The Open-code Clinic
Software quality assessment in the classroom

SIG
Software Improvement Group

Quality of student code?

- How do you objectively assess the quality of the code your students write?
- How do you make it clear to them that maintainability is important?
- Is there a way to look at maintainability without sounding subjective?

	Volume	Duplication	Unit size	Unit complexity	Unit interfacing	Module coupling	Component balance	Component independence
Analysability		X	X	X				X	
Modifiability			X		X		X		
Testability		X			X				X
Modularity							X	X	X
Reusability				X		X			

The SIG software maintainability model is based on the ISO/IEC 25010 standard 'System and Software Product Quality Model'.
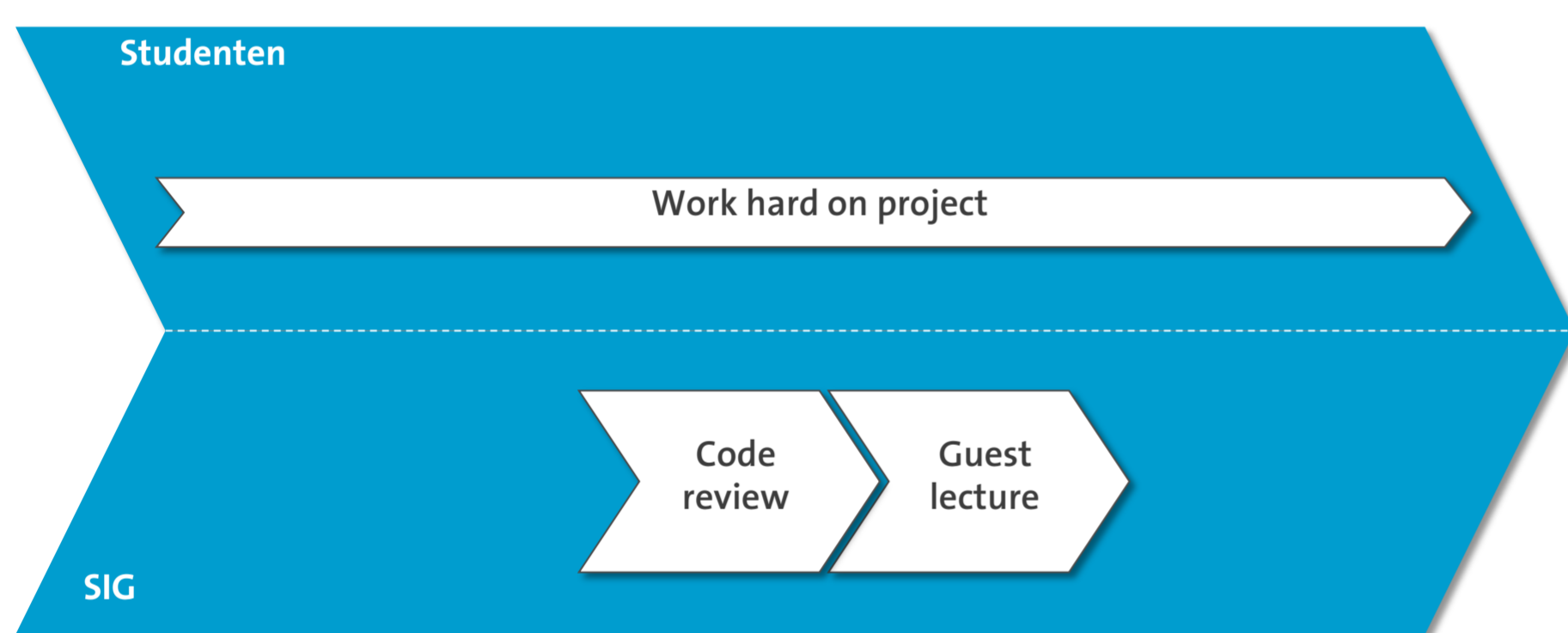
Yes there is!

