**COMP491 Assignment HW4: Project ranking**

This assignment is worth 60 points and should take about six hours to complete. Record your activity in real time on the Slack live log. Save the results of your work in a document that will be submitted to Moodle as PDF; this is called your *responses document*. In addition, you must add certain information to our course wiki.

**Part A: selecting and ranking 5 projects**

Part A should take 1-2 hours.

1. Read the assessments of all projects reviewed by students in our class. These are available on the class wiki “Project reviews” page.
2. Choose the five projects that seem most attractive based on the current information you have.
3. For each of the projects chosen in the previous step, read the detailed reviews of that project written by all students who reviewed it. These detailed reviews can be found in the responses documents from the previous homework assignment. The instructor will provide a link to a shared folder containing these documents.
4. Summarize your thoughts on each of the five projects very briefly: for each project, list the advantages in one sentence, and the disadvantages in another sentence. Insert these summaries into your responses document, listing each of the five projects chosen with its advantages and disadvantages (i.e. a total of two sentences per project).
5. Rank the five projects from most attractive to least attractive, taking into account all relevant factors. This includes your personal interest, the ease of contributing, the educational benefits to you, and any humanitarian aspect of the project. At this stage, no further information is required. Simply re-order your five projects, numbering them from 1 (most attractive) to 5 (least attractive).
6. For your top two choices, add five bullet points providing additional detail. This can be brief: one sentence per bullet point should be adequate. The five bullet points should cover the following topics:

* **Community**: *How would it be to work within this project’s community?* Consider issues such as the size and diversity of the user and developer communities, the availability and variety of communication channels, the quality and tone of communications, how newcomers are treated, etc.
* **Complexity**: *How technically hard is it going to be to work on this project?* Consider issues such as the size of the code base, the number of different tools/languages/technologies/frameworks used, degree and quality of documentation (both in code and in other documents), the modularity of the project (can you isolate what you have to know?), etc.
* **Activity**: *How active is this project?* Consider issues such as the recent responsiveness of community members, the rate at which the code is changing (is it too slow or too fast?), whether new issues are being opened and old issue are being closed, whether pull requests are being merged, how up to date documentations is, etc.
* **Approachability**: *How hard would it be to get started with this project?* Consider the availability, quality and complexity of install/setup directions and documents explaining how to get started as a contributor, the number of new contributors that have joined the project recently, indications and understandability of good first issues, alignment of your team’s current skill set with the projects tools/languages/technologies/frameworks, how much domain knowledge is necessary, etc.
* **Appeal**: *How interested is your team in this project?* Consider issues such as the application domain, the benefits to end users, the technologies employed, etc.

**Part B: Install spikes**

Part B should take 3-4 hours. Note the following definitions, which will be used in the instructions below:

* *Project #1* refers to the project you ranked first in Part A.
* *Project #2* refers to the project you ranked second in Part A.
* A *user install* of a project is an installation for using the product. For some projects, the user install is minimal or nonexistent. For example, an H/FOSS project that is available to users as a public website does not have any meaningful user install process.
* A *developer install* of a project is an installation for altering or improving the product. This will typically involve: downloading source code, software frameworks and libraries; compiling the source; deploying and executing the compiled product. The difficulty of developer installs varies enormously between projects. Some developer installs can be completed within a few minutes. For certain projects, however, it has required days or weeks to complete a developer install. The instructions below take account of this variability.

The tasks for Part B are as follows. As with all parts of this project, log your activities in the Slack live log every 30 minutes to provide evidence of your effort.

1. Perform a user install of Project #1. In your responses document, make a note of any difficulties encountered. Experiment with using the product for 10-20 minutes, making brief notes of any successes and failures in your responses document.
2. Repeat the previous step for Project #2.
3. Attempt a developer install of Project #1. Recall that the total amount of time spent on this assignment should be about six hours. Therefore, if you encounter difficulties during the developer install, it is acceptable to stop after a few hours: you may limit your total amount of time spent on this assignment to six hours. Regardless of how much progress you achieve, make a note of any successes, failures, and difficulties in your responses document.
4. If the total amount of time spent on the assignment so far is less than six hours, repeat the previous step for Project #2.

**Part C: revisit your ranking**

Attempting a user and developer install can have a major impact on your view of a project. Therefore, please stop at this point and re-examine your answers to Part A. If you would like to change your ranking, you can do that now. Describe any changes clearly so that the instructor can understand your initial choices of ranking and the revised ranking. If you now have a new project in the #1 or #2 position, there is no need to attempt a user or developer install. These will be addressed in a future assignment, if necessary.

**Part D: install notes on wiki**

Go to the “Project installation notes” page on the class wiki. If necessary, create new pages here for your Project #1 and Project #2. Paste your notes from Part B into these pages, following the fake examples provided.

**Part E: preferred teammates**

If there are specific people you prefer to work with for your capstone project, please list them on the confidential survey provided by the instructor. It is not possible to give a ranked list. You may list any number of preferred teammates, including zero.

If there are specific people you do not wish to work with, please communicate this to the instructor privately. You can do this by email, but it may be preferable to meet with the instructor.

**Submission**

* Submit your responses document to Moodle.
* Create or edit the “Project installation notes” wiki page for your Project #1 and Project #2, as described in the Part D instructions above.
* Optionally, submit your preferred teammates as described in Part E above.

**Rubric**

A good grade can be achieved by: following all instructions accurately; producing material that reflects about six hours of effort for this assignment (with evidence of time spent on the Slack live log); employing correct grammar and clear presentation of concepts. For an excellent grade, some evidence of additional research, insight, or thoughtfulness may be required.

**Acknowledgements**

This assignment builds from and adapts ideas and content from the following activities created by others:

* Professor Braught’s version of this assignment
* https://github.com/ChrisMurphyOnline/open-source-software-development-course/blob/master/activities/foss-evaluation-activity.txt
* http://foss2serve.org/index.php/Project\_Evaluation\_Activity\_V2