

Transition from passive learner to critical evaluator through peer-testing of programming artefacts

Léon McGregor, Dr Manuel Maarek, Dr Joe Wells, Dr Gudmund Grov (Edinburgh)

Dr Mohammad Hamdan, Dr Hind Zandout, Smitha S Kumar, Talal Shaikh (Dubai)

MACS Computer Science (Heriot-Watt University)

Horizons in STEM Higher Education Conference 2017: Making Connections, Innovating and Sharing Pedagogy – 29-30 June 2017

Plan

- Overview of motivations of the project
- The web platform implemented
- Study evaluating usefulness of platform
- Outcomes of evaluation
- Perspectives going forwards

Situation

- CompSci Courses use **individual** programming tasks for coursework
- **Feedback** on programming artefacts is **not timely**
- Students focus on working to solve a problem, and so don't always **critically analyse** and evaluate their solutions

Solution

- Peer-Testing: A web platform for peer-feedback on programming artefacts
- Students working together means **faster turnaround** of feedback
- Students get a chance to **critically evaluate** work of others
- Web allows for access **across campus**, home
- Platform **integrates testing**

Background

- **Peer-Assessment** has many benefits [1,2]
 - Improves students understanding of task
 - Students learn strengths & weaknesses
 - Improves attitude to learning
- **Peer-Testing** extends this
 - Focus on testing

[1] - Sadler & Good, 2006 – DOI 10.1207/s15326977ea1101_1

[2] - Falchikov, 2005 – ISBN 978-1-134-39575-0

Background

- CAP System for **Peer-Feedback** on reports^[1]
 - Highlights importance of anonymity
 - Peer Marking was positive experience
- Ceilidh System for **Automating Feedback**^[2]
 - Helps students get feedback quicker
 - Lets teachers track progress

[1] - Davies, 2000 – DOI 10.1080/135580000750052955

[2] – Foubister, 1997 – DOI 10.1046/j.1365-2729.1997.00012.x

Overview

- I took this on as an Honours Project
 - Teacher-informed experimental prototype
 - Student-led investigation
 - Feedback from students
- This is a QAA funded project (CS Education)

