This work was supported by the project PID2022-137646OB-C32 under Grant MCIN/AEI/10.13039/501100011033/FEDER, UE

SCAN ME

Software System Testing assisted by Large Language Models: An Exploratory Study

Cristian Augusto, Jesús Morán, Antonia Bertolino, Claudio de la Riva and Javier Tuya

{augustocristian, moranjesus, claudio, tuya**} @uniovi.es -** antonia.bertolino@isti.cnr.it

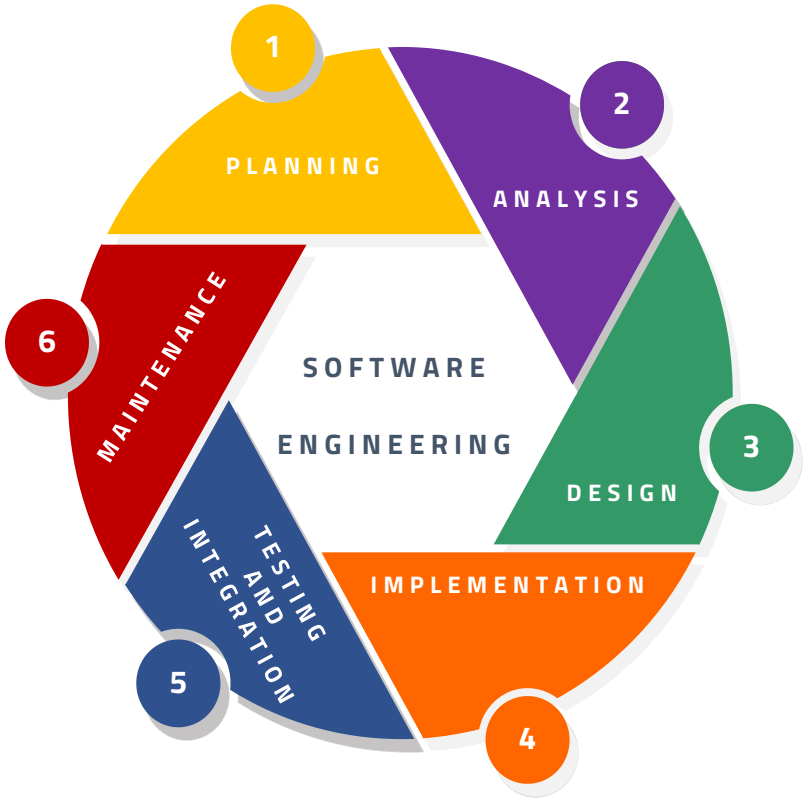
36th International Conference on Testing Software and Systems (ICTSS) 2024

