Andy

An educational tool to assess test code

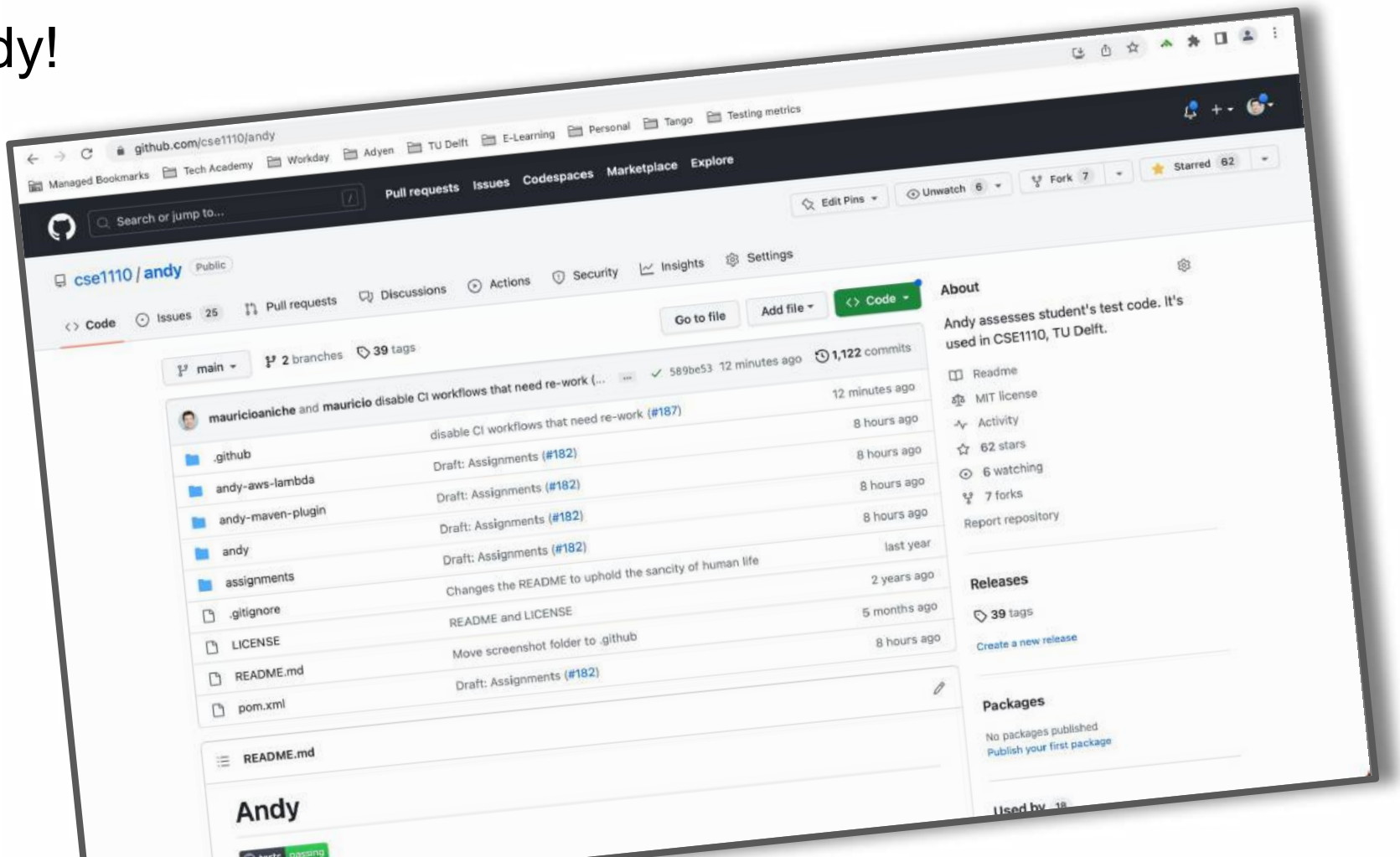
https://github.com/cse1110/andy

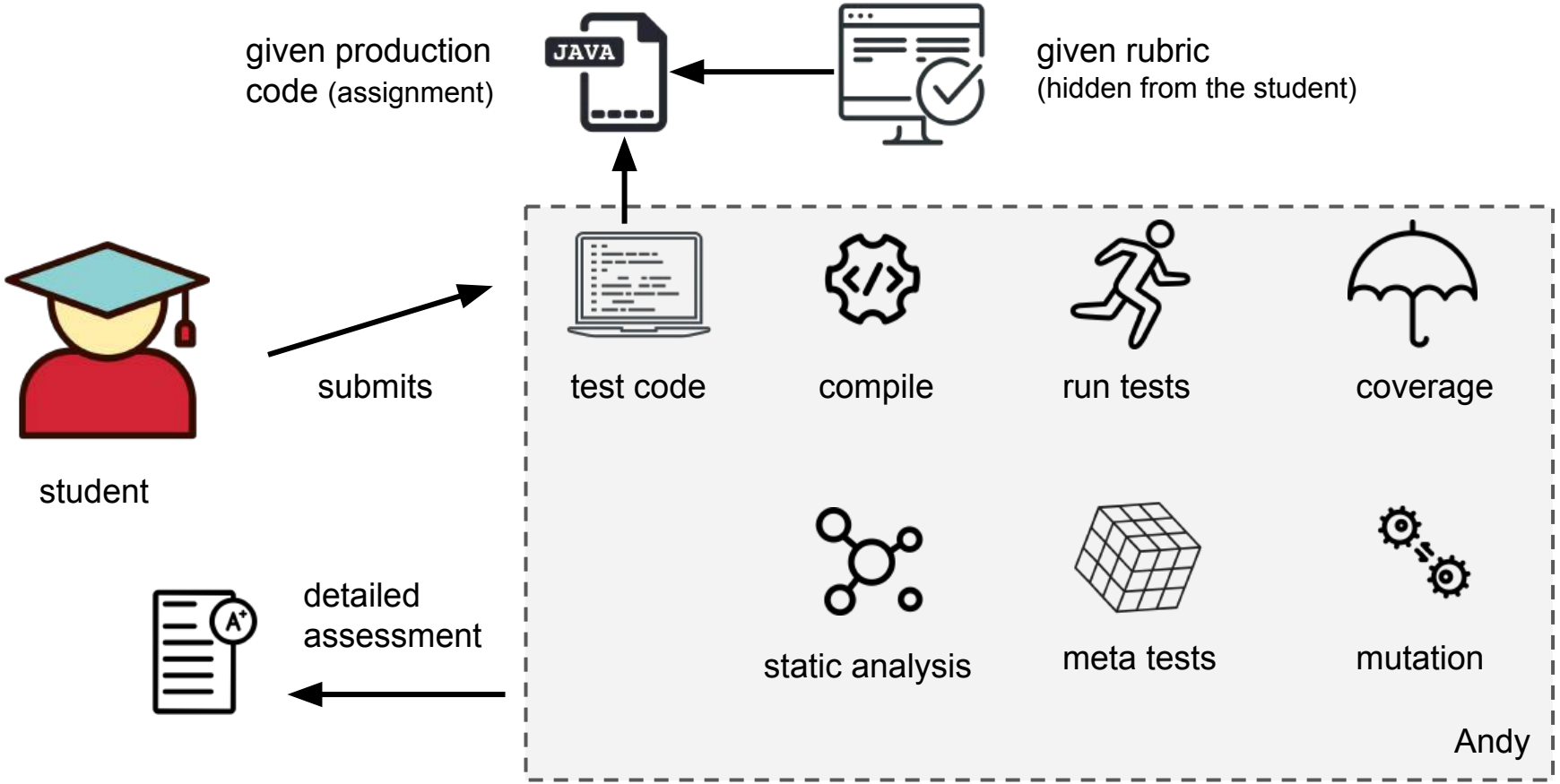
TU Delft

Teaching software testing is fundamental!

- It is an important skill to learn
- Not a lot of universities teach it.
  - In most of them, testing is just a lecture in a SE course
  - Delft has a dedicated testing course!
- Lack of tools to support testing education!
  - How to make sure the student wrote the "right tests"?

Andy!


github.com/cse1110/andy
Managed Bookmarks
Tech Academy
Workday
Adyen
TU Delft
E-Learning
Personal
Tango
Testing metrics
Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
Edit Pins
Unwatch 6
Fork 7
Starred 62
cse1110 / andy
Public
Code
Issues 25
Pull requests
Discussions
Actions
Security
Insights
Settings
Go to file
Add file
Code
main
2 branches
39 tags
mauricioaniche and mauricio disable CI workflows that need re-work (...
589be53 12 minutes ago
1,122 commits
.github
disable CI workflows that need re-work (#187)
12 minutes ago
andy-aws-lambda
Draft: Assignments (#182)
8 hours ago
andy-maven-plugin
Draft: Assignments (#182)
8 hours ago
andy
Draft: Assignments (#182)
8 hours ago
assignments
Draft: Assignments (#182)
last year
.gitignore
Changes the README to uphold the sancity of human life
2 years ago
LICENSE
README and LICENSE
5 months ago
README.md
Move screenshot folder to .github
8 hours ago
pom.xml
Draft: Assignments (#182)
README.md
Andy
About
Andy assesses student's test code. It's used in CSE1110, TU Delft.
Readme
MIT license
Activity
62 stars
6 watching
7 forks
Report repository
Releases
39 tags
Create a new release
Packages
No packages published
Publish your first package
Used by 18

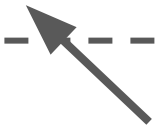

given production code (assignment)
JAVA
given rubric
(hidden from the student)
student
submits
test code
compile
run tests
coverage
static analysis
meta tests
mutation
detailed assessment
Andy

Some quick explanation of the assignment

The `concatenate` method concatenates a sequence of arrays. The return array consists of the entries of the input arrays concatenated in the order they appear in the argument list. Null arrays cause NullPointerExceptions; zero length arrays are allowed (contributing nothing to the output array).

Write tests for this program.

