Open Universiteit
www.ou.nl

Philipps Universität Marburg

Christoph Bockisch
Steffen Dick

QPED

Quality-focused Programming Education

Digitale Kaffeerunde

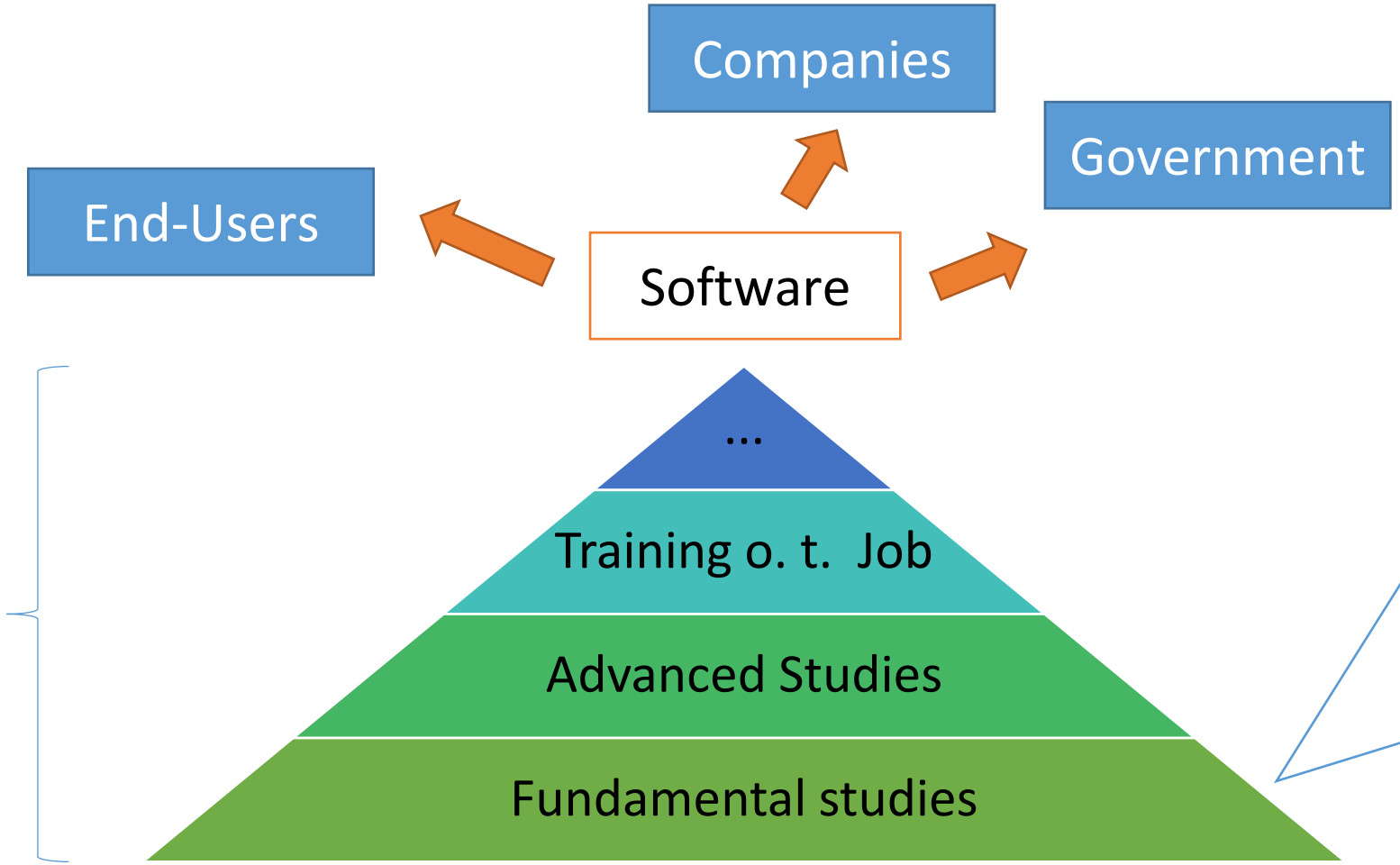
Universitat Oberta de Catalunya

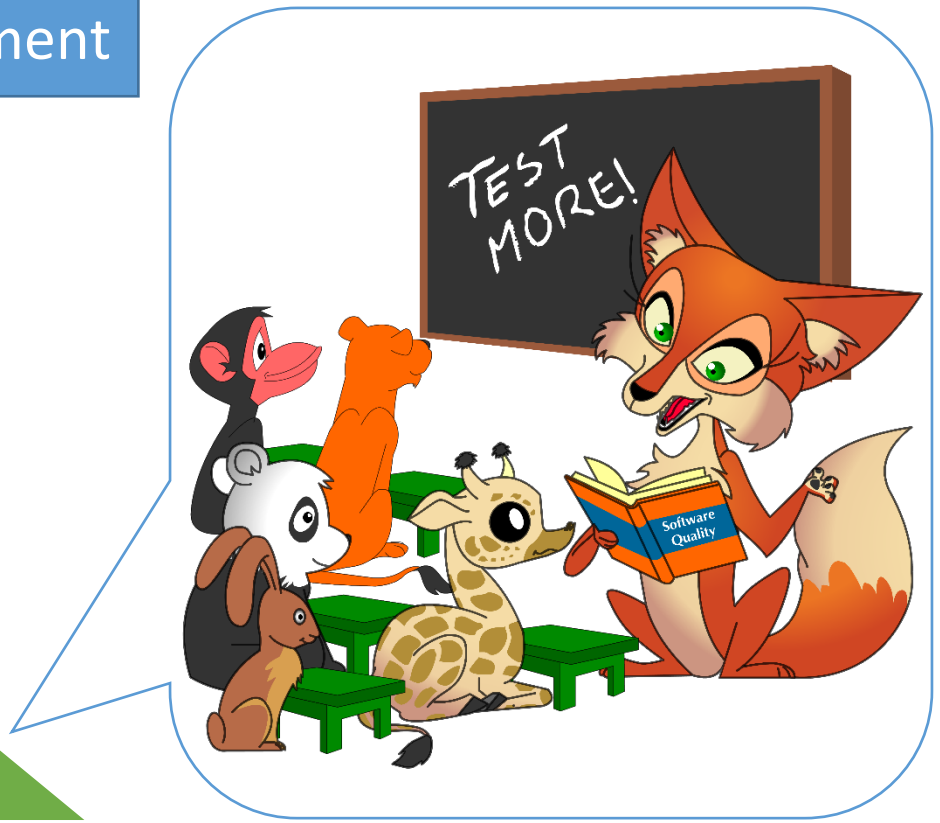
TU/e

Quarterfall

Co-funded by the Erasmus+ Programme of the European Union
QPED 2020-1-NL01-KA203-064626

Influence of Computer Science Education





Making Students More Quality-Aware

- **MORE** testing and documentation in education
- **EARLIER** testing in education
- **HABITUAL** testing and documentation
- Make it **FUN** and set an **EXAMPLE**

But our curriculum is already full.

But students don't know enough in the beginning.

But we don't have the resources for constant reminders.

But we have no time to update our materials.

Making Students More Quality-Aware

- **MORE** testing and documentation in education
- **EARLIER** testing in education
- **HABITUAL** testing and documentation
- Make it **FUN** and set an **EXAMPLE**

QPED wants to help.

QPED Project – Multiplier Event

- Erasmus+
  Strategic Partnership
- Five partners
- Germany, The Netherlands, Spain
- One traditional univeristy, Two distant teaching universities, one SME