London, Great Britain 2024

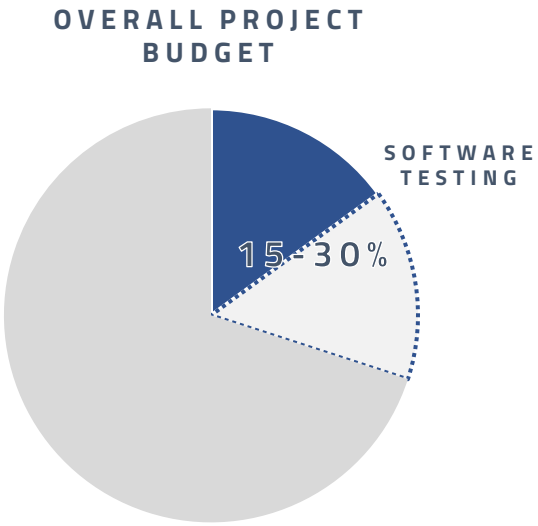
Universidad de Oviedo

Context

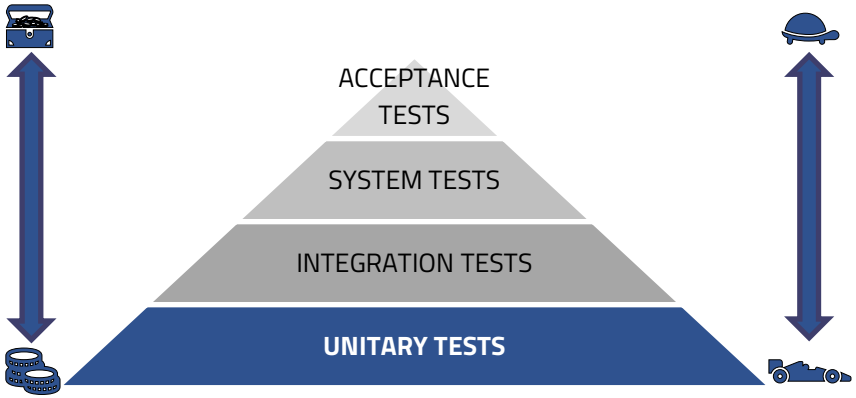
- Large Language Models → crashed into Software Engineering:
  - Generating code, documentation or reports
  - Explainability of code/patches
  - Finding-repairing bugs
  - TOILs
- Software Testing→ room for application



Motivation



- Software testing takes **15-30%** of the total project budget [1,2].
- LLMs in Software testing → mostly applied in the unitary level
- System Testing (E2E) → **EXPENSIVE**:
  - Business knowledge
  - Full Stack knowledge



[1] Lionel Sujay Vailshery. (2022, February 21). QA and testing budget allocation 2012-2019 | Statista.

[2] How Much Does Software Testing Cost in 2024? | by Creole Studios | Medium.