(Method from Apache Commons Math: https://github.com/apache/commons-math/blob/acfc27083412f57a71a497e500070a08a956300d/commons-math-legacy-core/src/main/java/org/apache/commons/math4/legacy/core/MathArrays.java#L1075)



Reference to where we got inspiration from

```java
class MathArrays {

        /** <code>MathArrays</code> should not normally be instantiated. */
        private MathArrays() {
        }

        /**
         * Concatenates a sequence of arrays. The return array consists of the entries
         * of the input arrays concatenated in the order they appear in the argument
         * list. Null arrays cause NullPointerExceptions; zero length arrays are allowed
         * (contributing nothing to the output array).
         *
         * @param x
         *            list of double[] arrays to concatenate
         * @return a new array consisting of the entries of the argument arrays
         * @throws NullPointerException
         *             if any of the arrays are null
         * @since 3.6
         */
        public static double[] concatenate(double[]... x) {
                int combinedLength = 0;
                for (double[] a : x) {
                        combinedLength += a.length;
                }
                int offset = 0;
                int curLength = 0;
                final double[] combined = new double[combinedLength];
                for (int i = 0; i < x.length; i++) {
                        curLength = x[i].length;
                        System.arraycopy(x[i], 0, combined, offset, curLength);
                        offset += curLength;
                }
                return combined;
        }
}
```

Javadoc that may complement the requirement

Plain old Java code

```java
public class Configuration extends RunConfiguration {

    @Override
    public Map<String, Float> weights() {
        return new HashMap<>() {{
            put("coverage", 0.15f); // was 1.0
            put("mutation", 0.45f); // was 0.0
            put("meta", 0.4f); // was 0.0
            put("codechecks", 0.0f);
        }};
    }

    @Override
    public List<String> classesUnderTest() {
        return List.of("delft.MathArrays");
    }

    @Override
    public List<MetaTest> metaTests() {
        return List.of(
                MetaTest.insertAt("does not work with no arrays provided", 23,
                        "if (x.length == 0) throw new RuntimeException(\"killed the mutant\");"),
                MetaTest.insertAt("does not work with a single array", 23,
                        "if (x.length == 1) throw new RuntimeException(\"killed the mutant\");"),
                MetaTest.withStringReplacement("does not work with more than 2 arrays",
                        "i < x.length;",
                        "i < x.length && i < 2;"),
                MetaTest.insertAt("does not work with an empty array provided", 32,
                        "if (curLength == 0) throw new RuntimeException(\"killed the mutant\");"),
                MetaTest.withStringReplacement("does not work with arrays of different sizes",
                        "curLength = x[1].length;",
                        "curLength = x[0].length;")
        );
    }

}
```

How much each part contributes to the final assessment

The name of the class under test

Meta tests

```java
class ReleaseEditions {

	private final BookService bookService;

	private final EmailService emailService;

	public ReleaseEditions(BookService bookService, EmailService emailService) {
		this.bookService = bookService;
		this.emailService = emailService;
	}

	/**
	 * Creates a copy of the book, now with the new edition in the title.
	 *
	 * @param author
	 *            name of the author
	 * @param keyword
	 *            keyword to search
	 * @param edition
	 *            number of the new edition
	 */
	public void releaseNewEdition(String author, String keyword, int edition) {
		List<String> allBooks = bookService.retrieveBooks(author);
		for (String bookTitle : allBooks) {
			if (bookTitle.contains(keyword)) {
				bookService.addBook(author, bookTitle + " - edition " + edition);
			}
		}
	}
}
```

Requires mocking!

```java
@Override
public CheckScript checkScript() {
    return new CheckScript(List.of(
            new SingleCheck("EmailService should be mocked",
                    new MockClass("EmailService")),
            new SingleCheck("BookService should be mocked",
                    new MockClass("BookService")),
            new SingleCheck("BusinessImpl should not be mocked", true,
                    new MockClass("BusinessImpl")),
            new SingleCheck("Spies should not be used", true,
                    new MockitoSpy()),
            new SingleCheck(2, "retrieveBooks should not be verified", true,
                    new MockitoVerify("retrieveBooks", MockitoVerify.MethodType.TEST,
                            Comparison.GTE, 1)),
            new SingleCheck("addBook should be verified",
                    new MockitoVerify("addBook", MockitoVerify.MethodType.TEST,
                            Comparison.GTE, 1, true))
    ));
}

@Override
public List<MetaTest> metaTests() {
    return List.of(
            MetaTest.withLineReplacement("change condition", 45, 50,
                    """
                            List<String> allBooks = bookService.retrieveBooks(author);
                            for (String bookTitle : allBooks) {
                                    if (!bookTitle.contains(keyword)) {
                                            bookService.addBook(author, bookTitle + " - edition " + edition);
                                    }
                            }
                    """
            )
    );
}
```

Static analysis to identify whether mocks were used properly!