PEER-TESTING WEB PLATFORM

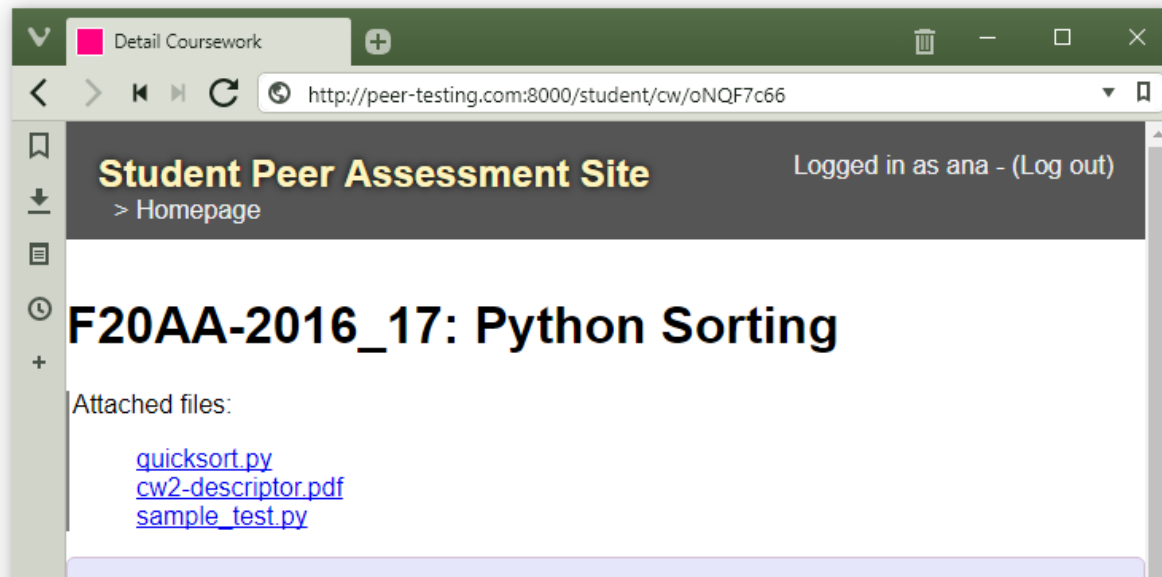
Features

- Offers programming support & **testing**
 - E.g. Python & Python unit testing
- Lets students interact **anonymously**
- **Feedback sharing** between tester, developer

Stage 0

Coursework Setup

- Teacher creates coursework
- Makes available to students
- Students can download relevant files