The SIG benchmark model for maintainability is used to assess hundreds of industry systems every year, for a wide range of companies vital to the Dutch economy.
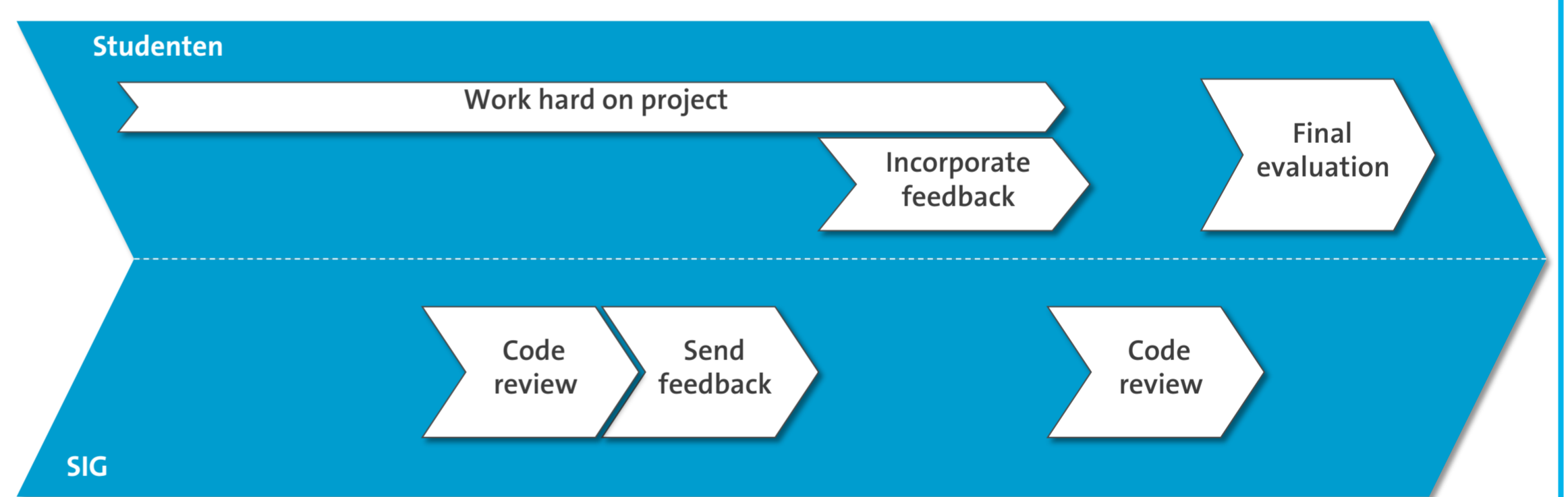
Assessment Process

