

## ① Introduction

Continuous Integration is a good practice that increases productivity and ensures early bug detection [1, 2]. However, it is still uncertain how CI should be implemented depending on the context of the project. If we find descriptive metrics, we can provide a way for maturing the CI process.

**Our goal:** find descriptive metrics to describe project activity.

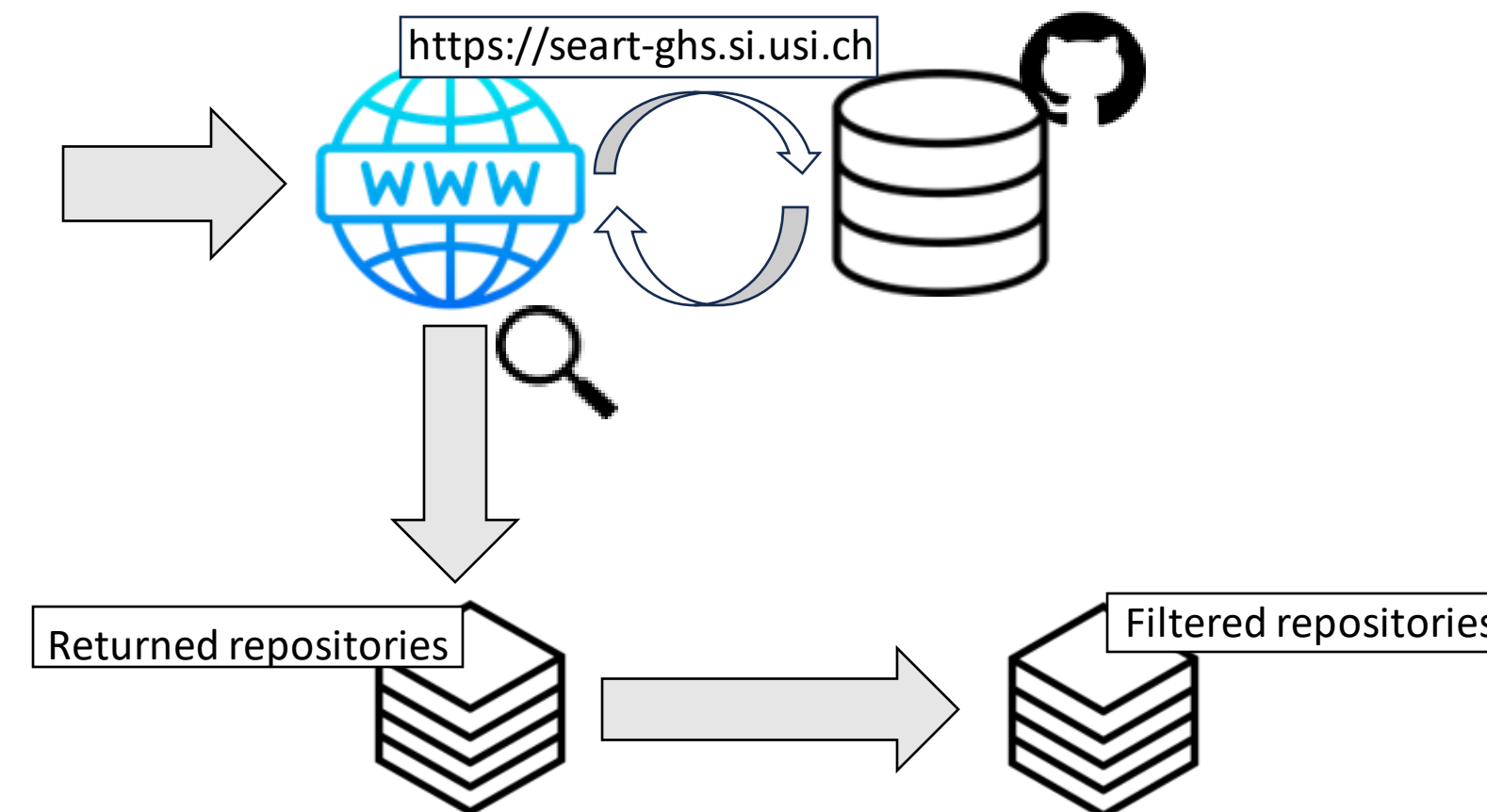
## ② Research Questions

- What metrics can be extracted to describe project activity?
  - **RQ1:** What metrics do other researchers use?
  - **RQ2:** What additional metrics can be used to describe project activity

## ③ Method

### 3.1 Project selection

Select 500 random repositories from GitHub Search, filtered on CI and with the criteria of 100 repositories per main programming language.



Filter the repositories on whether they used CI and selected 500 random projects with 100 projects per language.

Figure 1: Visualization of the project selection process.

### 3.2 Metric selection

In the literature review, five default metrics were obtained to answer **RQ1**. Then, **RQ2** was answered with several techniques, correlation analysis with the metrics from **RQ1** among one of them.

## ④ Results

Clustering all the 500 repositories on their main language<sup>1</sup>, you can see that Java a more active language than the other 4.

<sup>1</sup>Main language is decided as the language with the most lines of code (LOC) throughout the whole repository.

Language	avg. # commits	avg. # closed pull req.	avg. # contributors	avg. # issues	avg. # releases
C	1103.97	34.52	23.53	93.1	7.49
Java	4142.56	1241.86	129.6	2570.47	41.27
Python	191.67	23.98	7.05	50.69	4.96
C#	415.63	32.73	11.16	59.64	8.45
Swift	181.19	34.26	7.19	64.81	9.01

Table 1: Result of performing correlation analysis between the repository size and the five default metrics.

## ⑤ Conclusions

The five default metrics can be used to describe project activity, and on top of that: the release date, the main programming language, and the repo size.

## ⑦ References

- [1] Ravi Teja Yarlagadda. Understanding devops & bridging the gap from continuous integration to continuous delivery. *Understanding DevOps & Bridging the Gap from Continuous Integration to Continuous Delivery*.
- [2] Moritz Beller, Georgios Gousios, and Andy Zaidman. Oops, my tests broke the build: An explorative analysis of travis ci with github.

## ⑥ Limitations

- **Limited metrics:** further research is needed to consider more metrics for project activity analysis.
- **Small sample size:** Only 500 repositories were used in this study. Certain results could have been missed due to this low amount.