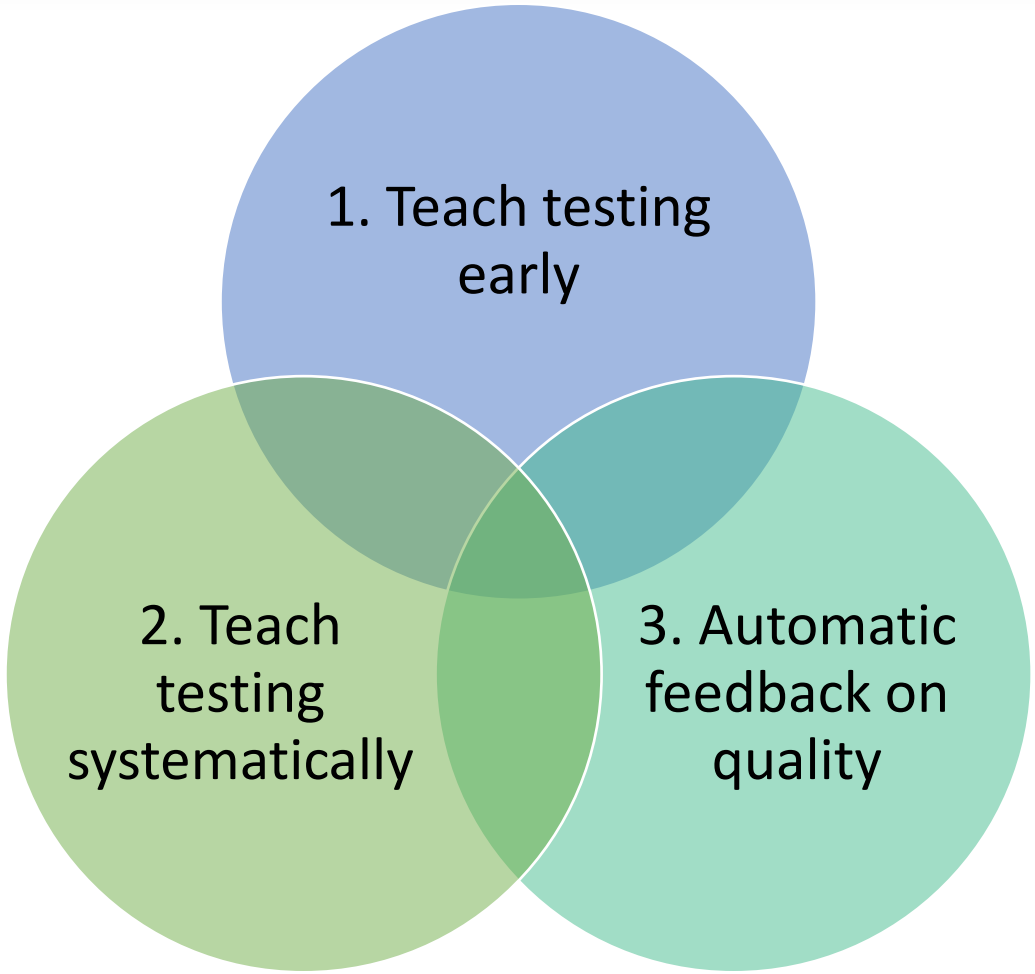
QPED Homepage

https://qped.eu

QPED Newsletter

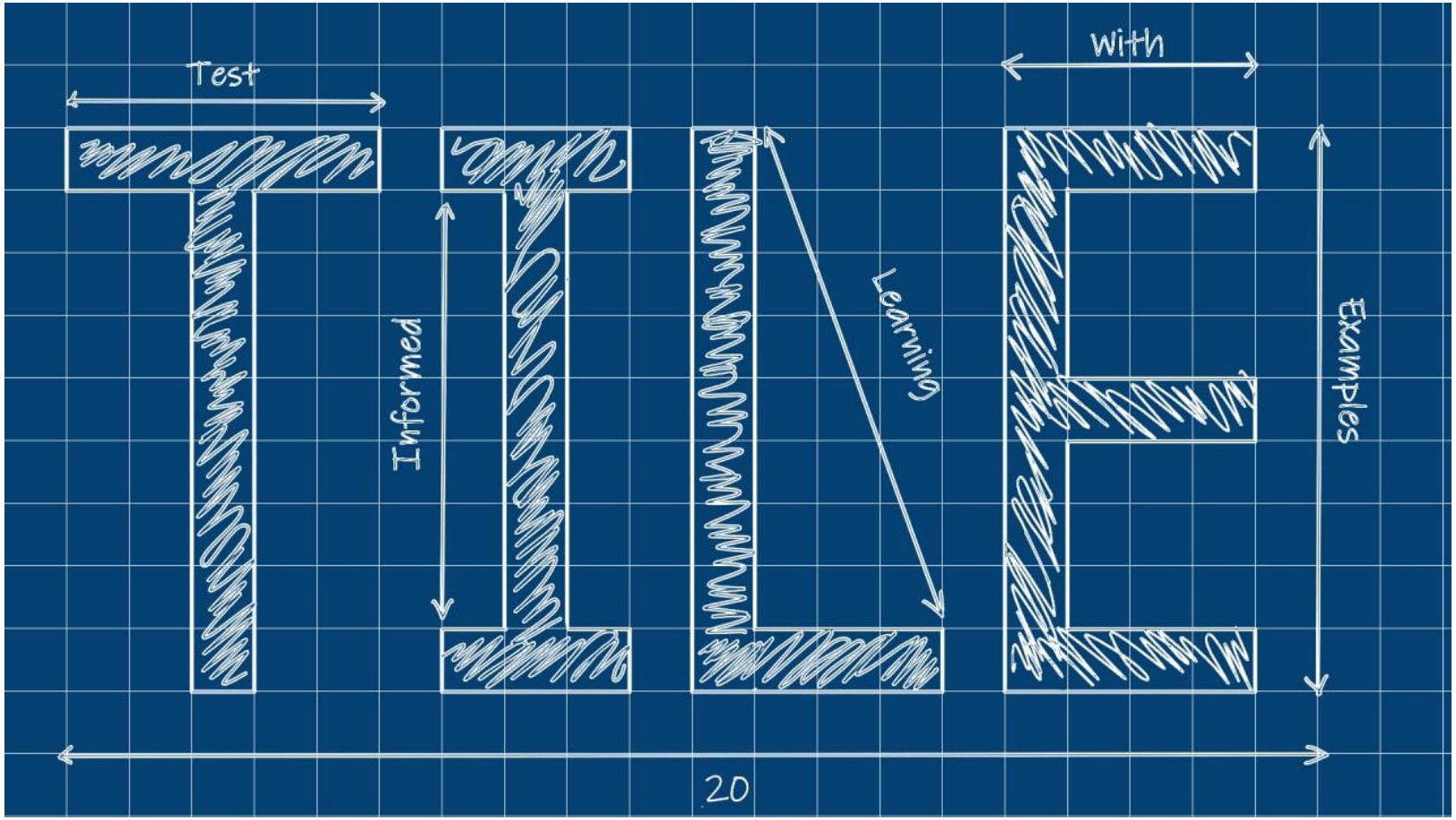
https://listserv.dfn.de/sympa/subscribe/qped-info