The Open-code Clinic has been performed in the past 5 years to assess the quality of student projects for both Technical University Delft and Radboud University Nijmegen. It has been used in two forms:.
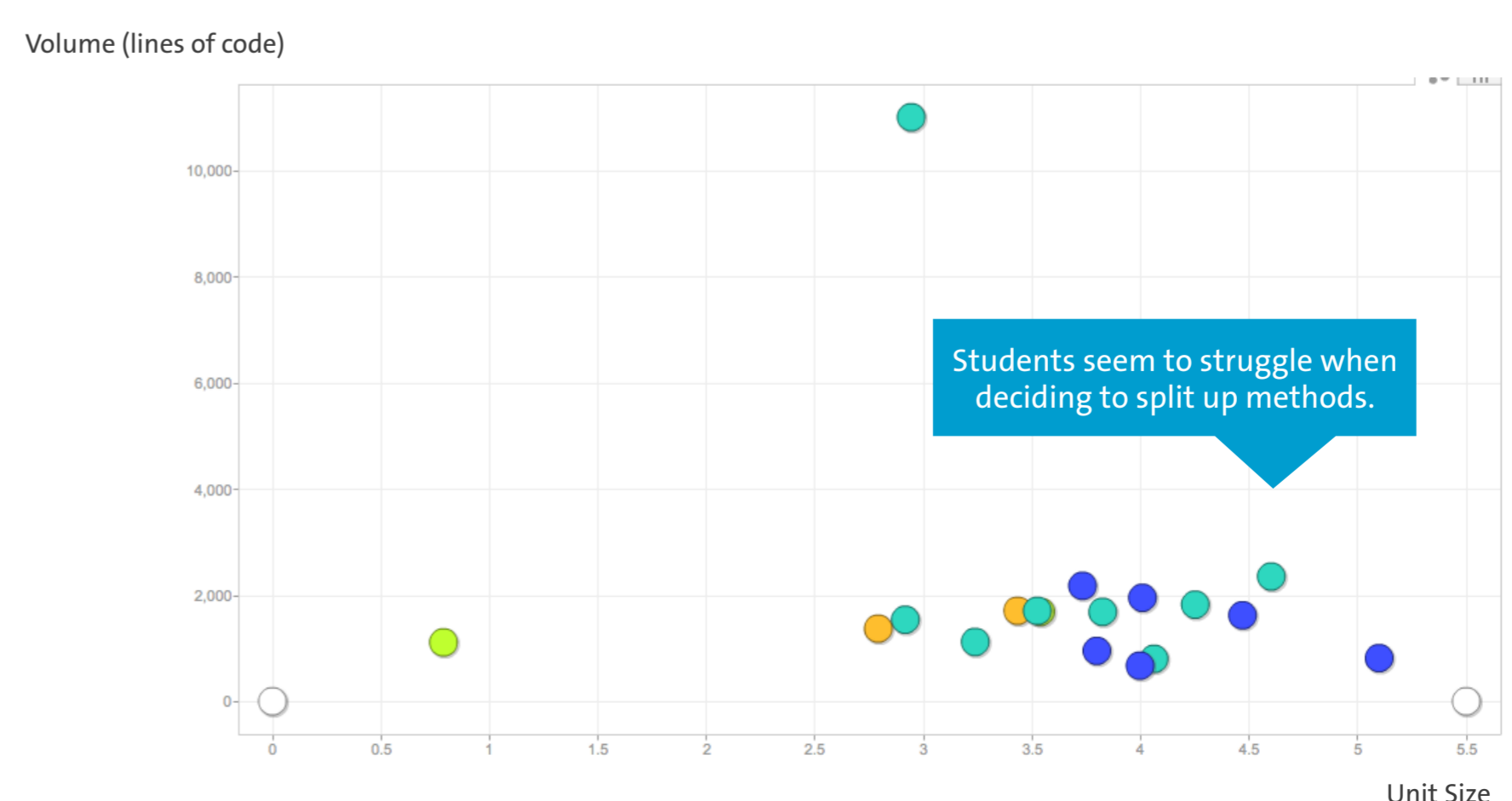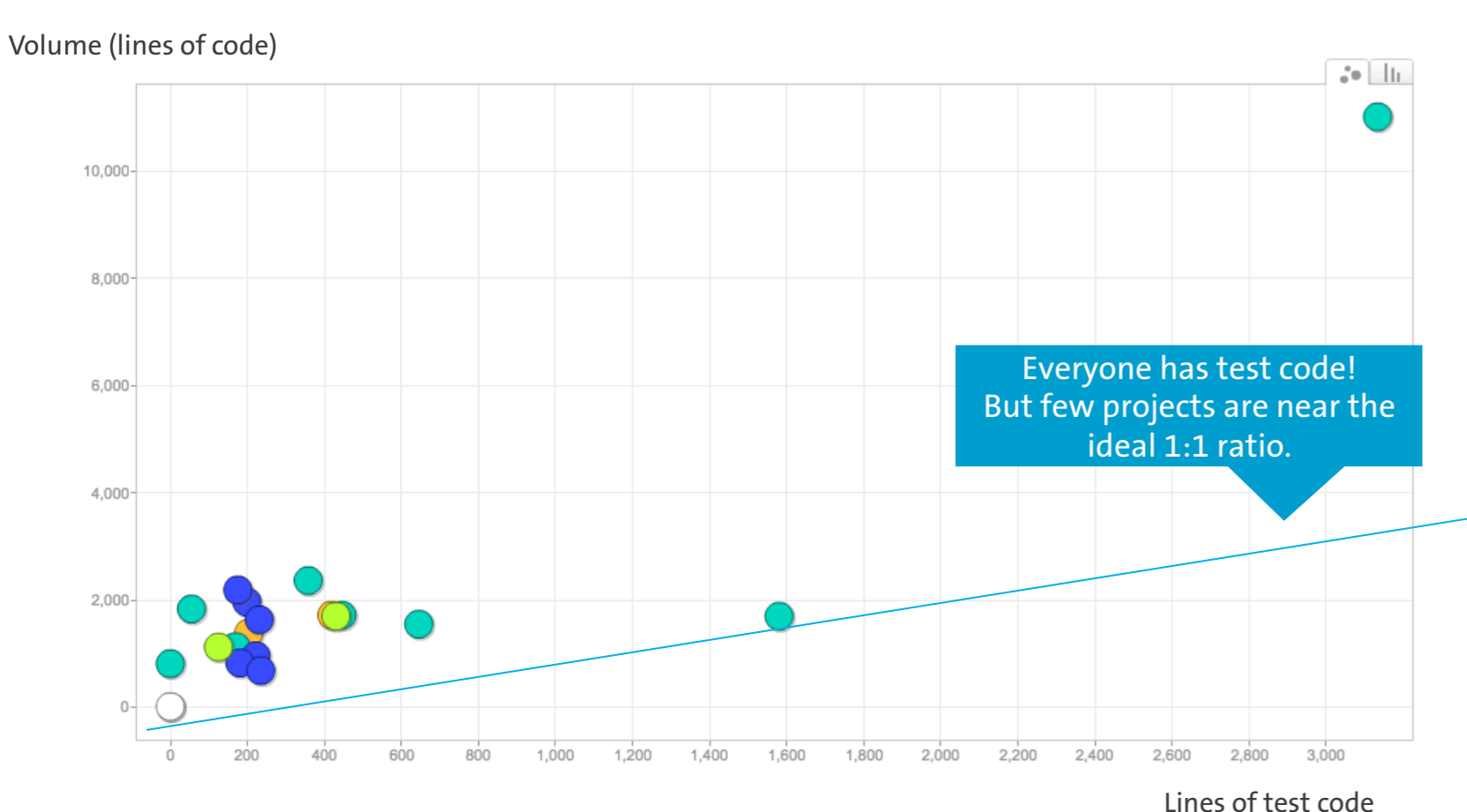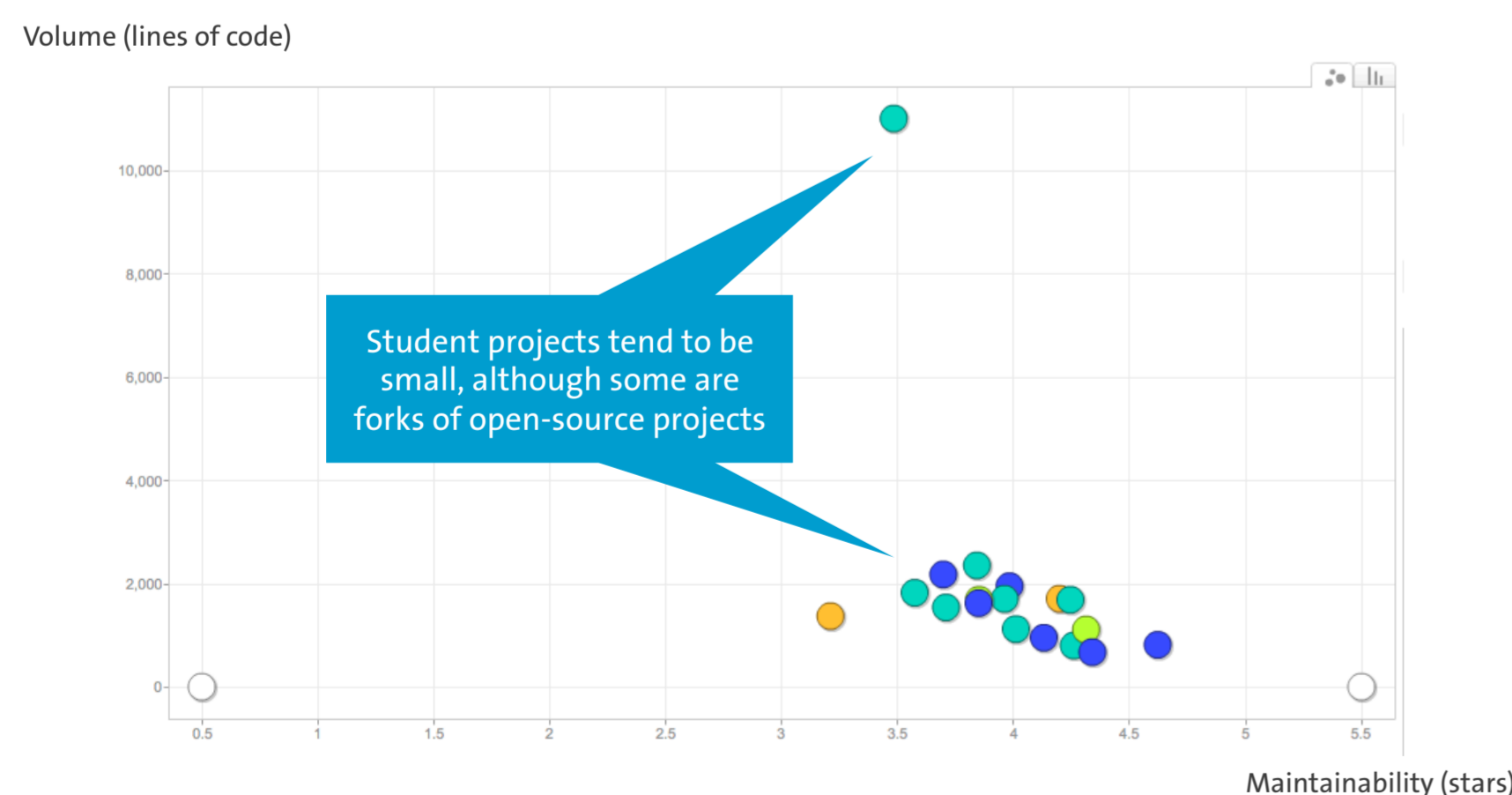

Studenten — Work hard on project
SIG — Code review → Guest lecture
One-phase Assessment

Studenten — Work hard on project → Incorporate feedback → Final evaluation
SIG — Code review → Send feedback → Code review
Two-phase Assessment

Assessment Results

Volume (lines of code)

Student projects tend to be small, although some are forks of open-source projects

Maintainability (stars)

Statistics since 2010:

- Evaluated groups: 175
- Main technologies: Java, JavaScript, C#, PHP, Python, C++
- Most problems with: Unit Size, Duplication
- Most common scores: ★★★☆☆, ★★★★☆
- Percentage of applied feedback: 70%

Volume (lines of code)

Everyone has test code! But few projects are near the ideal 1:1 ratio.

Lines of test code

Volume (lines of code)

Students seem to struggle when deciding to split up methods.

Unit Size