Stage 1

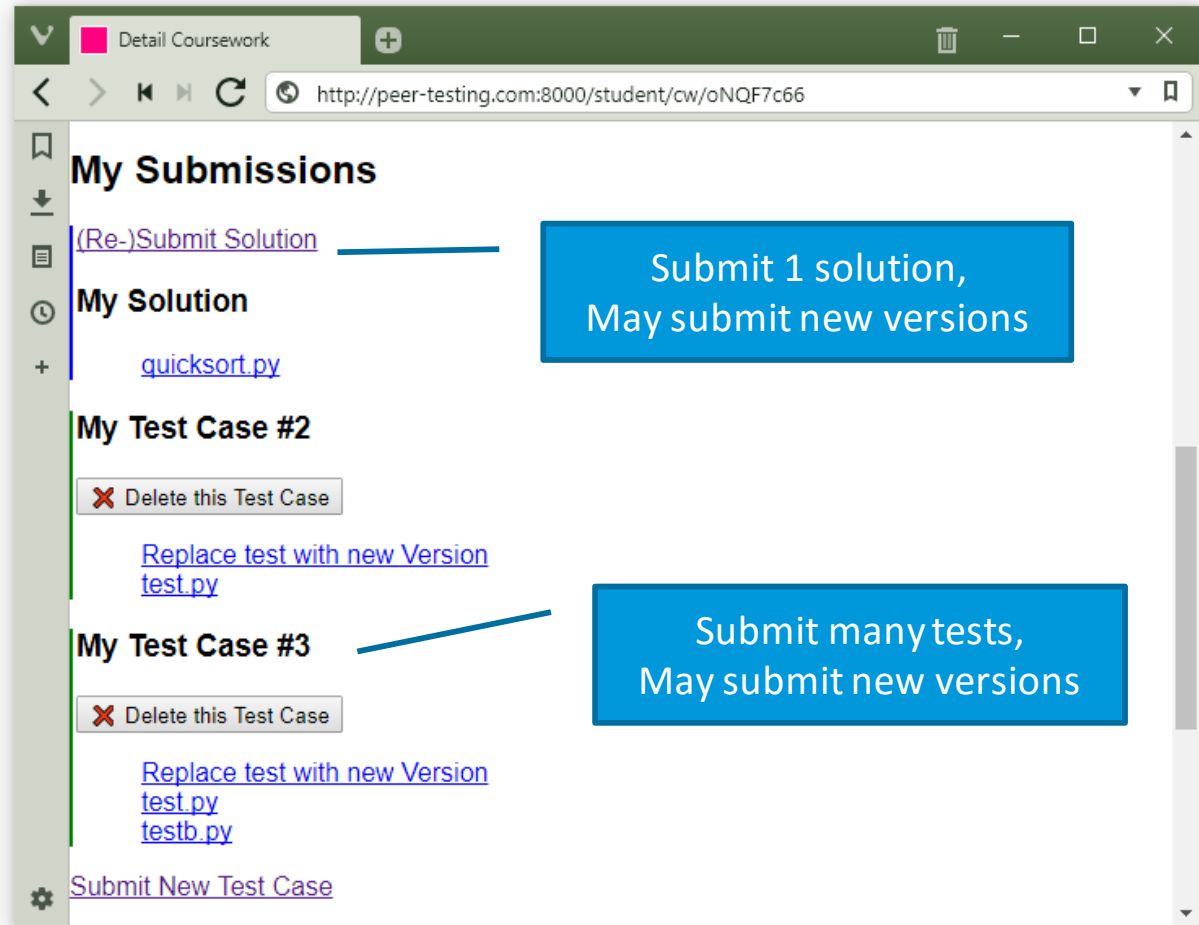
Development & Self-Testing

- Upload solutions & tests
- Define and run tests
- View test results

Stage 1

Development & Self-Testing

- Upload solutions & tests
- Define and run tests
- View test results



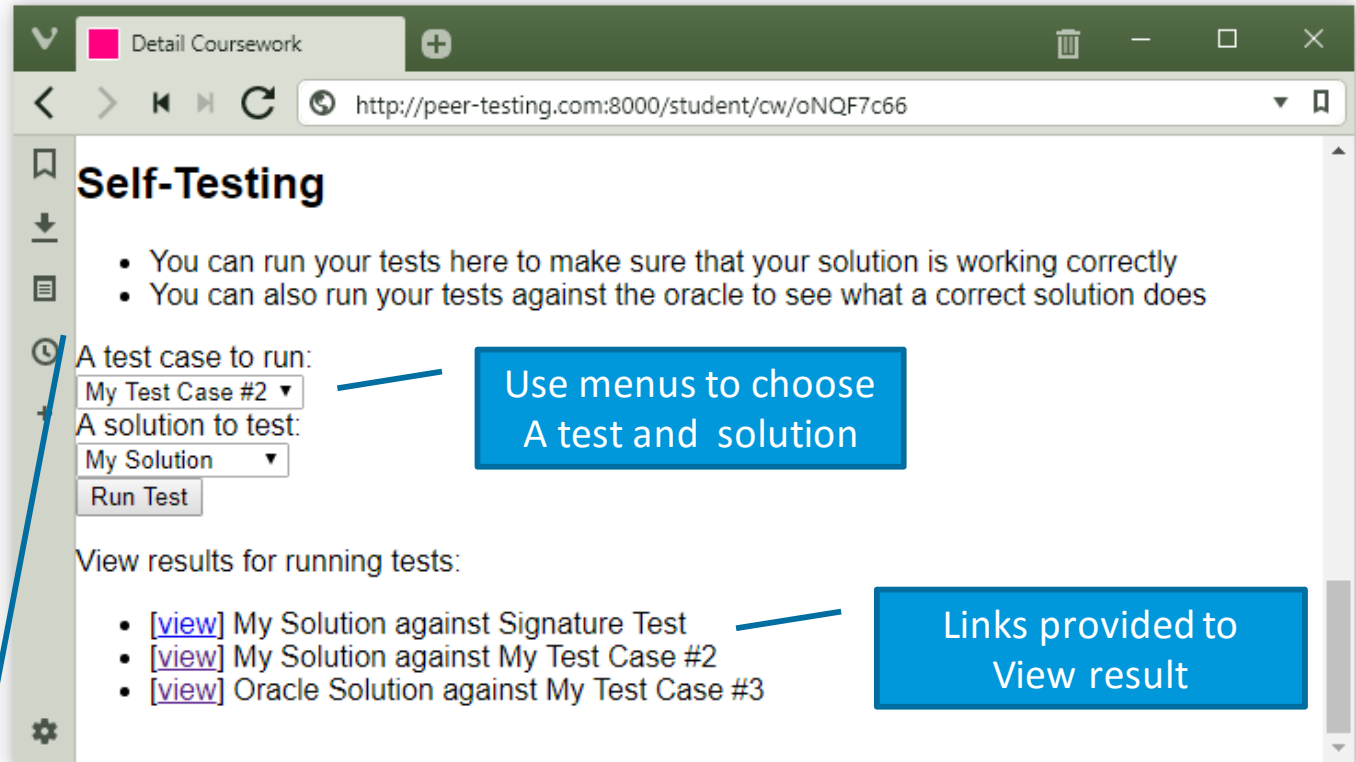
The screenshot shows a web browser window with the URL <http://peer-testing.com:8000/student/cw/oNQF7c66>. The page is titled "Detail Coursework" and contains the following sections:

- My Submissions**: Includes a link [\(Re-\)Submit Solution](#). A blue callout box points to this link with the text: "Submit 1 solution, May submit new versions".
- My Solution**: Includes a link [quicksort.py](#).
- My Test Case #2**: Includes a button [Delete this Test Case](#) and a link [Replace test with new Version test.py](#).
- My Test Case #3**: Includes a button [Delete this Test Case](#) and links [Replace test with new Version test.py](#) and [testb.py](#). A blue callout box points to this section with the text: "Submit many tests, May submit new versions".
- At the bottom, there is a link [Submit New Test Case](#).

Stage 1

Development & Self-Testing

- Upload solutions & tests
- Define and run tests
- View test results



Self-Testing

- You can run your tests here to make sure that your solution is working correctly
- You can also run your tests against the oracle to see what a correct solution does

A test case to run:
My Test Case #2 ▼

A solution to test:
My Solution ▼

Run Test

View results for running tests:

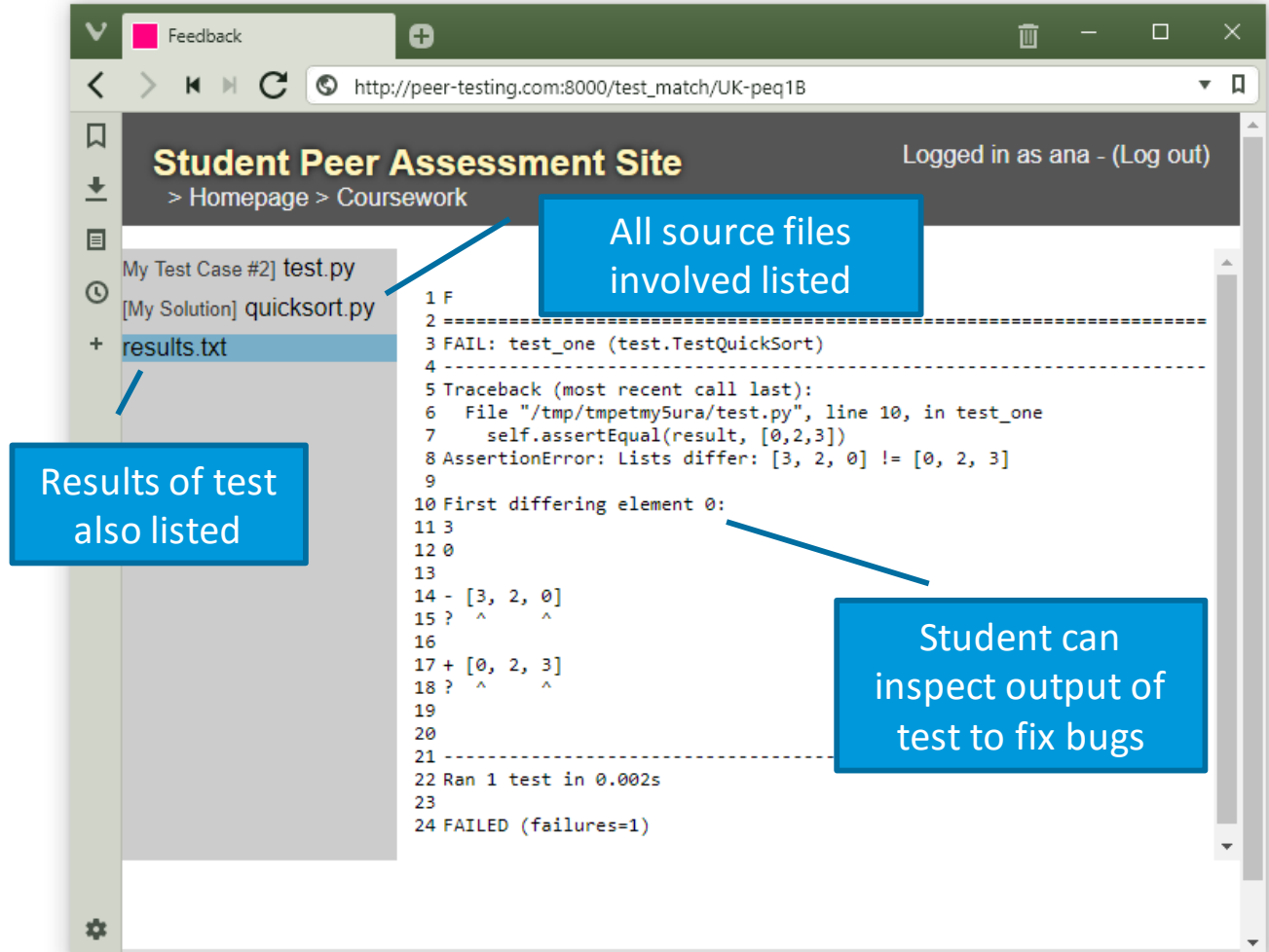
- [\[view\]](#) My Solution against Signature Test
- [\[view\]](#) My Solution against My Test Case #2
- [\[view\]](#) Oracle Solution against My Test Case #3

