Sergio Di Meglio

End-to-End Web Testing: A Focus on Adoption and Maintainability

Tutor: Sergio Di Martino

co-Tutor: Dott. Fabio Scippacercola

Cycle: XXXVIII

Year: Third

Candidate’s information

- **MSc degree**: Computer Science
- **PhD start date – end date**: 1/11/2022 - 31/10/2025
- **Scholarship type:** PNRR-DM 352
- **Partner company:** Fervento srl
- **Periods abroad:** 1/10/2024 – 1/04/2025, *@Software Language (SOFT) Lab*, Department of Computer Science, Vrije Universiteit Brussels
- **Periods in company**: A total of 18 months, typically three or more days per week.

Summary of study activities

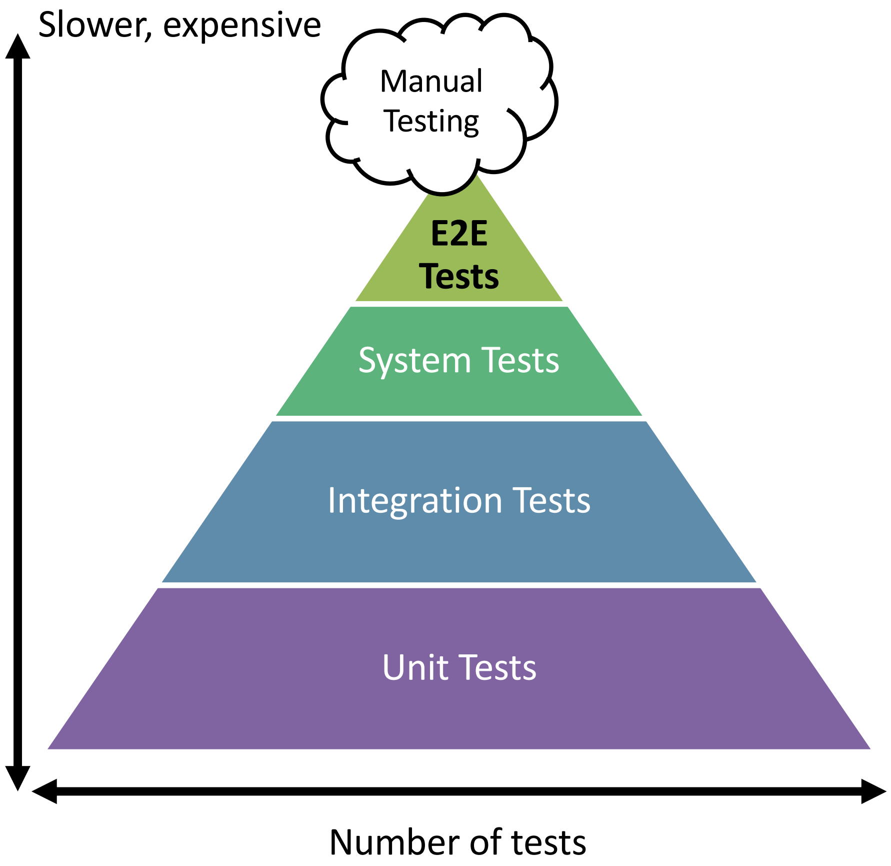
- **Ad hoc PhD courses / schools:**
  - ✓ Virtualization technologies and their applications
  - ✓ How to boost your PHD
  - ✓ Big Data Architecture and Analytics
  - ✓ Artificial Intelligence and Natural Language Processing
  - ✓ Using Deep Learning Properly
  - ✓ Software Testing
  - ✓ International Summer School on Software Engineering
- **Conference**
  - ✓ 2023 IEEE Conference on Software Testing, Verification and Validation (ICST), Dublin, Ireland, **attended**
  - ✓ 41st IEEE International Conference on Software Maintenance and Evolution (ICSME), Auckland, New Zeeland, 2025
  - ✓ *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR),* Ottawa, ON, Canada
  - ✓ 22th International Conference on Evaluation and Assessment in Software Engineering (EASE), Salerno, Italy, **attended**
  - ✓ 2025 IEEE Conference on Software Testing, Verification and Validation (ICST), Napoli, Italy, **attended**
  - ✓ *Euromicro Conference on Software Engineering and Advanced Applications. Springer, Salerno, 2025*
  - ✓ *37th International Conference on Software Engineering & Knowledge Engineering. (SEKE), Pompei, 2025*
  - ✓ International Conference on Software Measurement, IWSM Mensura 2023, Rome, Italy, **attended**