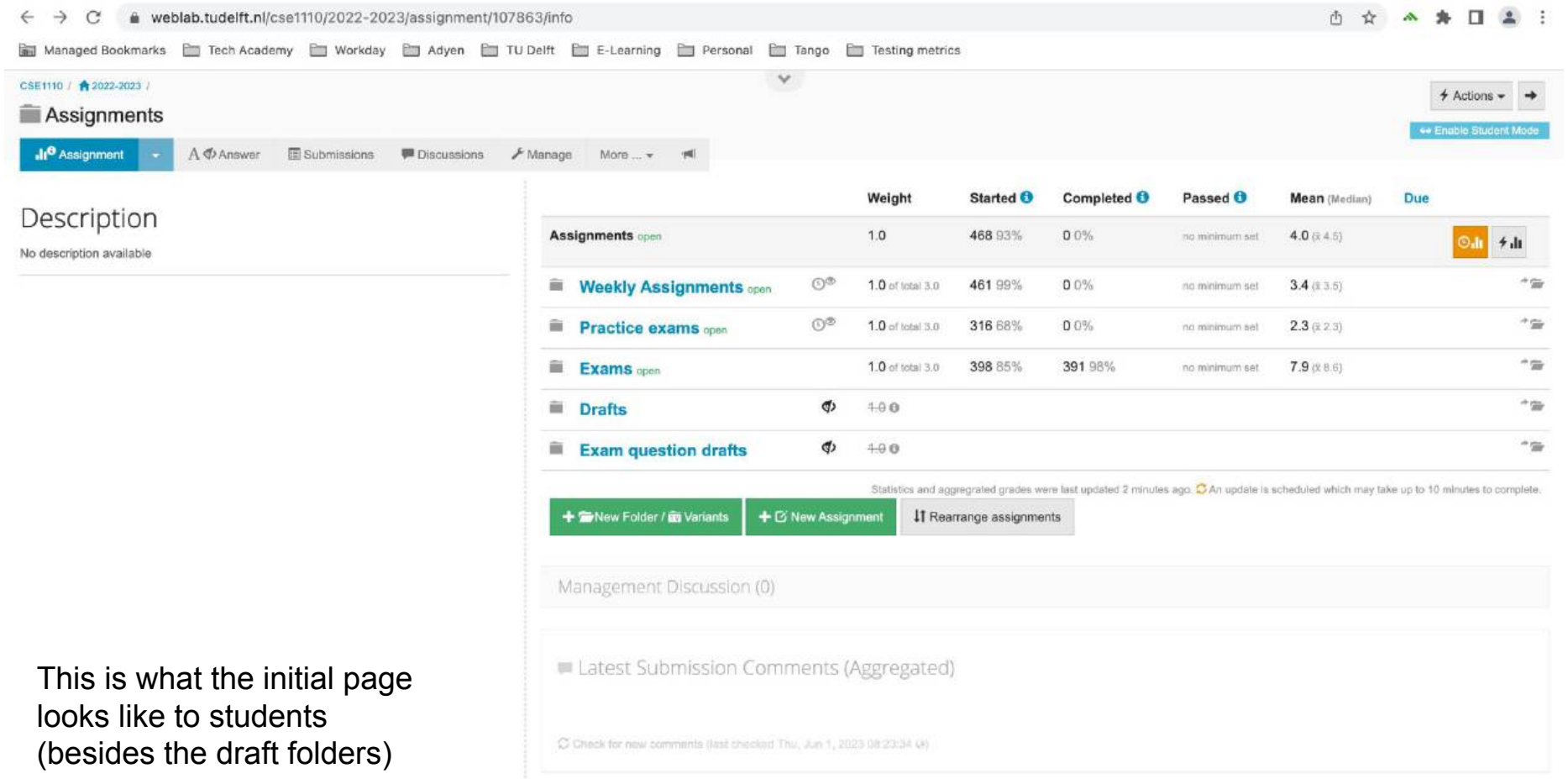
Code checks

The tool contains different checks for JUnit, Mockito, and JQWik tests:

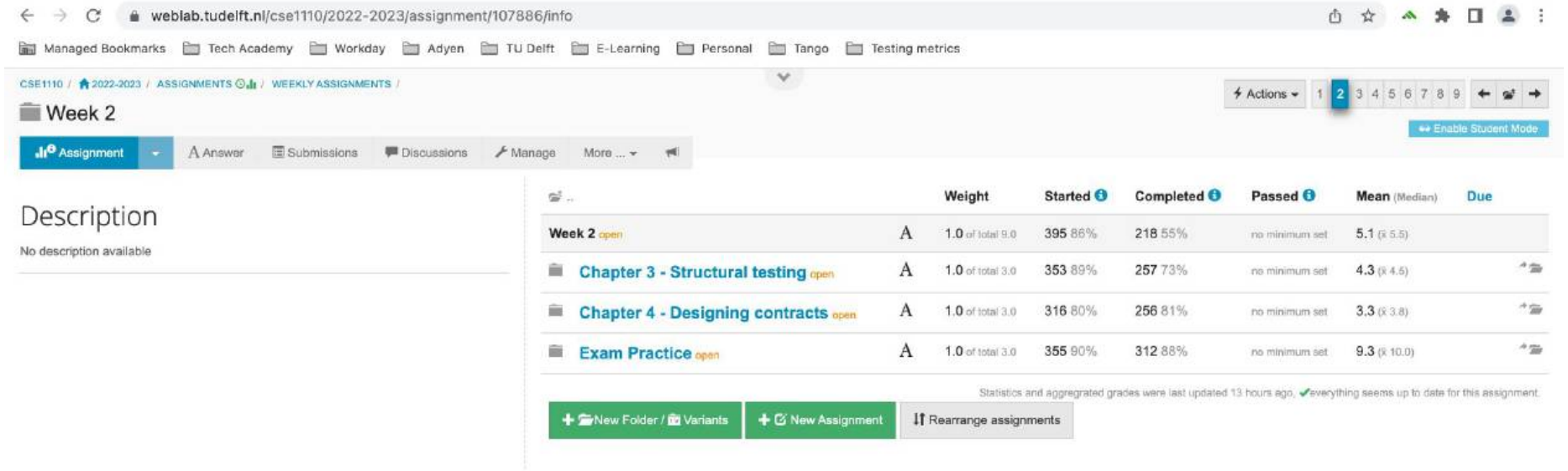
- Test methods:
  - `NumberOfTests` : checks whether the test suite has a minimum number of tests.
  - `TestMethodsHaveAssertions` : checks whether all test methods have assertions.
  - `LoopInTestMethods` : checks whether there is a loop in a test method.
  - `UseOfStringLiterals` : checks whether there is a string literal in a test method.
  - `MethodCalledInTestMethod` : checks whether a method was invoked in a test method.
- Mockito:
  - `MockClass` : Checks whether a class was mocked in the test suite.
  - `MockitoSpy` : Checks whether spies are used.
  - `MockitoVerify` : Checks whether a specific verify has happened.
  - `MockitoWhen` : Checks whether a specific when() has happened.
- JQWik:
  - `JQWikProperty` : checks whether the test suite has a minimum number of properties.
  - `JQWikProvide` : checks whether the test suite has a minimum number of provide.
  - `JQWikCombinator` : checks whether a Combinator was used.
  - `JQWikArbitrary` : checks whether a specific Arbitrary is provided by any method in the test suite.
  - `JQWikArbitraries` : checks whether a Arbitraries.x() is used in the test suite.
  - `JQWikProvideAnnotations` : checks whether tests use Provide annotations, e.g., @ForAll, @Positive.

Each of these checks receive different parameters. Check their specific Javadoc for more details.

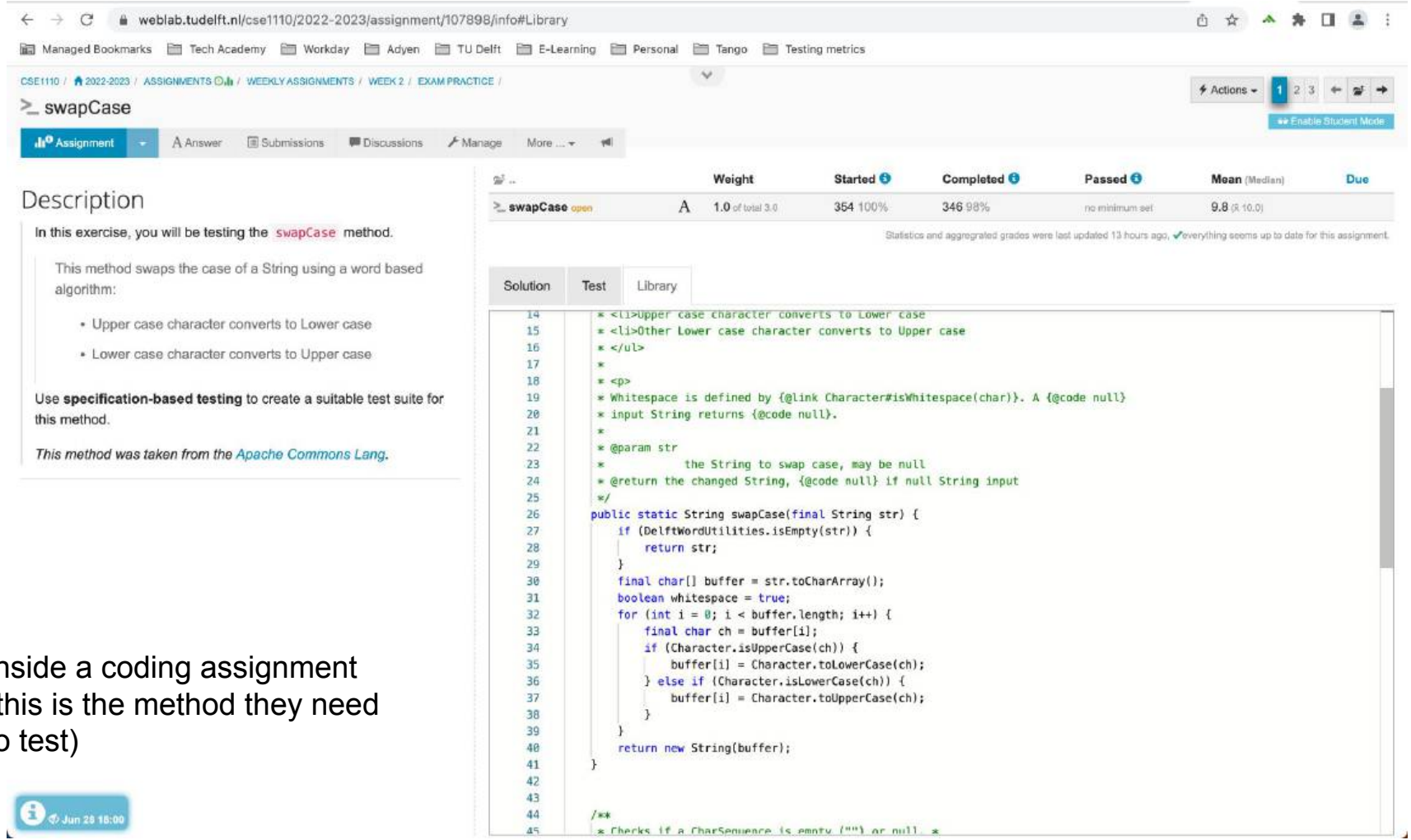
It's demo time!



