

Developing Automatic Methods for Teaching Code Quality in Introductory Programming

Anna Řechtáčková, Ph.D. candidate

anna.rechtackova@mail.muni.cz

Faculty of Informatics, Masaryk University, Czechia

Background

- code quality is desirable for programming practice; reflects student's mastery
- teaching to write quality code is not easy
 - manual reviews are resource-intensive and scale poorly
 - existing automated tools are insufficient

Contributions

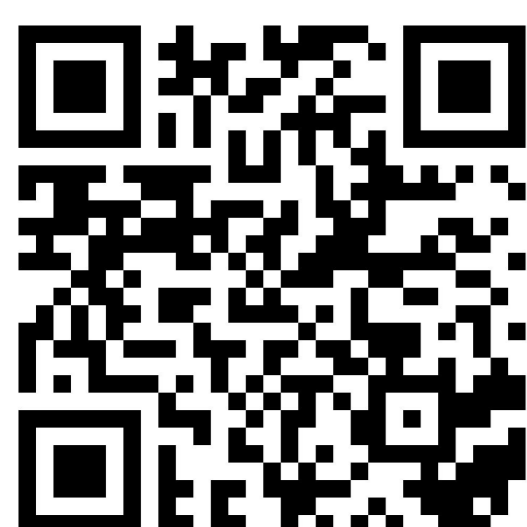
Catalog of code quality defects

- 80 code quality defects
- categorized for easier aggregate analysis
- defect severity set based on survey with 72 participants
- published in proceedings of ITiCSE '24
- still being extended

defect name	description	code example	prog. construct	defect type	tool support	lang. indep.	severity	prevalence	EduLint code
duplicate block	Two very similar blocks of code.	<pre>if c: f(1); g(1); f(1) else: f(2); g(2); f(2)</pre>	function	duplicate code		1	1	5	3 R6505
redundant if-else	If-else statement with both branches being return True/False.	<pre>if c: return True else: return False</pre>	condition	simplifiable		3	1	5	3 R6201
one-letter name	One-letter variable name (with several exceptions like i, j, and n).	l = 4	variable	poor name		1	1	4	3

EduLint

- an open-source tool for checking Python code
- gives code quality feedback to novice programmers
- available through PIP, web interface, API



scan for:

- defect relevance survey
- code quality defect catalog and paper on it
- EduLint analysis example
- contact information

Student code is **really bad!**

```
def impose_fine(age, beer):
    if age < 21 and beer == True:
        return True
    else:
        return False
```

can be improved

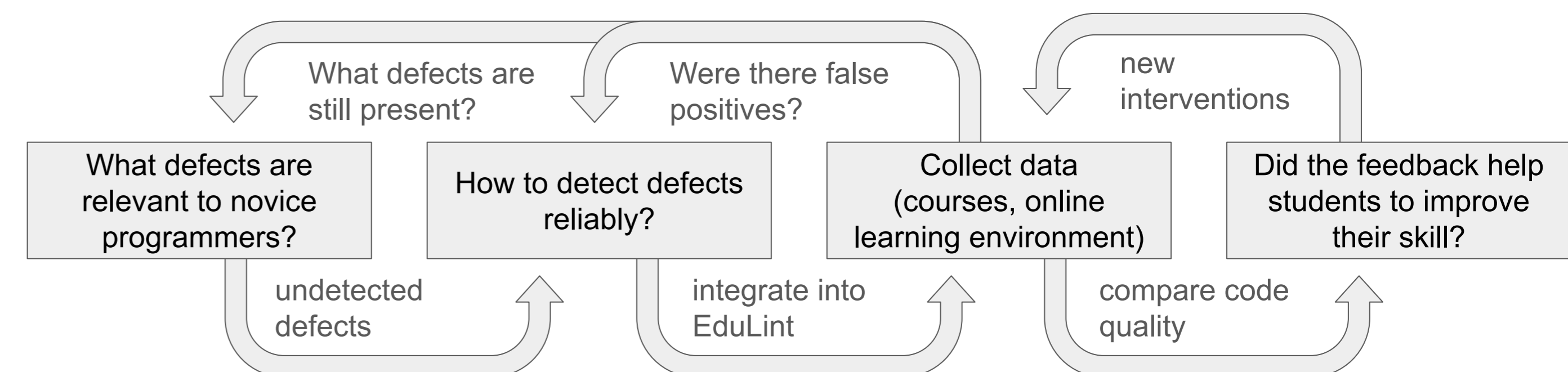
How can we teach students to write **better quality code?**

Our plan:

- ✓ create a **catalog of code quality defects**
- ⚠ create a **dataset of student code labelled with feedback**
- ✓ develop **automatic defect detectors** and make them open-source
- ⚠ study **how student coding changes** after they receive feedback

Current work

- extending the defect catalog
- **surveying educators for opinion on defect severity (participate by scanning the QR code)**
- developing detectors for duplicate code



My questions for you

- What would you need from a code quality feedback providing tool to use it in class?
- How would you study impact of such tool on students' ability?
- What use would you have for a dataset of cca 100 code snippets manually labelled with possible feedback?