

### 3) Little Code Reuse

Belief: Carver et al. [20, 24, 26, 27] warn that there is little code reuse in CSc projects

Notes: Scientific developers have a history of not adopting or re-using the software developed by others (or even their own), e.g., linear algebra software [14]. They say this is due to:

- The structural assumptions from the others would be too strict and narrow [13, 20].
- Most of the software is not built with comprehensibility requirements as the top priorities [24] (which is doubted previously in III-C2). Hence, adapting old code for new domains is difficult.
- CSc scientists believe that their time and efforts can be conserved by being spent on implementing new libraries & framework rather than understanding existing ones.

Modeling Assumptions: Reuse libraries here encompass reuse scientific software. To measure reuse, we can record the code called via libraries that come from outside of a repository. This is to say that the amount of external imports (EI) and files that have external imports (FEI) are indications of the reuse activities within the software. There are four attributes for this that we define below. For all of them, the higher the value the better reuse within their projects:

- $EI\_per\_File = EI / Total\_number\_of\_Files$
- $ILOC\_Ratio = EI / LOC$  (total number lines of code)
- $FF\_Ratio = FEI / Total\_number\_of\_Files$
- $II\_Ratio = EI / Total\_number\_of\_Imports$

Prediction: CSc projects have less reuse than SE projects if the above ratios are lower to CSc than SE.

Observed: Table VIII summarizes the median and interquartile range for reuse metrics in both CSc and SE projects. The lines of code reuse are low for CSc projects (just 10%) but its nearly the same as SE projects (13%). In fact, after applying a Scott-Knott test, we can report that the SE projects are statistically indistinguishable from CSc projects, on all the metrics of Table VIII. That, in this sample, we found no difference in the reuse rates of SE and CSc code. Hence:

TABLE VIII  
MEDIAN AND IQR SUMMARY FOR  
FOUR ATTRIBUTES PORTRAYING THE  
REUSE STATE OF CSc AND SE  
PROJECTS. IQR IS THE DELTA BETWEEN  
THE 75TH AND 25TH PERCENTILE.

Metric	Project	Median	IQR
EI per File	CSc	3.2	1.6
	SE	2.9	1.6
ILOC Ratio	SE	13‰	9‰
	CSc	10‰	8‰
FF Ratio	CSc	86%	19%
	SE	81%	19%
II Ratio	SE	70%	27%
	CSc	55%	19%

Conclusion: We **doubt** that CSc reuses less code than SE.

Discussion: The ratios used here only reflect code reuse. Other kinds of reuse include the design or conceptual reuse. Also missed by the above ratios is non-verbatim reuse (where code is reused, but modified). Further, the above ratios may miss certain important code measures (e.g., text-based, token-based, tree-based, metric-based, semantic and hybrid).

We did not explore those additional measures of reuse since their implementation leads to  $O(n^{(m-1)})$  complexity with

$n$  as the current section of codes within the project and  $m$  is the number of the projects to compare to. We hence leave reuse measurement in CSc to future work.