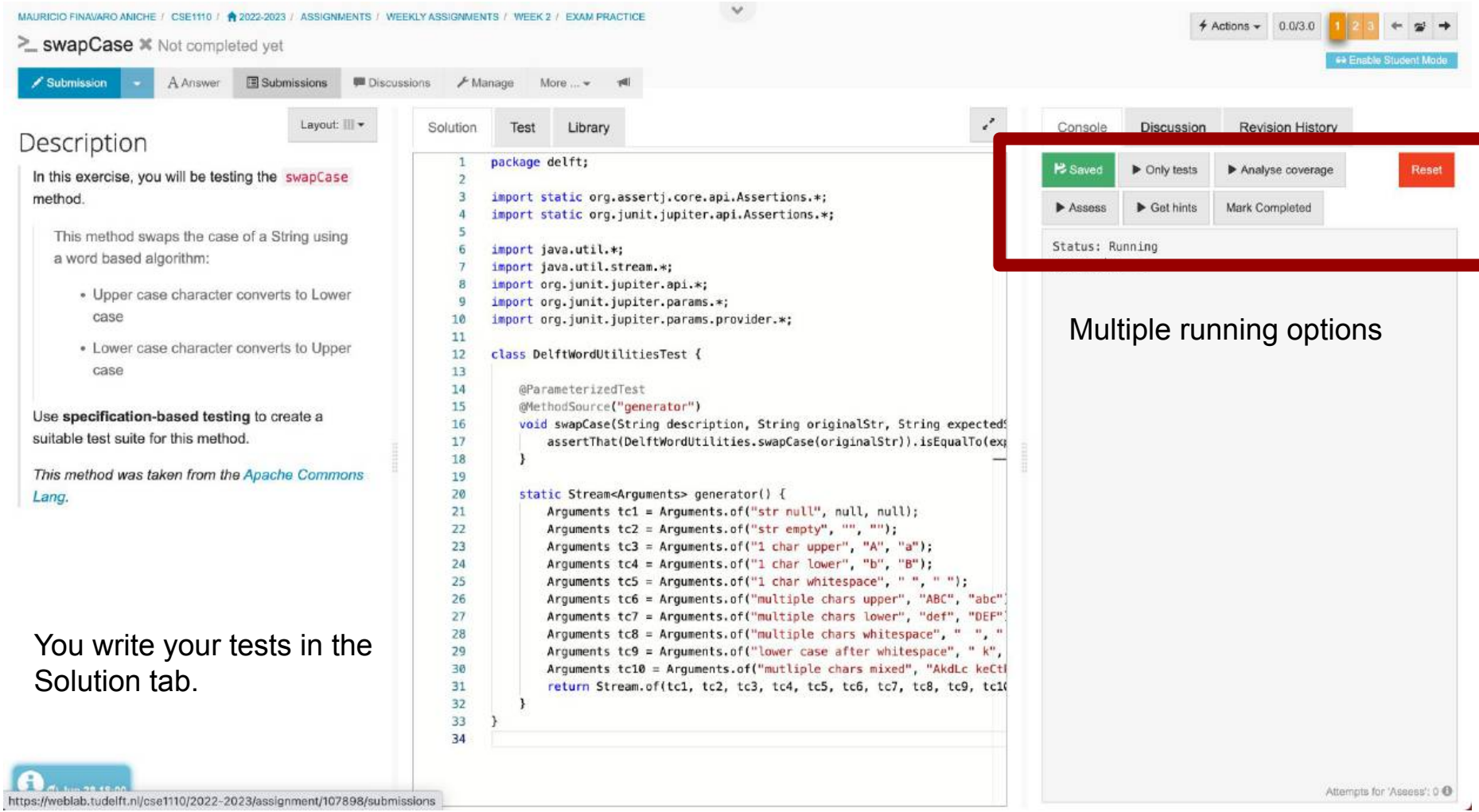
This is what the initial page looks like to students (besides the draft folders)