Summary of study activities

- **Tutorship:**
  - ✓ Delivered approximately 15 lectures across the Object-Oriented Programming, Software Engineering, and Software Project Management and Evolution courses for B.Sc. and M.Sc. students in Computer Science.
  - ✓ Co-supervised 12 Bachelor’s and 1 Master’s thesis student.
- **Awards:**
  - ✓ **Data/Tool Track Distinguished Dataset Award** at 22nd International Conference on Mining Software Repositories (MSR) for the paper: E2EGit: A Dataset of End-to-End Web Tests in Open-Source Projects.
  - ✓ **Best Paper Award -Testing Tools and Data Showcase** at IEEE Conference on Software Testing, Verification and Validation (ICST) for the paper: E2E-Loader: A Tool to Generate Performance Tests from End-to-End GUI-Level Tests.
  - ✓ **Best Paper Award** at 37th Inter-national Conference on Software Engineering & Knowledge Engineering (SEKE) for the paper: Large Language Models in the Travel Domain: An Industrial Experience

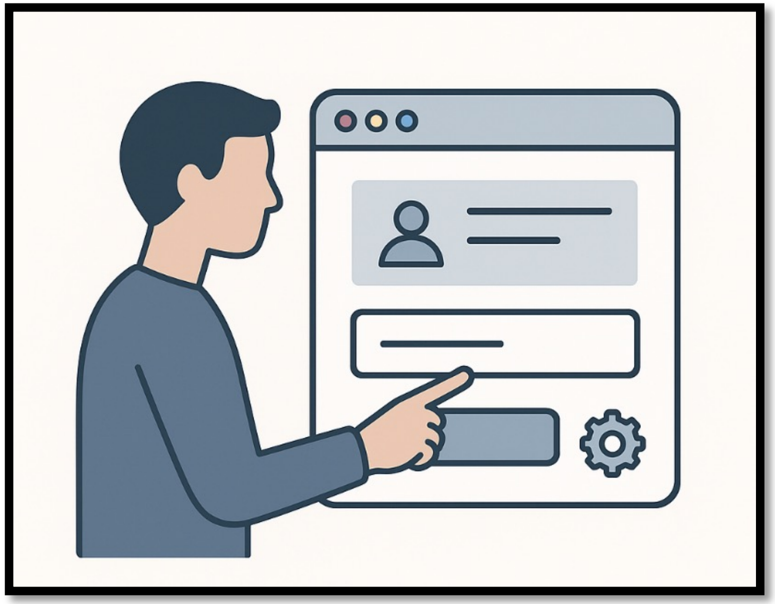
Research area

- **End-to-End** (E2E) Web testing is a testing approach that validates the **entire workflow** of an application.
- Simulating **realistic usage scenario** from an end-user's perspective.



