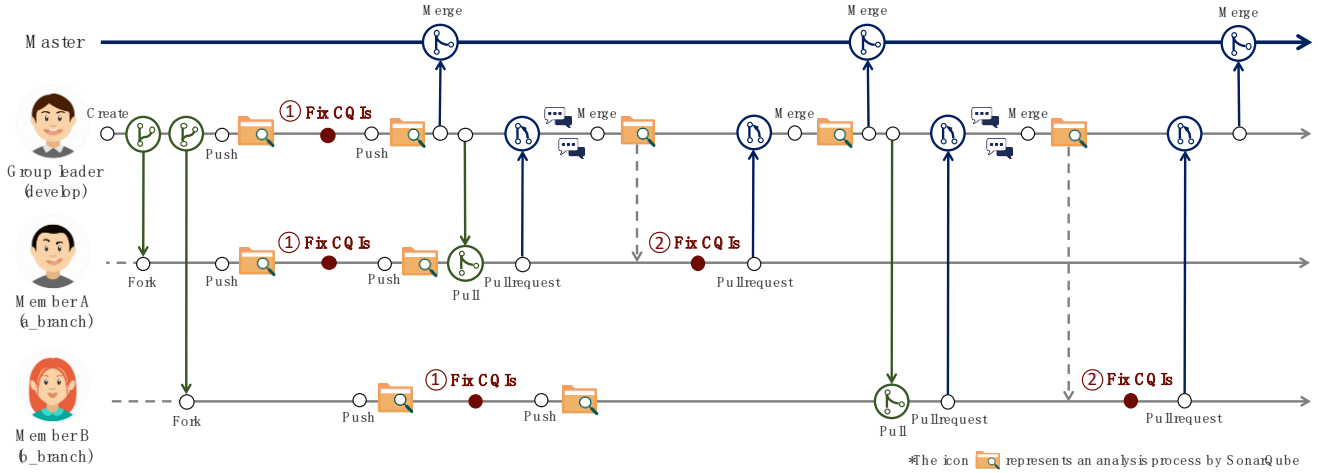
Improving students' programming quality with the continuous inspection process: a social coding perspective

Yao LU, Xinjun MAO, Tao WANG, Gang YIN, Zude LI

Frontiers of Computer Science, DOI: 10.1007/s11704-019-9023-2

Problems & Ideas

- Both the teaching and learning of high-quality programming skills are challenging in most college education systems.
  - Students usually cannot sufficiently master programming skills through the ’knowledge-inculcation’ model of the instructors.
  - When students conduct programming assignments or projects, teachers usually do not have effective programming-quality improvement and assessment methods.
- Ideas: Continuous Inspection

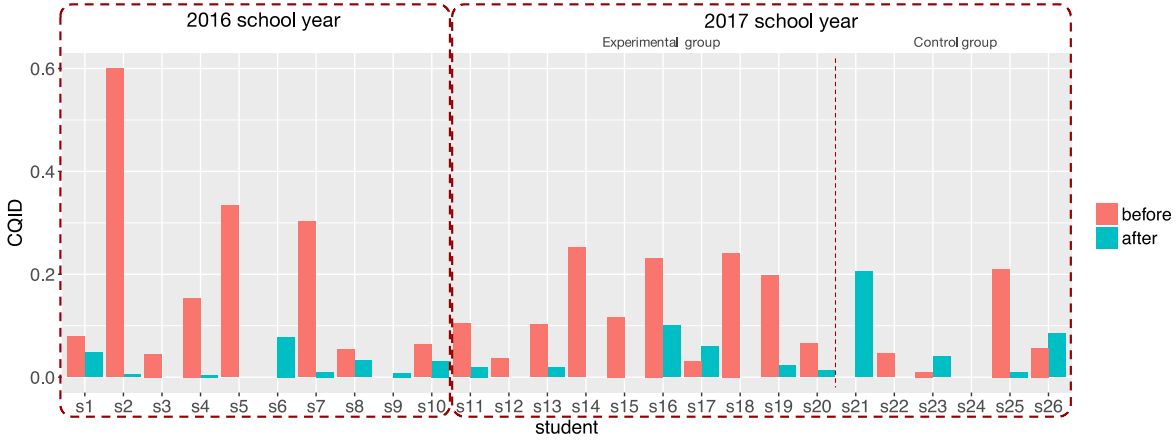


The continuous inspection process for students' collaborative projects

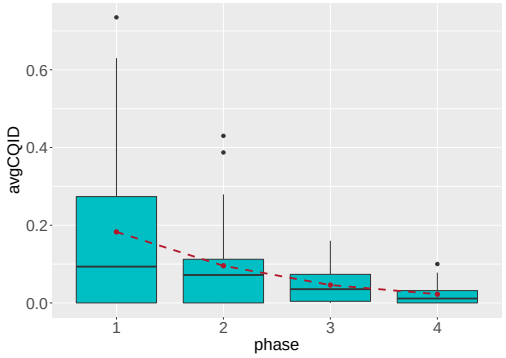
  - Each student is responsible for fixing the CQIs of her modules on her own branch.
  - The CQIs should be fixed within 24 hours after being reported by SonarQube.
  - All CQIs on the *master* branch should be fixed or closed. If a CQI is marked as ‘*won't fix*’, reasons should be given as comments in SonarQube.

Main Contributions

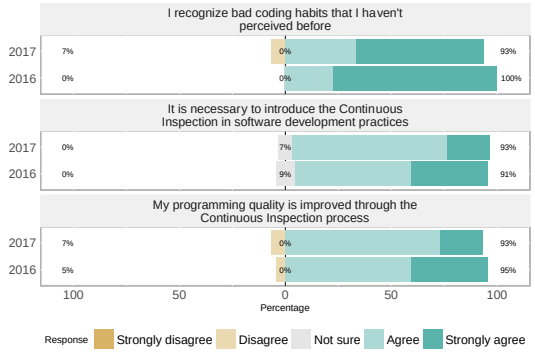
- **We find that continuous inspection can help students to identify their bad coding habits, master a set of good coding rules and significantly reduce their introduced CQIDs, thereby improving their programming quality.**



Students' introduced CQIDs before and after adopting continuous inspection



The distribution of students' average CQIDs in the four phases.



The students' perceptions of the process