Pre-Built oracle, and
signature test available

Stage 1

Development & Self-Testing

- Upload solutions & tests
- Define and run tests
- View test results



The screenshot shows a web browser window with the URL `http://peer-testing.com:8000/test_match/UK-peq1B`. The page title is "Student Peer Assessment Site" and it shows the user is logged in as "ana" with a "(Log out)" link. The breadcrumb navigation is "> Homepage > Coursework".

On the left sidebar, there is a list of files: "My Test Case #2] test.py", "[My Solution] quicksort.py", and "results.txt". The "results.txt" file is selected and highlighted in blue.

Three blue callout boxes provide additional context:

- A box labeled "All source files involved listed" points to the file list on the left.
- A box labeled "Results of test also listed" points to the "results.txt" file in the list.
- A box labeled "Student can inspect output of test to fix bugs" points to the test output text on the right.

The main content area displays the test output for the selected file. It shows a failure (F) and a traceback. The output is as follows:

```
1 F
2 -----
3 FAIL: test_one (test.TestQuickSort)
4 -----
5 Traceback (most recent call last):
6   File "/tmp/tmpetmySura/test.py", line 10, in test_one
7     self.assertEqual(result, [0,2,3])
8   AssertionError: Lists differ: [3, 2, 0] != [0, 2, 3]
9
10  First differing element 0:
11  3
12  0
13
14  - [3, 2, 0]
15  ? ^    ^
16
17  + [0, 2, 3]
18  ? ^    ^
19
20
21 -----
22 Ran 1 test in 0.002s
23
24 FAILED (failures=1)
```

Stage 2

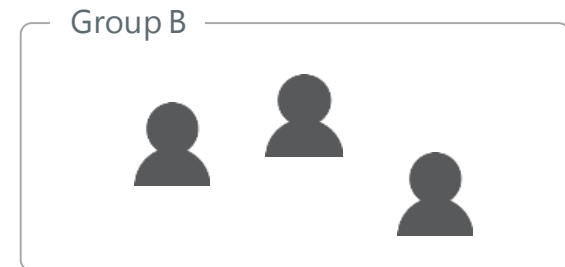
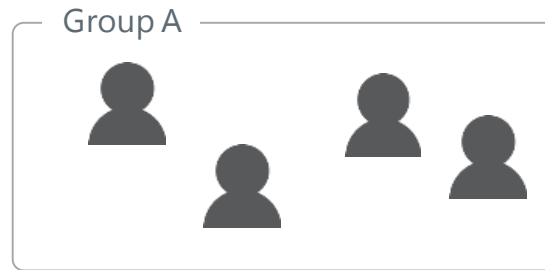
Peer-Testing & Feedback

