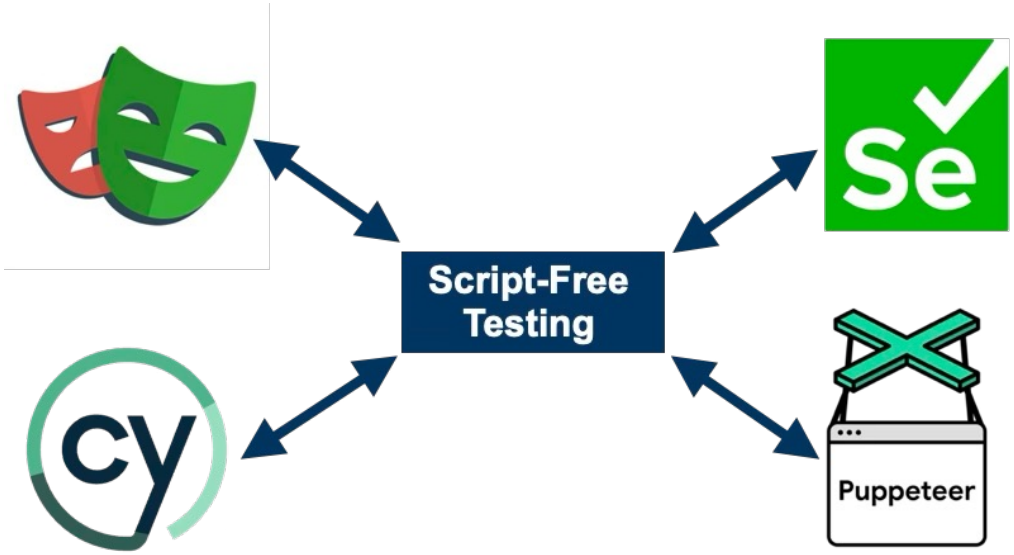

RUHR-UNIVERSITÄT BOCHUM

Script-Free Testing Framework

Language-Agnostic Migration Framework for Script-Free Testing

Thiago Santos de Moura, M.Sc.
Prof. Dr. Yannic Noller
Software Quality group

Motivation and Background

- Web GUI test automation relies heavily on scripted test cases, utilizing tools such as Selenium[1], Cypress[2], Puppeteer[3], and Playwright[4]. These scripts rely on fragile locators that often break when the web pages evolve [5] [6].
- To address this, the script-free testing tool [7] [8] offers an alternative by enabling test cases to be written in natural or Domain-Specific Language (DSL), avoiding the use of locators, and automatically converting them into Selenium test cases.

- This project proposes a language-agnostic framework that converts script-based test cases from the four most used web testing tools into a DSL version for script-free testing and then allows the generation of commands for all four testing tools.
- Enabling this two-way conversion between script-based and script-free formats would lower maintenance costs, enhance test portability, and encourage reusable, resilient test automation. Researchers and practitioners would benefit from such a framework.

Student Task and Responsibilities

- Develop a framework to automatically convert script-based test cases (from Selenium, Cypress, Puppeteer, and Playwright) into a DSL compatible with an existing script-free testing tool. Extend the tool to support generating executable commands for Cypress, Puppeteer, and Playwright.

- Students must first expand the DSL to cover a complete range of web GUI actions. This includes navigation, waits, assertions, clicks, text input, and option selection. The goal is to ensure that all operations in script-based tests can be expressed in the DSL.
- Next, they should implement a conversion strategy. This can be an AST-based transpiler that parses and rewrites test scripts into the DSL, or an LLM-based agent using prompt engineering or fine-tuning on paired examples of script-based and DSL tests.
- Finally, the script-free tool must be extended to optionally generate test cases in Cypress, Puppeteer, and Playwright. The framework implemented must be evaluated using the E2EGit web GUI test suites [9], and all findings must be thoroughly documented.

Minimum Expectation and Extensions

- The minimum expected result is the implementation of a framework that converts script-based test cases into a DSL supported by the existing script-free testing tool, along with the generation of executable commands for at least one testing tool, besides Selenium.
- The group must extend the DSL to represent a full set of web GUI actions and implement at least one conversion strategy, either AST-based or LLM-based.
- The framework should be designed for maintainability and extensibility, with proper testing. The group must validate their solution using at least part of the E2EGit test suites. A live demonstration of its conversion and execution capabilities must be given.