Objective

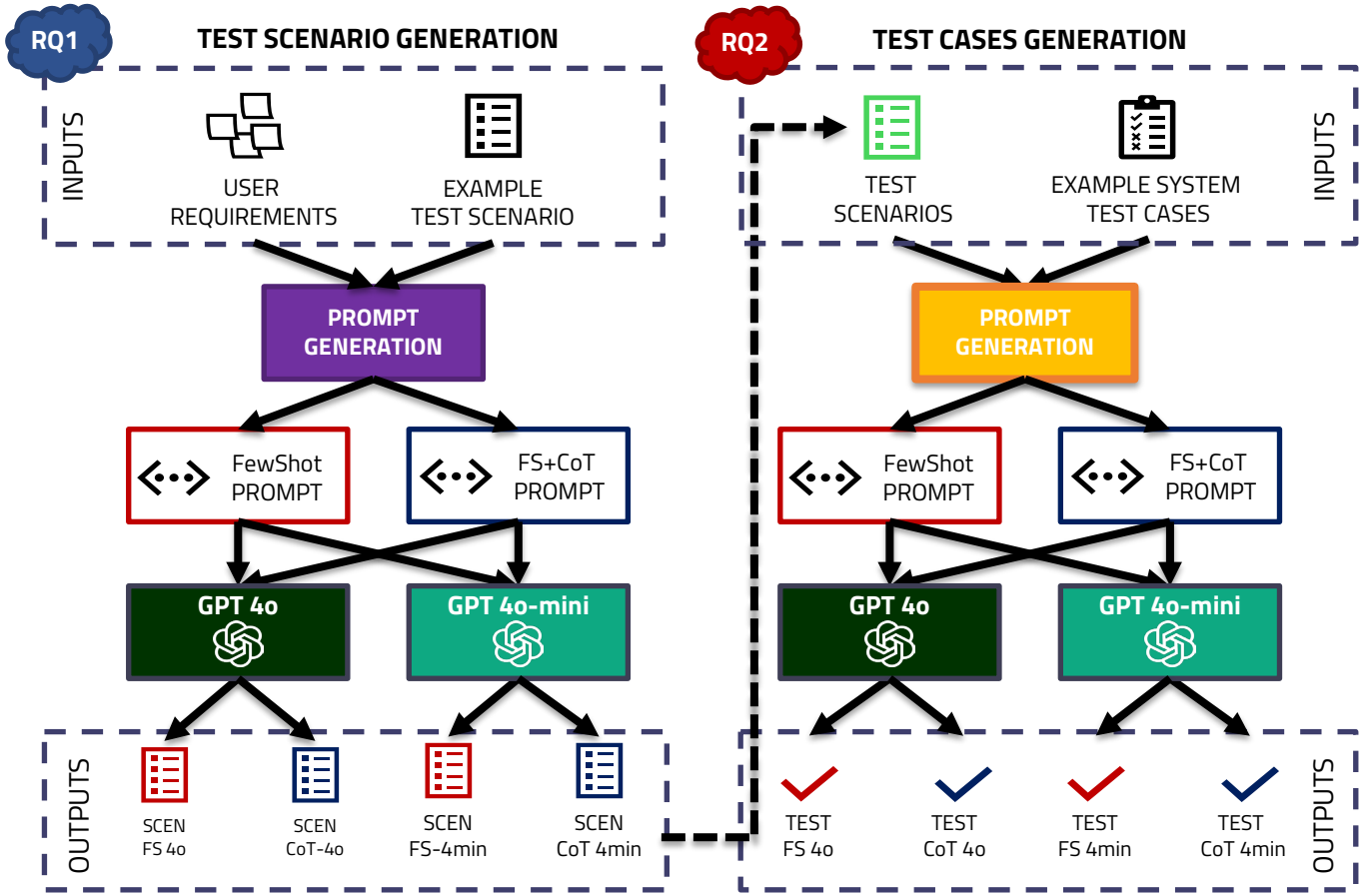
EXPLORE THE CAPABILITY OF

LARGE **L**ANGUAGE **M**ODEL**S**

OF SUPPORT

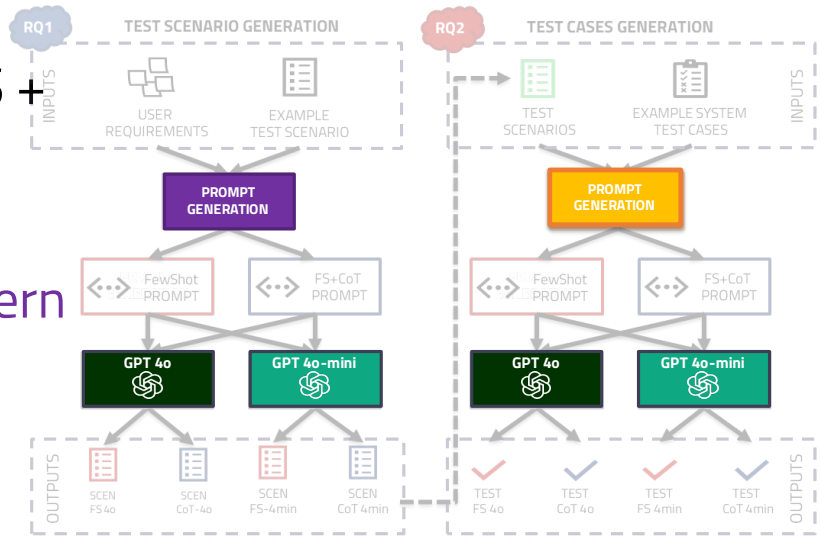
SYSTEM TEST DESIGN AND IMPLEMENTATION

Approach: Overview



Prompt Engineering

- 2 Prompting techniques:
  - Few-Shot (FS)
  - Few-Shot+ Chain of Thought (FS + CoT)
- 2 Prompting patterns:
  - Generate Scenarios: Recipe Pattern
  - Generate System Test Code: Context Manager Pattern
- 2 Popular Transformer-based LLMs:
  - GPT-4o
  - GPT-4o-mini