- Group peers
- Run tests
- Test results & feedback

Stage 2

Peer-Testing & Feedback

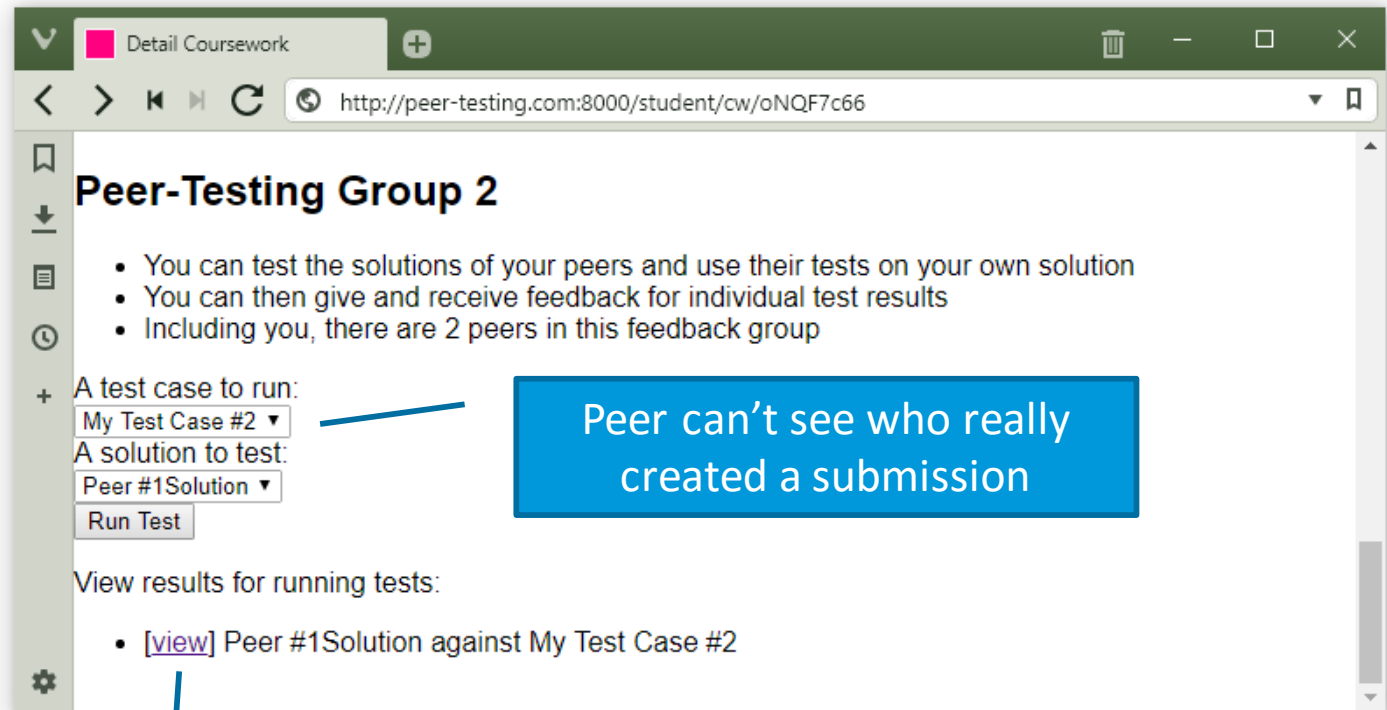
- **Group peers**
- Run tests
- Test results & feedback



