ere are som comments about what I've gathered so far.

Data selection process seems invalid. If data is invalid, the whole analysis will be invalid. The two main reasons why I think data is invalid are:

1. Measuring how sustainable a project is as a function of development time is not an accurate measure.

There may be projects that are sustained for a long time because they have corporate sponsors even though they had low quality code, and there may also be projects that had great quality code but died because the developers could not afford to continue working on them.

// I’m totally agree with you and this what we try to prove so we include this sentence anyway in the conclusion

Unfortunately project sustainability is very hard to measure objectively. So we could continue with this way, knowing that there's this handicap, or we can incorporate other factors like number of time the project has been cloned, number of times the project has been "starred", number of contributor addition over time, commit acceleration into the repository, etc. These metrics indicate that the project is being useful to others, and that people want to contribute to the project. If we take the commit acceleration into account and find that new code is created instead of a lot of refactoring, we could argue that the project is well designed as it allows an accelerated commit rate (as succesful open source projects do) and so good quality code improves sustainability. However we can not get this information from looking only into the date range for commits.

// Correct , this what I want to do in second step as I told you but I can’t do it now since I want to do this first as a first step and I have to show them What correlation and relation between these metrics and S

The metrics I mentioned can be extracted from the repository itself. Also, we are able to find which files tend to be more problematic (people don't want to change them) or stable (they don't need too much change because they were general enough), through analyzing the commits. Unfortunately this could take up to months of work.

// correct

2. You included many repositories for which the main measure (sustainability, measured as the range between the first and last commits), is not valid.

As you pointed out in the document, there are repositories that have a lifespan of 0 days because they were just "put into" a Git repository, even though they may be pretty big. This introduces a big bias into the data and voids any results from the analysis that include that data.

The solution is to keep only repositories that were started within a Git repository since day one. This may be very hard to verify. I suggest we use a combination of the two points I mentioned. I suggest we incorporate more metrics for "sustainability" other than the day range of activity, and I suggest we filter out repositories we can not validate we have the real metrics for (like not having full lifespan).

// I will say that in the discussion part of the report that some developers put their code into GitHub to save it and this why I measured 0 days

For each repository, I restive the first commit and last commit then I get S value

I found some project they have started before 2009 but have not added to GitHub till 2009 so I exclude them. And I excluded other also for other reasons. So all 3113 projects are created during 2009 and it first commit is in 2009

I wrote a script to do that and calculate S metric for each project

If we don't do that, I do no think the analysis will be objective and valid. However, since there may be not enough time, then we can just continue using the metric you propose but mention that there's more research coming up that uses other metrics to measure sustainability, and filter out those repositories we do not have full lifespan for (at least make sure they are not in the projects selected for the metric computation part (the ones I'll use for the analysis), but doing this by hand is not statistically correct).

// I use a tool ( script ) that calculated code quality metrics and S metric

Another thing is that, the more metrics you can compute about the code, the better analysis I can make, so if you chose for some reason only those 6 metrics, but you are able to compute others, please do so.

//we chose these 6 metrics since each one measure specific thing of code quality.

Some minor questions are:

1. Why did you decide on using 19 segments for the sustainability metric?

Based on projects distribution among the period, Im not be allowed to change this. This is a request from them.

1. Why did you select 2009 as the cutoff point for the start date?

The first year when github become popular

I'm not sure if this will be useful for you, but Google recently made all of GitHub's repositories available for analysis in Google Cloud through Big Query. Maybe it's a place where you could get the data from (I have not used it so I am not certain it will be useful).

- https://cloudplatform.googleblog.com/2016/06/GitHub-on-BigQuery-analyze-all-the-open-source-code.html

- https://www.githubarchive.org/

- https://github.com/blog/2201-making-open-source-data-more-available

- https://cloud.google.com/bigquery/public-data/github

There are other questions, but those should wait until we figure out what I mentioned before. Please let me know how you want to procede. Thanks.

Sorry I did not understand this part

More explanation :

- I have been asked to divide S into 19 segments

S1- 0 days

S2- 1-7 days

S3 8-97 days

etc... done it like  based on projects distribution among the period.

* I retrieve  projects  information from github that started in 2009 and I calculated S = Last commit- first commit in days

then I got a range for S which is from 0-2752 days as I think

then I asked to segment this period into 19 seg as I explained

then from each seg I have been asked to select 7 projects randomly So I wrote a script

to picke 7 projects  and I performed code quality analysis on it one by one using specific tool.

So IF I want to increase the sample later on I will do it easily since I script wont select the same projects again so what projects I have analysed so I will add different projects

then I have been asked to take average of average ( as I explained when I sent data ) then perform the statistical analysis

so I suppose to do statistical analysis on excel file called  ( code quality analysis) only

in order to find a relation between each metric and S

S vs LOC   S vs DIT ..etc

then All code quality  metrics vs S

But Im not quite confident  if   average of average is a correct way so I send all data to u and tell u what I want

but I just want to find a correlation between each metric and S

not between individual project and S