Assignment 1

Course: IN3240/4240/ Software Testing

Delivery format: Deliver in a zip-file with sensible filenames respectively to the exercises. *

Delivery Info: Required to work in groups of 2-4

Assignment deadline: **Friday 27.03.2026** kl. **23:59**

General infomation:

This assignment is divided into four (4) parts.

The first part is related to the guest lecture on test automation in PlayWright, which will be given by **Itera** on Thursday March 12th. The details of this part will be published the same day.

The second part is about getting familiar with the online static analysis tool SonarQube – used for detecting bugs, vulnerabilities and other issues in code.

The third part focuses on unit testing, where you will apply **Test-Driven Developement** in **Cyber-Dojo** in order to solve programming exercises.

The fourth and last part is about static testing techniques, i.e. Reviews.

It is important to **read the assignment carefully and perform the tasks as described**.

*Students are responsible for forming groups themselves. If you are able to form the group with all members on Devilry's group function - submit the assignment as usual. If you cannot form a group with all group members in Devilry due to different course versions (IN3240 versus IN4240) - let only one group member submit the assignment, but when you submit the assignment, write the names, usernames, and course versions of all members in the comments field. Also write this on the front page of the assignment.

Part 1 - Test automation in Playwright
Test automation involves scripting tests to enable focus on higher-value tasks, such as exploratory testing, while automating time-consuming and repetitive tests like regression testing. By adopting test automation, we can execute a larger number of tests more quickly and comprehensively, ensuring greater consistency and objective assessment. This approach significantly increases test coverage compared to manual testing.
The assignments are developed by Iteria and is based on the guest lecture held on March 13, along with the exercises conducted during the same week.
Delivery: Answers to the exercises.
Go to Part 1

Part 2 - Static Analysis
Static analysis allows us to analyze the source code without actually executing it. This way, we can evaluate the System Under Test (SUT) and detect errors, vulnerabilities, and other weaknesses in the software that are difficult to uncover during dynamic testing. Static analysis is therefore an important tool in security testing. and in Continuous Delivery. Static analysis is also crucial for detecting defects early in the software development process .
Delivery: Answers to the questions.
Go to Part 2

Part 3 - Unit testing and Test Driven Development
testing practically possible. In Test Driven Development (TDD), each test is written before the related code, and testing is thus fully integrated into the development process.
Delivery: Four files, the session IDs and screenshot of the session website.
Go to Part 3

Part 4 - Static testing - Reviews
Among the most powerful testing techniques, we find reviews. In principle, all software work products can be tested using various review techniques, with the goal of helping the author improve the quality of the document.
Delivery: An essay
Go to Part 4