Quality-focused Programming Education



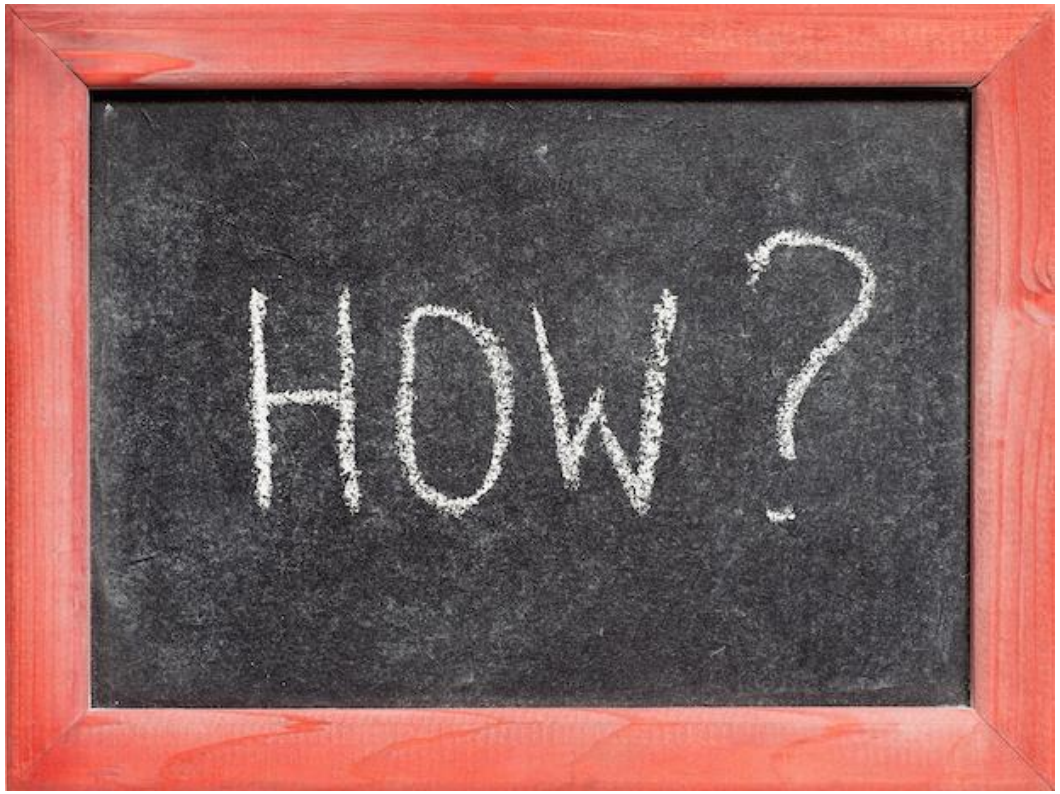
4. Evaluate Effectiveness

1. TILE - Test Informed Learning with Examples



TILE - Test Informed Learning with Examples

- **Early** - introduce testing from the very first program they see and write
- **Seamless** - as an inherent part of programming, and *not* as a separate activity
- **Subtle** – using clever and indirect methods



Learning by examples

- Common way of introducing programming concepts to students:

  1. We explain the concept(s) using an **example**.
  2. We show **example** executions to see what comes out.
  3. We let students practice the concepts with exercises about **example** problems.

No extra lecture time needed.

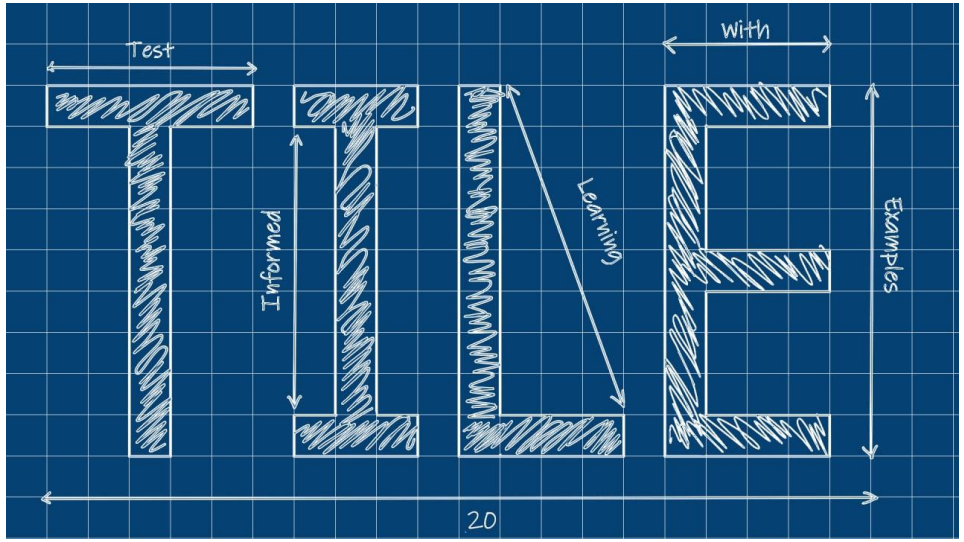
- TILE will relate these **examples** to testing