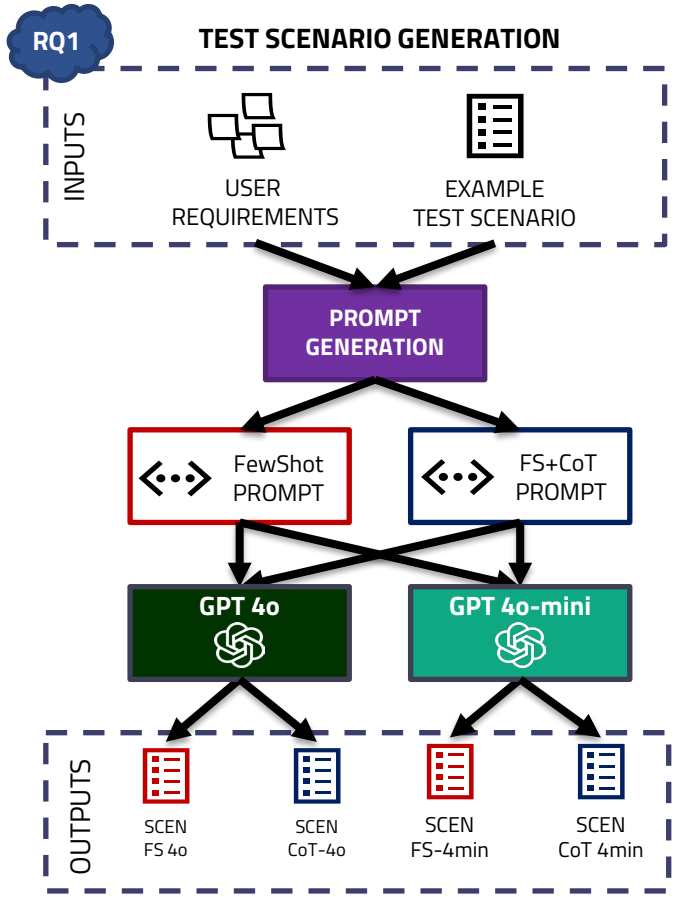
Demonstrator: FullTeaching

- Web platform that eases online teaching: web classrooms, video calls, forums…
- Real-world European project demonstrator:
  - System test suite composed of 21 test cases
  - User requirements used during the development.

https://github.com/giis-uniovi/retorch-st-fullteaching

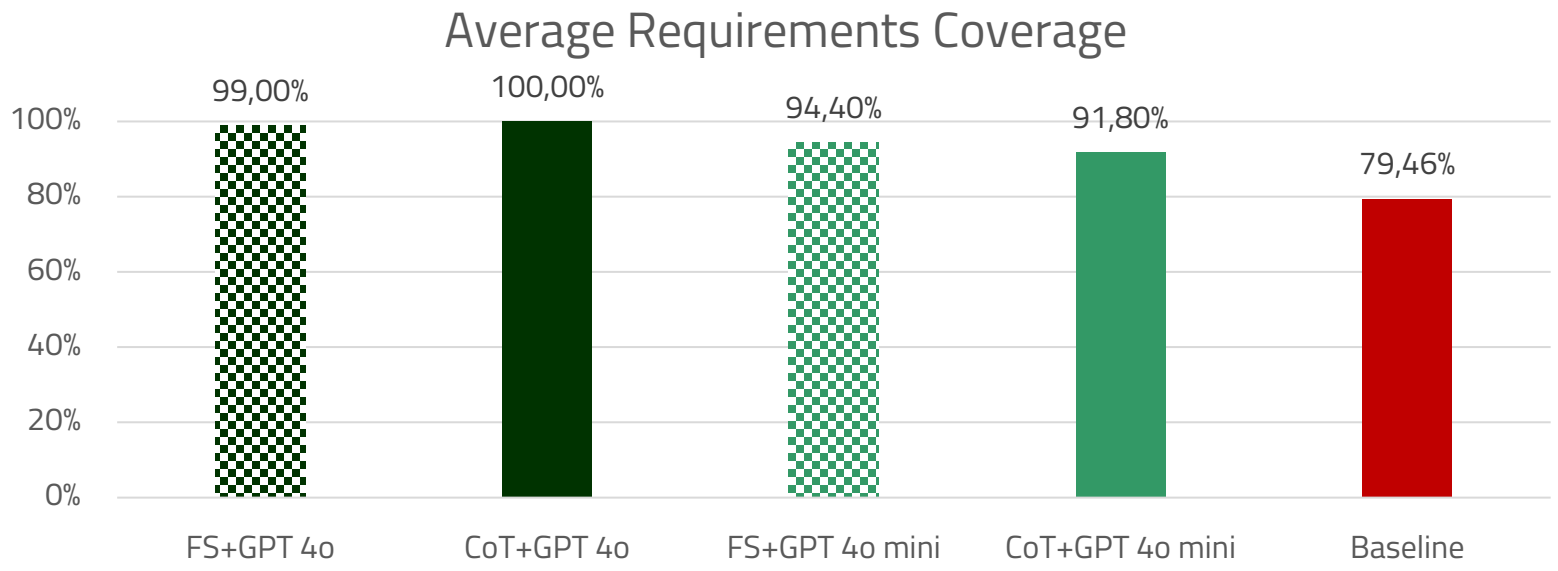
Test Scenarios Generation

- Inputs:
  - User Requirements
  - Test Scenario example
- Outputs: Test scenarios
  - Few-Shot + 4o
  - Few-Shot + 4o mini
  - CoT + 4o
  - CoT + 4o mini