Exercises are divided by week

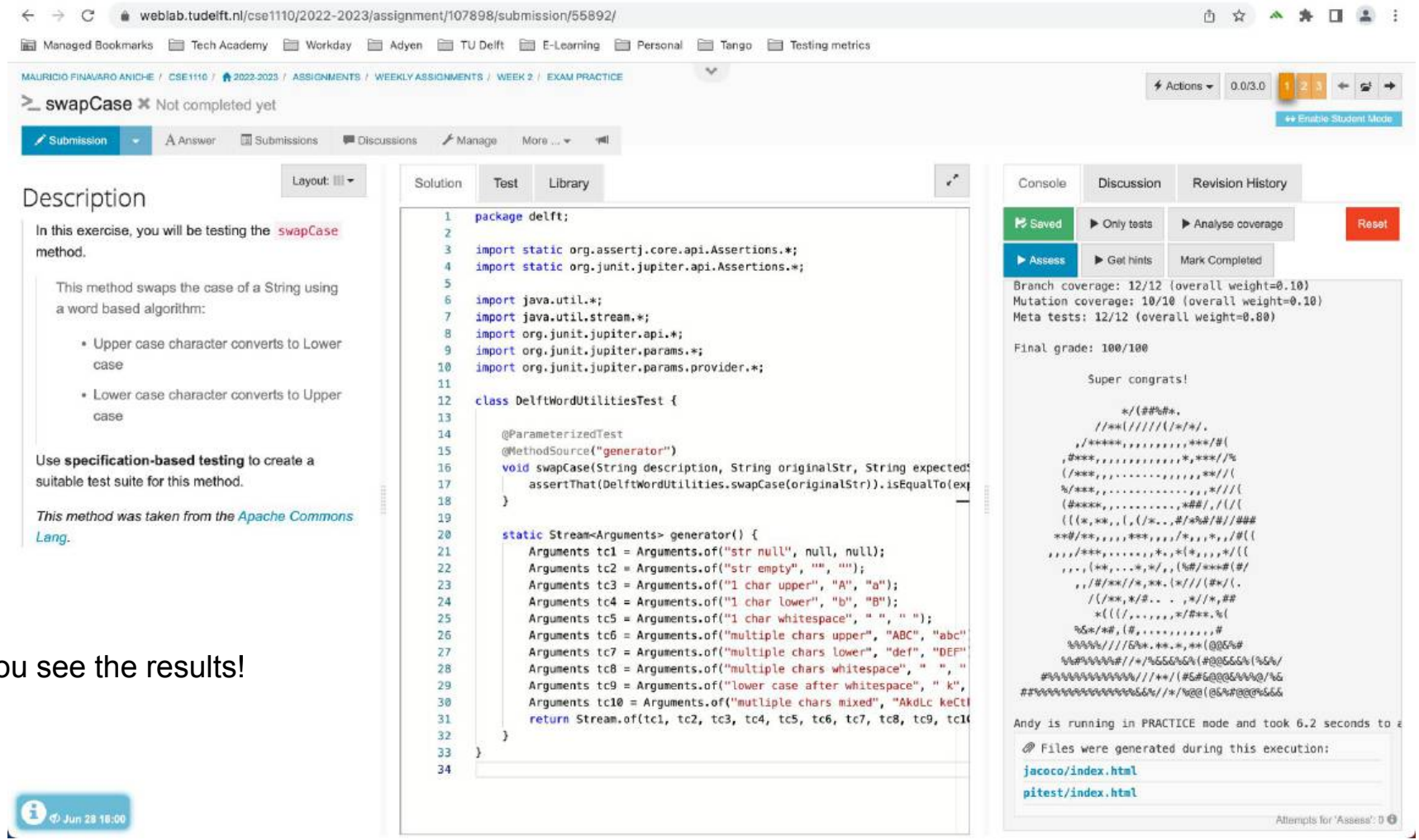


Inside a coding assignment (this is the method they need to test)



Multiple running options

You write your tests in the Solution tab.



You see the results!

Fine-grained analytics per exercise (useful to grade the exam)

Search Table

Use '|' to match *multiple|terms*.

Showing: 1 - 100 / 397 (397)

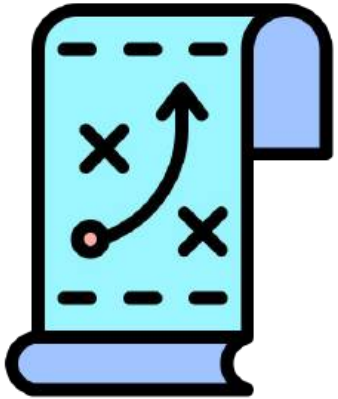
Domain t...enerator	Pass	Computed Grade (1.0)	Metascore: Branch coverage	Metascore: Mutation coverage	Metascore: Code checks	Metascore: Meta tests	Metascore: does not work at the beginning of the file	Metascore: off by one error at the end of the file	Metascore: does not preserve relative indentation	Metascore: does not trim indentation correctly with shorter blank/whitespace lines in snippet	Metascore: places arrows on all lines
Search...	Sear	Search...	Search...	Search...	Search...	Search...	Search...	Search...	Search...	Search...	Search...
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1
7.2	true	7.2	12	19	0	5	1	1	1	0	1