- Four types of TILES are distinguished:

  1. test run examples
  2. test cases examples
  3. Test message TILES
  4. test domain examples

TILE - Repository

- Reusable, worked out examples
  - Over 100 TILEs developed
  - Still growing
- Available at: https://tile-repository.github.io
- Categorized
  - TILE aspects
  - Topics
  - Technology used
  - Audience
  - Programming learning goals
  - Testing learning goals
  - Prerequisites



Reuse our teaching material.

2. Procedural Guidance - Motivation

- Several studies demonstrate experiences:
- S. Edwards and Z. Shams, “Do student programmers all tend to write the same software tests?” in Proceedings of the 2014 conference on Innovation & technology in computer science education. ACM, 2014, pp. 171–176.

  → ~101 implementations, ~2500 failures, only 12 % bugs found
- M. Lawende, H. Passier, G. Alpár. Reproduction for Insight: Towards Better Understanding the Quality of Students Tests. In Proceedings of ITiCSE 2021. ITiCSE 2021

  → Same findings
- A. Bijlsma, N. Doorn, H. Passier, H. Pootjes, S. Stuurman. How do students test software units? ICSE 2021 – JSEET

  → Students perform only happy path tests
- Y. Kolikant, “Students’ alternative standards for correctness,” in Proceedings of the first international workshop on Computing education research. ACM, 2005, pp. 37–43

  → Attitude: “It compiles and there is console output, so we are ready!”