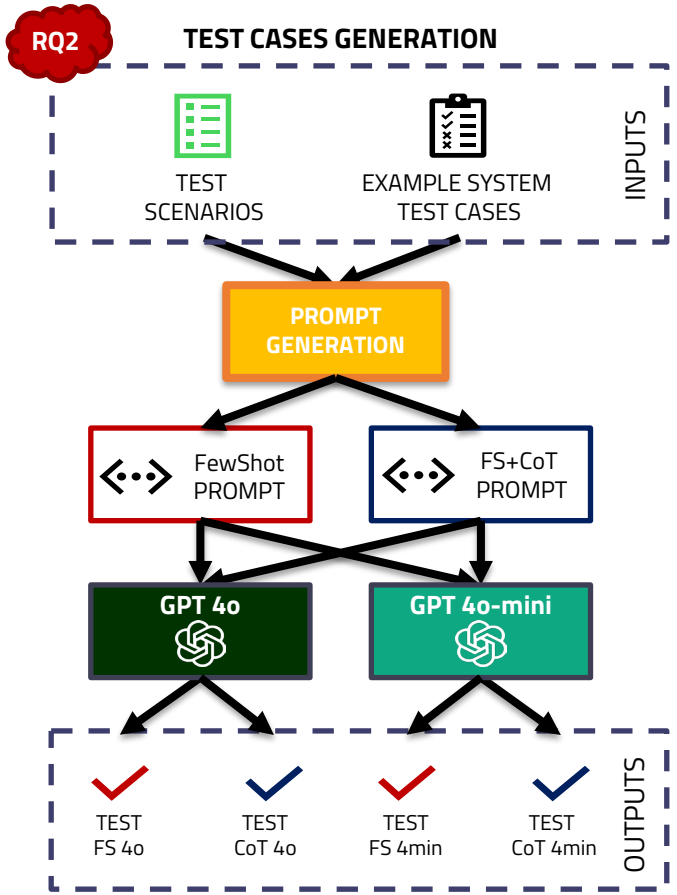
Test Scenarios Generation: Results

- Metric: User Requirements Coverage (URC)
- Baseline: URC Test Suite



System Test Cases Generation

- Inputs:
  - Best Test Scenario
  - System Test Code
- Output: Test cases
  - Few-Shot + 4o
  - Few-Shot + 4o mini
  - CoT + 4o
  - CoT + 4o mini