Challenges in building such a tool

- Make it performant
  - First version would simply do "mvn test coverage pitest static-analysis" and the parse the results. Too slow.
  - Makes native use of JaCoCo, Pitest and JUnit.
  - Around 2 seconds in my M1, around 5 seconds in AWS lambda. More in complex PBT assignments.
- Too much classloading magic and shadowing
  - All these tools bring their own classloaders, putting all of them together was tricky
  - Pitest shadows JUnit, it's hard to bump dependency versions.
- Create a DSL for configuration that's not too hard to use
  - First version was a simple text file with conventions. Became too hard to express everything.
  - Java version is a bit better although requires some knowledge to use it.
  - Trying YAML (as anyone, I'm sure we'll regret that soon enough)
- The Maven plugin
  - Due to the so much magic we do, implementing a Maven plugin was harder than we thought.
  - Things would fail, classes wouldn't be found, etc…

Ongoing and next steps

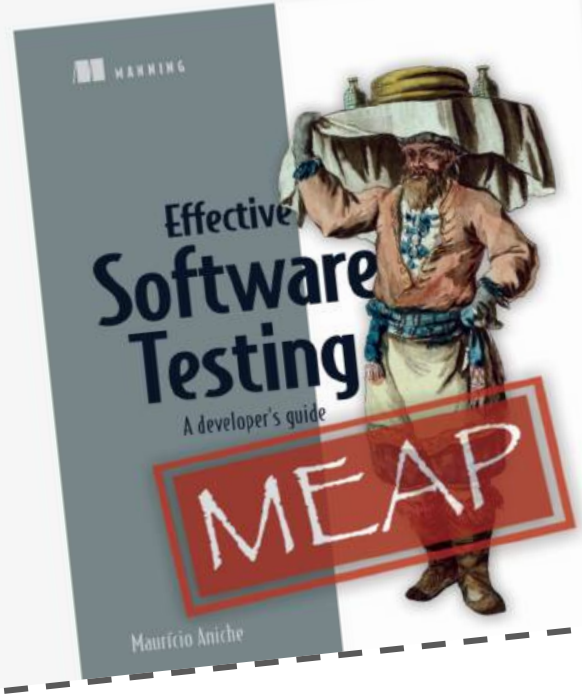
- ~~Support Selenium tests~~
  - Unit tests, mocking, property-based tests, SQL tests, Selenium tests.
- ~~Analytics~~
- ~~Performance improvements~~
- Make it easy for others to adopt
  - You can use WebLab for 5 euros per student
  - Maven plugin
  - AWS lambda if you want to embed in your IDE
  - Github, help me!!! (Challenge: get approved by Delft for exams)
- Write a paper!
  - I leave academia, but academia doesn't leave me!



Based on my book!

Learn how to write tests in an effective and systematic way!

- Create tests that really find bugs
- Use code coverage in the right away
- When to go for unit, integration or system testing
- Use mocks and stubs to simplify your unit testing
- Write property-based tests
- Design testable systems
- Write maintainable test code



https://www.effective-software-testing.com/

TU Delft education fellow

We already know by now the importance of programming skills. But whenever we think of becoming programmers, we think of learning a programming language or algorithms and data structures. However, a skill that is as important as writing programs is to test those programs, and make sure that they actually work. After all, society depends on working software!

The goal of my fellowship is to develop a platform fully focused on software testing education. Software testing education offers lots of challenges for both teachers and students. In this platform, students will exercise different testing skills, such as writing automated test cases that achieve a certain level of coverage, chasing bugs in real-world programs, and developing code in a test-driven manner. Teachers will also leverage the analytics data to understand how their students are doing. The platform will also keep proposing new exercises based on real-world open source bugs.

https://www.tudelft.nl/teachingacademy/get-inspired/education-fellows

Thanks to the entire team!

- Martin Mladenov, the hand!
- Jan Warchocki
- Paul Hübner
- Florena Buse
- Thijs Nulle
- Diman Uzunov
- Nadine Kuo
- Teodor Oprescu
- Yoon Hwan Jeong
- Wouter Polet (earlier versions)
- The WebLab development team

Why is it called Andy?

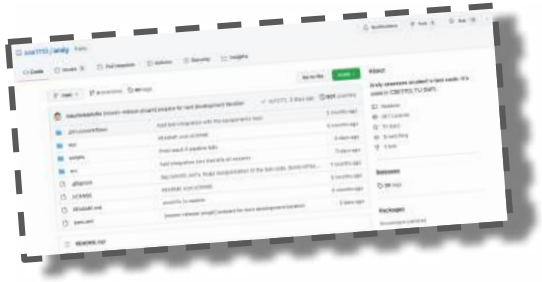
It is a tribute to **Dr. Andy Zaidman**, Professor in Software Quality and Director of Studies at TU Delft.

Andy is a strong proponent of testing. Students meet Andy in their first quarter. Andy teaches them introduction to programming. In weeks 3-4 (!!), Andy already shows them JUnit!

https://azaidman.github.io/

Questions?

- Andy: https://github.com/cse1110/andy
- Manual: README in Andy's repository



Give us a ⭐ !

Maurício Aniche

mauricioaniche@gmail.com

@mauricioaniche

TU Delft