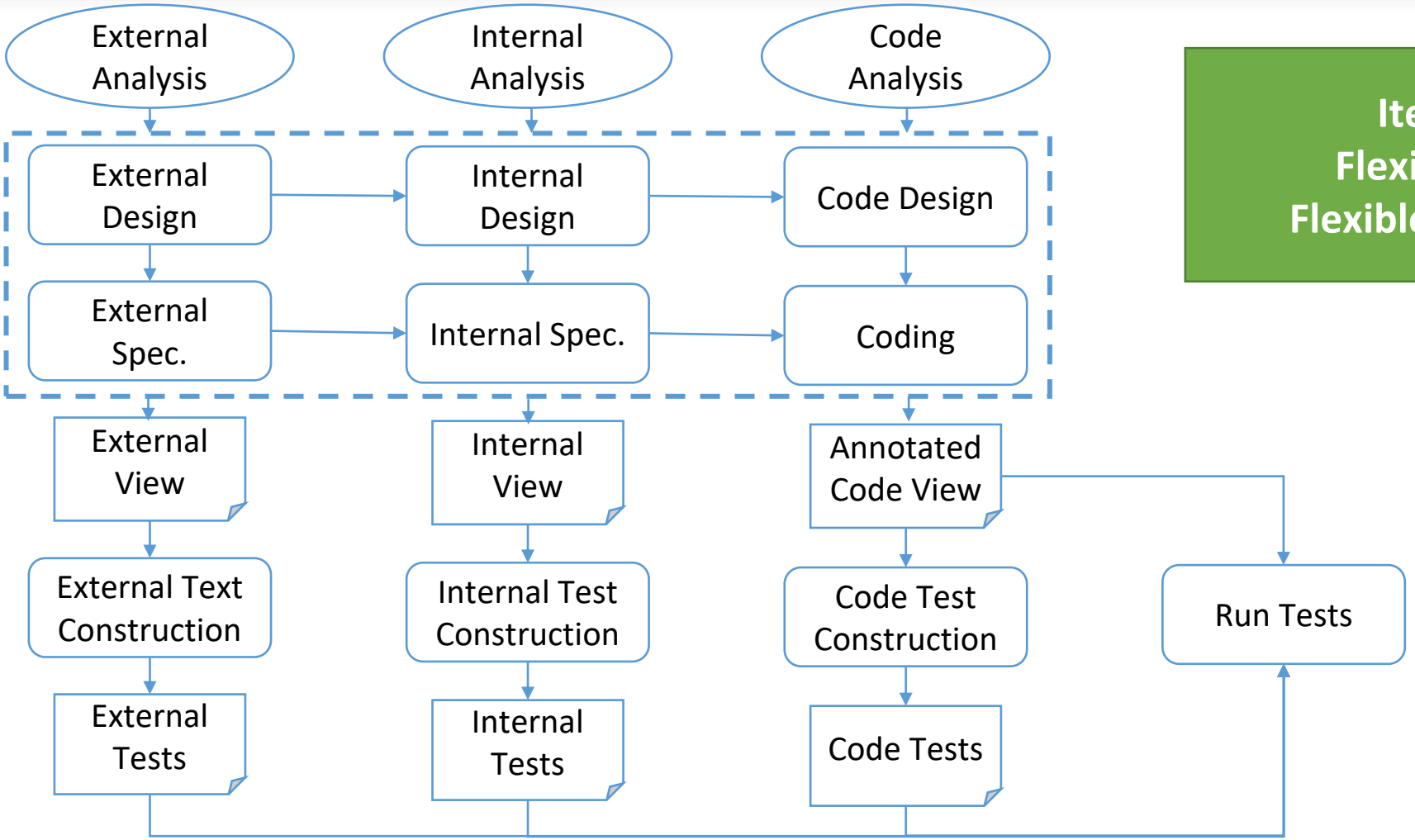
Programming is a complex activity

- Often focus on conceptual knowledge
  - Syntax: variable declaration, if-then-else, …
  - Concepts: control structure, polymorphism, dynamic binding, inheritance, …

- For complex activities, also procedural knowledge is required (J. Merrienboer and P. Kirschner, 2017)

- Procedural knowledge, e.g.,
  - What activities are needed to solve this problem?
  - What decisions need to be made?
  - In which order?
  - Which conceptual knowledge is needed?
  - …

Procedural Guidance



Iterative and incremental
Flexible in development order
Flexible in level: formal or informal

Procedural Guidance - Experience

- Students don’t like a first-think-then-act approach
- BUT: Students appreciate the contracts (including the specifications)

- Quality (correctness) increases significantly:
  - Number of bugs decreases (~70 % test correctly)
  - Number of bugs found by testing increases (~80% bugs detected)

Procedure Fully Worked Out

Specification based OO development:
Procedural Guidance

OUNL-CS Technical Report, No 6, 2022

Harrie Passier & Lex Bijlsma & Ruurd Kuiper
With the cooperation of:
Greg Alpar
Niels Doorn
Stijn de Gouw
Cornelis Huizing
Harold Pootjes
Stefano Schivo

Co-funded by the Erasmus+ Programme of the European Union

3. Feedback Toolkit

- Goals
  - Support students working on assignments
  - Feedback should be: Timely, Individual
  - Support teachers

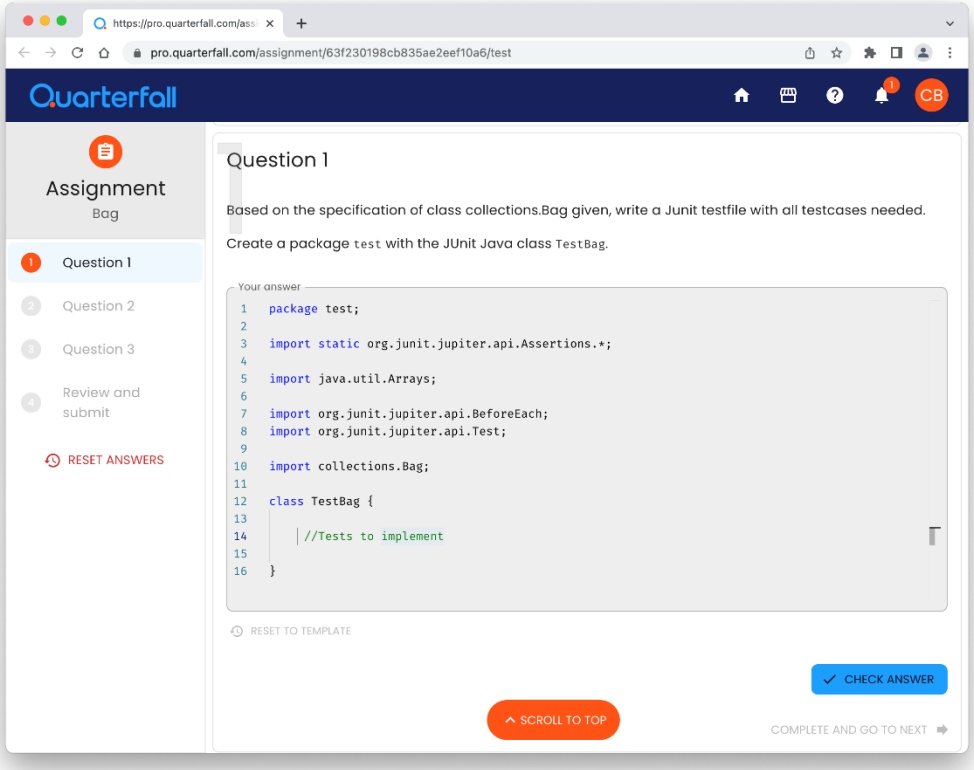
- Automatically generate feedback
  - Available 24/7 with quick response times
  - Easily scalable
  - Should be easy and flexible toconfigure
  - Always monitor quality criteria
    – even if they are not the current learning objective

Feedback Toolkit

- Feedback tools
  - Java
    - Syntax
    - Style
    - Solution approach
    - Test coverage
    - …
  - Python
    - Syntax
    - Style
  - Mermaid diagrams (design)
- Available as plugins for Quarterfall platform
- Also possible to use stand-alone
- Open-source

Simple, constructive, configurable feedback messages.