Stage 2

Peer-Testing & Feedback

- Group peers
- Run tests
- Test results & feedback



The screenshot shows a web browser window titled "Detail Coursework" with the URL <http://peer-testing.com:8000/student/cw/oNQF7c66>. The page is titled "Peer-Testing Group 2" and contains the following text:

- You can test the solutions of your peers and use their tests on your own solution
- You can then give and receive feedback for individual test results
- Including you, there are 2 peers in this feedback group

Below the list, there are two dropdown menus: "A test case to run:" with "My Test Case #2" selected, and "A solution to test:" with "Peer #1Solution" selected. A "Run Test" button is located below these menus. Below the button, the text "View results for running tests:" is followed by a list item: "• [\[view\]](#) Peer #1Solution against My Test Case #2".

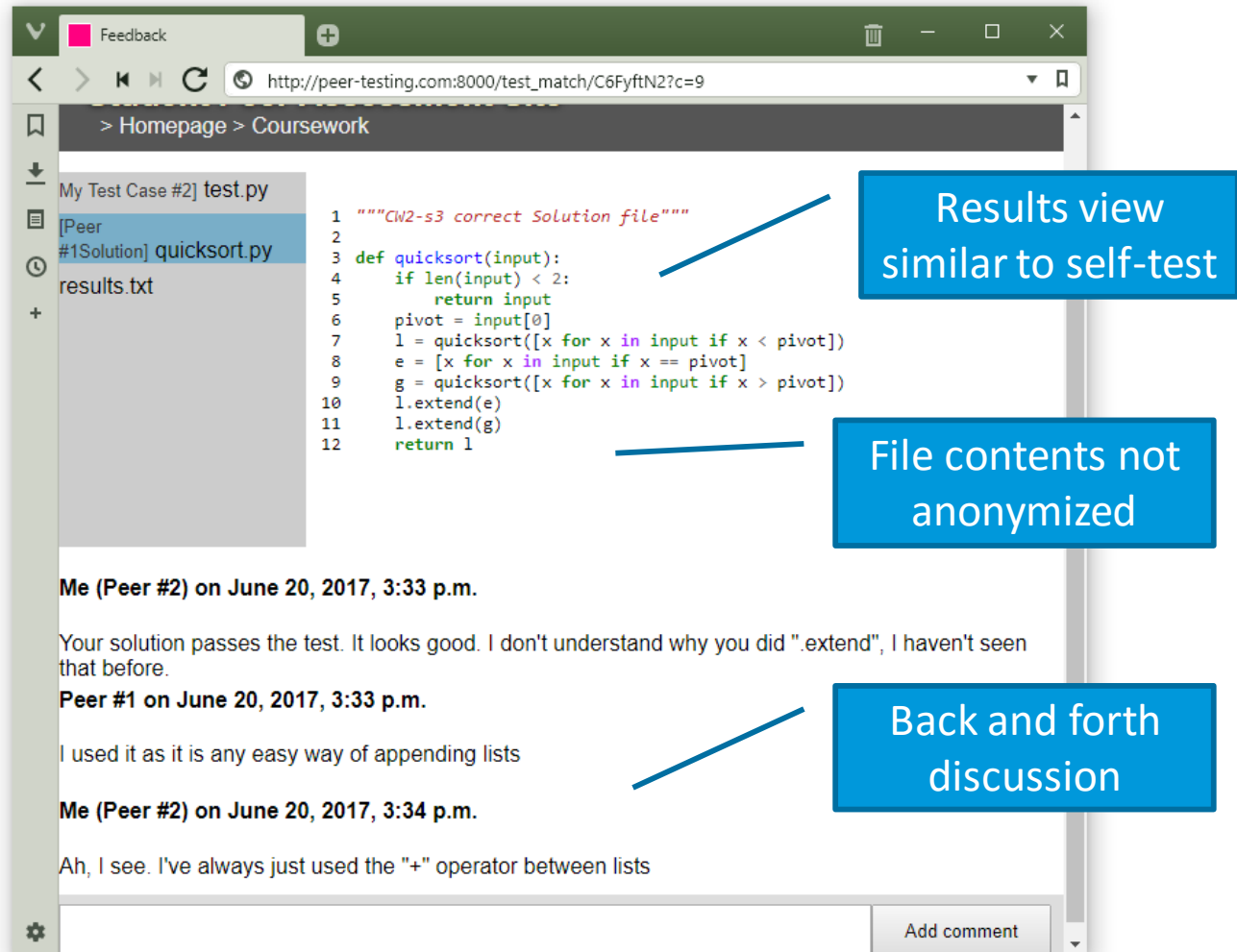
Peer can't see who really
created a submission

