net/p/libreoffice. Many of the questions listed in Section 4 can be answered by browsing this page. In particular, answer these questions first:

- How many lines of code does *LibreOffice* have?
- What percentage of the code is comments?
- How many commits were made in the last 30 days, based on *OpenHub* statistics?
- Click on "`User & Contributor Locations`" (lower right side of screen). List some of the locations of the developers.

The code is hosted on *GitHub*. Evaluate the `core` part of the project. You will see that there are five separate locations for code; the `core` is the last of these.

5.4 Sahana Eden

Sahana Eden is an *Open Source Humanitarian Platform* that can be used to provide solutions for Disaster Management, Development, and Environmental Management sectors. The code is hosted on *GitHub*, but it is mostly Python, not C++. If you are interested, search for it and evaluate it using the questions in Section 4.

5.5 DOxygen

Doxygen is a documentation generator, a tool for writing software reference documentation. It is mostly written in C++. Start on *OpenHub* to explore this project. The page for *Doxygen* is https://www.openhub.net/p/doxygen. Many of the questions listed in Section 4 can be answered by browsing this page. In particular, answer these questions first:

- How many lines of code does *Doxygen* have?
- What percentage of the code is comments?
- How many commits were made in the last 30 days, based on *OpenHub* statistics?
- Click on `"User & Contributor Locations"` (lower right side of screen). List some of the locations of the developers.

6 Due Date and Deliverables

This work should be completed by November 1. Answer all questions in your blog post for week 9, `2018-11-05-week09.md`. This will be a large post. Make sure that you organize it in a clear and structured way.

References

[1] Heidi J.C. Ellis, Michelle Purcell, and Gregory W. Hislop. An approach for evaluating foss projects for student participation. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, SIGCSE '12, pages 415–420, New York, NY, USA, 2012. ACM.