Unit tests for solution already built-in in Quarterfall.



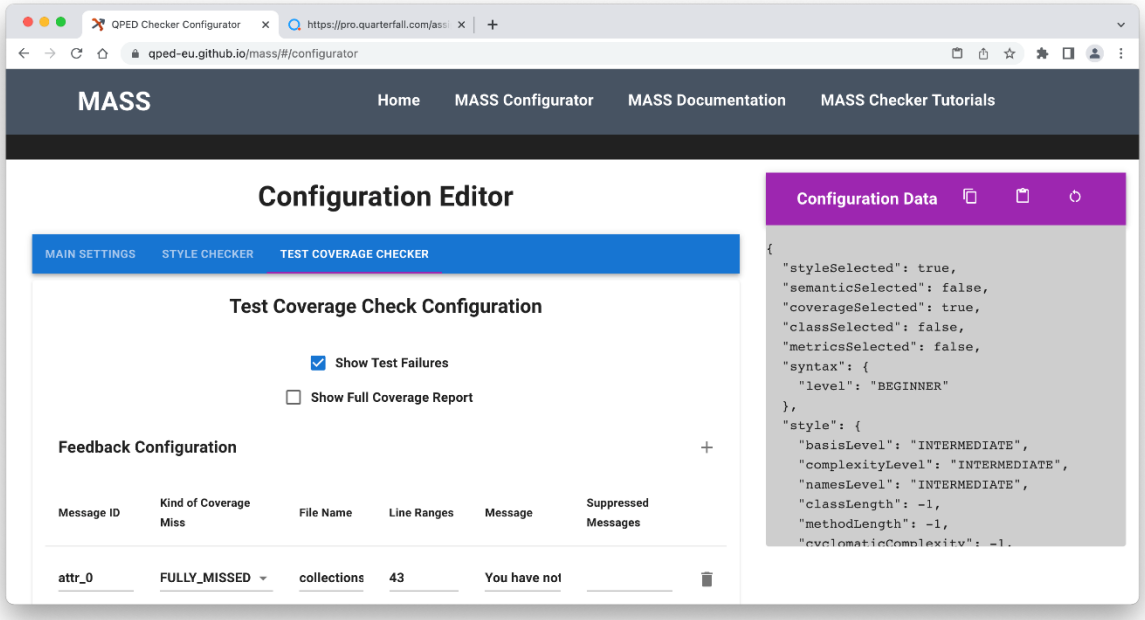
Students view

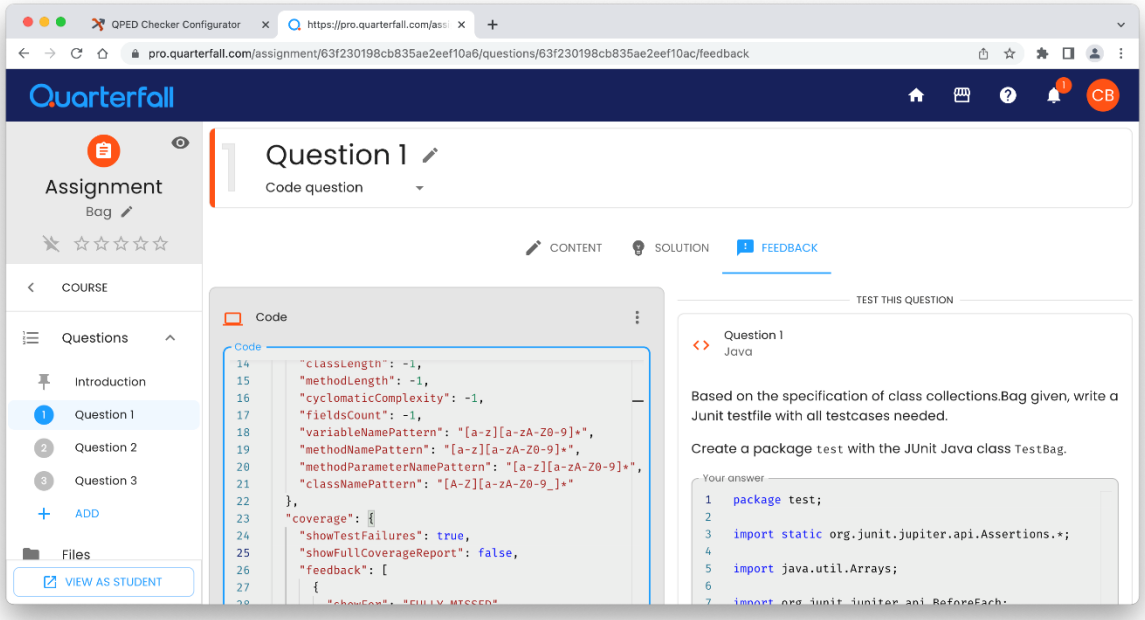
Individual Feedback Messages

- Feedback should be configurable, depending on
  - Level of student
  - Progress in course
  - Teaching style
  - Etc.