- Potential extension points:
  - Explore and compare both conversion strategies to evaluate robustness and accuracy.
  - Conversion of all available web GUI test suites in the dataset and perform a detailed evaluation of the execution fidelity.

Initial Timeplan

Week 1 (13.10. – 19.10.)[M1]	Kick-Off & Introduction
Week 2 (20.10. – 26.10.)	Planning, Requirements Engineering, and Design
Week 3 (27.10. – 02.11.)[M2]	
Week 4 (03.11. – 09.11.)	***1st Coding Cycle*** Implementation of prototype (no complete workflow implementation expected)
Week 5 (10.11. – 16.11.)	
Week 6 (17.11. – 23.11.)	
Week 7 (24.11. – 30.11.)[M3]	
Week 8 (01.12. – 07.12.)	***2nd Coding Cycle*** Implementation of complete workflow
Week 9 (08.12. – 14.12.)	
Week 10 (15.12. – 21.12.)[M4]	

Week 11 (22.12. – 28.12.)*	***3rd Coding Cycle*** Improvements and extensions
Week 12 (29.12.. – 04.01.)*	
Week 13 (05.01. – 11.01.)	
Week 14 (12.01. – 18.01.)	
Week 15 (19.01. – 25.01.) [M5]	Finalization code (code freeze after week 15)
Week 16 (26.01. – 01.02.)	Final documentation and report writing/submission
Week 17 (02.02. – 08.02.)[M6]	

** Christmas Holidays*

Milestones:

- M1: Project Kick-Off
- M2: Code Design Submission
- M3: Demonstration of Prototype
- M4: Demonstration of Complete Workflow
- M5: Final Code Submission
- M6: Final Report Submission

Working Mode

- Weekly meetings with advisor (will be arranged taking into account all schedules)
- Expected are at least one additional weekly group-internal meeting and active discussions on Slack/Discord
- Kick-Off Meeting: in the week of 13th October 2025 (details will be announced)

Other Information

- **Prerequisites:** Programming experience, preferably in JavaScript or Python, is needed for this project. Familiarity with web testing tools such as Selenium, Cypress, Puppeteer, or Playwright would be beneficial. Prior attendance in courses related to Software Engineering or Web Engineering is highly recommended.
- **Deliverables:** Source code, its documentation, and a publishable report (incl. the evaluation results), ideally to submit to a conference (e.g., as a Demo paper) or at least publish as a technical report on arxiv.org.
- **Number of Participants:** 2-4
- **Target Group:** Bachelor and Master students
- **Industrial partner:** None (done at RUB)

Contact

Thiago Santos de Moura, M.Sc.

Research Associate / PhD Student

- Room: MC 4.115
- E-Mail: thiago.santosdemoura@rub.de
- Office hours: By Arrangement

- https://scholar.google.com/mourats/
- https://www.linkedin.com/in/mourats/

References

[1] https://www.selenium.dev/

[2] https://www.cypress.io/

[3] https://pptr.dev/

[4] https://playwright.dev/

[5] Nass, Michel, Emil Alégroth, and Robert Feldt. "Why many challenges with GUI test automation (will) remain." *Information and Software Technology* 138 (2021): 106625. https://doi.org/10.1016/j.infsof.2021.106625

[6] Hammoudi, Mouna, Gregg Rothermel, and Paolo Tonella. "Why do record/replay tests of web applications break?" *2016 IEEE International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 2016. https://doi.org/10.1109/icst.2016.16

[7] Kirinuki, Hiroyuki, et al. "NLP-assisted web element identification toward script-free testing." *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2021. https://doi.org/10.1109/icsme52107.2021.00072

[8] Kirinuki, Hiroyuki, et al. "Web element identification by combining NLP and heuristic search for web testing." *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2022. https://doi.org/10.1109/saner53432.2022.0012

[9] Di Meglio, Sergio, et al. "E2EGit: A Dataset of End-to-End Web Tests in Open Source Projects." *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*. IEEE, 2025. https://doi.org/10.1109/msr66628.2025.00121