Own solution listed when
feedback ready

Stage 2

Peer-Testing & Feedback

- Group peers
- Run tests
- Test results & feedback



The screenshot shows a web browser window titled "Feedback" with the URL `http://peer-testing.com:8000/test_match/C6FyftN2?c=9`. The page displays a file explorer on the left with files: "My Test Case #2] test.py", "[Peer #1Solution] quicksort.py", and "results.txt". The main area shows the content of "quicksort.py", which is a Python function for quicksort. Below the code, there is a discussion thread. The first message is from "Me (Peer #2)" dated June 20, 2017, at 3:33 p.m., stating "Your solution passes the test. It looks good. I don't understand why you did '.extend', I haven't seen that before." The second message is from "Peer #1" dated June 20, 2017, at 3:33 p.m., replying "I used it as it is any easy way of appending lists". The third message is from "Me (Peer #2)" dated June 20, 2017, at 3:34 p.m., replying "Ah, I see. I've always just used the '+' operator between lists". At the bottom right, there is an "Add comment" button.

```
1 """CW2-s3 correct Solution file"""
2
3 def quicksort(input):
4     if len(input) < 2:
5         return input
6     pivot = input[0]
7     l = quicksort([x for x in input if x < pivot])
8     e = [x for x in input if x == pivot]
9     g = quicksort([x for x in input if x > pivot])
10    l.extend(e)
11    l.extend(g)
12    return l
```

Results view similar to self-test

File contents not anonymized

Back and forth discussion

EVALUATION & OUTCOMES

Evaluation Methodology

- Initial **prototype** of web platform
- Evaluated **interactions**
- 17 year 2/3 students
- Across campuses (Edinburgh & Dubai)
- Small python exercises
 - Quicksort, binary search tree
 - Tasks completed using email or web platform

Evaluation Methodology



Online post-experiment cross-campus discussion

Opinions on Peer-Testing

- Giving & receiving feedback helps learning
- Helps understand code testing & review
- Peer-Testing exercises should be part of university courses
- Exercises using a web platform could be part of university courses

Improvements to platform

- Participants in Study requested improvement to
 - Back-And-Forth Feedback
 - Managing Uploaded Files
- Managing Investigation revealed need for
 - Peer Groups

Positive Outcomes

- Provides experience with testing
- Web Platform encourages test-driven development
- Cross-Campus Activity
- Shares Information that might not be common teaching in both campuses
- “Once you realise you’re working cross-campus, the learning becomes more interesting”

PERSPECTIVES GOING FORWARDS

Integration

- VLE (Virtual Learning Environment) Integration
 - Built-in file uploading support
- O365 Single Sign-on Login
- Peer-Feedback Platforms
 - Aropä, WebPA
 - Built-in ways to manage peer activity

Deployment

- Extend support to Java & Junit
- Ensure this works with an existing course
 - Trial with unmarked lab exercises
 - Then full coursework exercise

Public Release

- Eventually release this web platform to allow others to use it
- Contact M.Maarek@hw.ac.uk if interested

ANY QUESTIONS

More info at tiny.cc/peer-testing