- Translate messages for international students

Configuring Feedback Messages





Teachers view

4. Pilot Study

- Choose one course per partner
- Collect data from the courses:
  - without applying the QPED approach (aka Baseline)
  - after applying the QPED approach (aka Validation)
- Compare both collections of data

Can we measure quality-awareness?

Instruments for Assessing Quality Awareness

- **Rubric**: to assess assignments by using a common instrument which helps us to compare results

- **Questionnaire**: to gather students' perceptions about how they apply quality issues while programming.

- **Diagnostic test**: to assess students' awareness in an objective way.

Rubric

4-point Likert scale

QPED O4 - Rubric

File | /Users/bockisch/QPED-Workspace/rubrics-creation-tool/Rubric.html

Feature	Negative Examples	1 - Fully Failed	2 - Partly Failed	3 - Partly Satisfied	4 - Fully Satisfied	Positive Examples
Readability	☐ Wrong indentation ☐ Broken lines of code ☑ Parentheses wrongly placed ☐ Poor naming ☑ Bad comments	○ 1	○ 2	◉ 3	○ 4	☑ Correct indentation ☑ Considered lines of code ☐ Correctly placed parentheses ☑ Good names ☐ Useful comments
DRY principle	☐ Repeated code ☑ Magic numbers	○ 1	◉ 2	○ 3	○ 4	☐ Helper functions are used ☐ Use of constants
Correctness	☐ Wrong file format ☐ Not compiling/running ☐ Specifications not met	○ 1	○ 2	○ 3	◉ 4	☑ Functions properly ☑ Correct results

Achieved Points

0 | Apply Computed

Calculated Points

2.5 (avg. wght. score: 1

Maximum Points

5

Generate Feedback

Next Student

Export all feedback for chosen task

Choose a format for exporting (default is JSON):
JSON-Format

10+ items related to quality issues

Pro and Con examples

Can generate grade and response based on selection.

Pilot Study: collect rubric data

Questionnaire

Assessment of Basic Self-efficacy at SQ

4. Evaluate the following statements:
(1 = strongly disagree; 2 = disagree; 3 = neither agree nor disagree; 4 = agree; 5 = strongly agree)

Concept	1	2	3	4	5
a. I can create to create correct programs for a given task.	☐	☐	☐	☐	☐
b. I can find errors and correct them easily.	☐	☐	☐	☐	☐
c. I can test my code effectively to check for errors.	☐	☐	☐	☐	☐
d. I can create code that can be easily readable by others.	☐	☐	☐	☐	☐
e. I am capable of reusing previous code for new projects.	☐	☐	☐	☐	☐

Perception of Utility of SQ

5. Evaluate the following statements:
(1 = strongly disagree; 2 = disagree; 3 = neither agree nor disagree; 4 = agree; 5 = strongly agree)

Concept	1	2	3	4	5
a. I understand what software quality means.	☐	☐	☐	☐	☐
b. Using testing enhances the quality of programs.	☐	☐	☐	☐	☐
c. Testing is as important as coding.	☐	☐	☐	☐	☐
d. Readability is as important as the correctness of the program.	☐	☐	☐	☐	☐

23 items

5-point Likert scale

Diagnostic test

We want to create a code that calculates the factorial of a number in the range of 0 and 12, both included.

```
public int factorial (int number) {
  if (number<0 || number>=12){
      System.out.println("Error: number out of range");
      return -1;
  }else  if (number==0){
      return 1;
  }else{
     return number * factorial(number-1);
  }
}
```

Indicate three test cases that we must to create in order to check the correctness of the program

The following code has some semantic style errors.

```
public int specialFunction(String word, int times) {
  int length;
  length = (int) (word.length() * times);

  if(word.length() * times % 2 == 0){
   System.out.println("New length: "+length);
   return true;
  }else if(word.length() * times % 2 != 0){
   System.out.println("New length: "+length);
   return false;
  }
}
```

a) Which are they?
b) Refactor the code:

Resources

- Available from the our web site (https://qped.eu)
  - TILE repository
    - Open-source Git repository
  - Procedural Guidance
    - Fully worked out documentation
  - Feeback tools for Java and Python
    - Tutorials
    - Documentation
    - Open-source Git repository
  - Rubric tool
    - Open-source Git repository
    - Available as Github Page

QPED Project – Thank you

QPED Homepage

https://qped.eu

visit our demo
on the
exhibition area

QPED Newsletter

https://listserv.dfn.de/sympa/subscribe/qped-info