System Test Cases Generation: Results

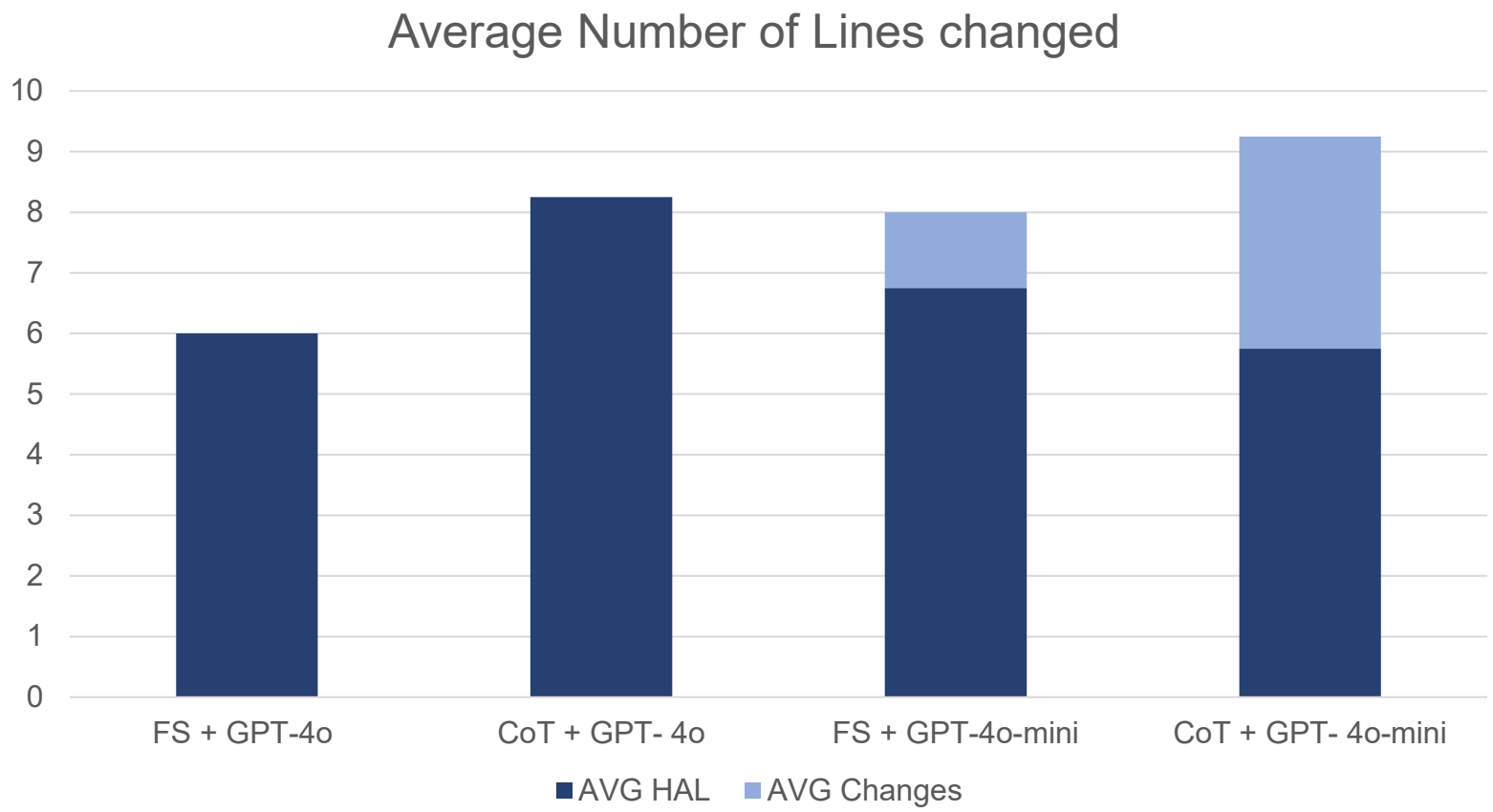
- Metric: Number of lines changed → Test pass/work
- Types of changes:
  - Requested by the LLM
  - Hallucinations (HAL)

```
@ParameterizedTest
@MethodSource("data")
void viewEnrolledCoursesTest(String mail, String password, String role) {
    // Step 1: User logs into the application
    this.slowLogin(user, mail, password);

    // Step 2: User navigates to the dashboard
    try {
        if (NavigationUtilities.amINotHere(driver, COURSES_URL.replace("__HOST__", HOST))) {
            NavigationUtilities.toCoursesHome(driver);
        }

        // Step 3: System displays a list of courses the user is enrolled in
-       Wait.notTooMuch(driver).until(ExpectedConditions.visibilityOfElementLocated(By.id("courses-list")));
+       Wait.notTooMuch(driver).until(ExpectedConditions.visibilityOfElementLocated(By.id("course-list")));
-       List<WebElement> courses = driver.findElements(By.className("course-item"));
+       List<WebElement> courses = driver.findElements(By.className("course-list-item"));
```

System Test Cases Generation: Results



Open Data: Results and data available in GitHub + Zenodo

Replication Package

https://github.com/giis-uniovi/retorch-llm-rp

Raw Datasets

https://zenodo.org/records/13761150

Conclusions

- LLMs arise as a promising tester support tool:
  - Generate an initial set of scenarios
  - Deriving test sets from the test scenarios
- LLMs are a great tool but need to be supervised by a tester
- Challenges:
  - Reduce-improve the scenarios generated
  - Reduce the hallucinations generating test code.

Future work

- Evaluate the approach with different tunings, models and test suites.
- Explore how prompting techniques can be improved to achieve better results
- Explore how LLMs could support the generation of negative test cases

This work was supported by the project PID2022-137646OB-C32 under Grant MCIN/AEI/10.13039/501100011033/FEDER, UE

Questions?

Cristian Augusto, Jesús Morán, Antonia Bertolino, Claudio de la Riva and Javier Tuya

{augustocristian, moranjesus, claudio, tuya**} @uniovi.es -** antonia.bertolino@isti.cnr.it

36th International Conference on Testing Software and Systems (ICTSS) 2024

London, Great Britain 2024