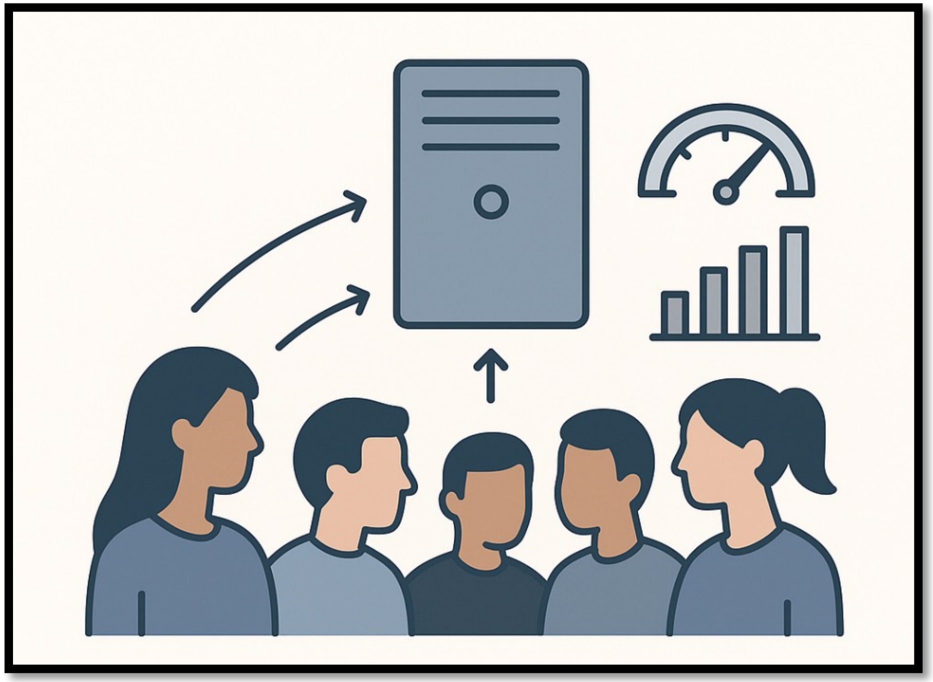
Research area

- *Functional E2E Web Testing* (**GUI-Level**):
    - Focuses on replicating realistic user interaction through the **browser** (e.g., clicks, form inputs etc.)
    - Relies on browser **automation frameworks** such as **Selenium**, **Cypress**, etc.
    - Used to verify that features work as expected from user’s perspective.



Research area

- *Non-Functional E2E Web Testing* (**Performance**):
  - Focuses on evaluating the application under varying **workload** conditions.
  - In the web domain, a workload represents multiple **concurrent users** interacting with the system.
  - Common workload generator tools are **Apache JMeter** and **Locust**



Research results overview

- **Large-scale empirical analysis** of the **adoption** and **maintenance dynamics** of functional (GUI-level) and non-functional (performance) End-to-End testing in web applications.
- Definition and validation of a **taxonomy** of **bad practices** in performance testing through a systematic feedback-driven study.
- Introduction and industrial validation of a new methodology and tool for **automatic workload generation** from functional E2E test executions.

Research products

[J1]	**Sergio Di Meglio**, Luigi Libero Lucio Starace, Valeria Pontillo, Ruben Opdebeeck, Coen De Roover, and Sergio Di Martino, *Investigating the Adoption and Maintenance of Web GUI Testing: Insights from GitHub Repositories,* Information and Software Technology Journal, [Published]
[J2]	**Sergio Di Meglio**, Luigi Libero Lucio Starace, and Sergio Di Martino, *Web App Performance Testing in Industrial Contexts: Supporting Workload Generation with E2E-Loader++,* Journal of System and Software, [In press]
[J3]	**Sergio Di Meglio**, Luigi Libero Lucio Starace, Valeria Pontillo, Luana Martins, Fabio Palomba, and Dario Di Nucci, *Bad Practices in Performance Testing: Insights from a Multi-Perspective Empirical Study,* ACM Transactions on Software Engineering and Methodology Journal, [Under 1st round of revision]
[J4]	**Sergio Di Meglio**, Luigi Libero Lucio Starace, and Sergio Di Martino, *Semi-automated Generation of Web App Performance Tests from End-to-End GUI-Level Tests with E2E-Loader,* Science of Computer Programming Journal, [Under 1st round of revision]
[J5]	**Sergio Di Meglio**, and Luigi Libero Lucio Starace, *Evaluating Performance and Resource Consumption of REST Frameworks and Execution Environments: Insights and Guidelines for Developers and Companies,* IEEE Access, [Published]

Research products

[C1]	Ermanno Battista, Sergio Di Martino, **Sergio Di Meglio**, Fabio Scippacercola and Luigi Libero Lucio Starace, *E2E-Loader: A Framework to Support Performance Testing of Web Applications,* 2023 IEEE Conference on Software Testing, Verification and Validation (ICST), [Published]
[C2]	**Sergio Di Meglio**, Luigi Libero Lucio Starace, Valeria Pontillo, Ruben Opdebeeck, Coen De Roover, and Sergio Di Martino, *Performance Testing in Open-Source Web Projects: Adoption, Maintenance, and a Change Taxonomy,* 41st IEEE International Conference on Software Maintenance and Evolution (ICSME), [Published]
[C3]	**Sergio Di Meglio**, Luigi Libero Lucio Starace, Valeria Pontillo, Ruben Opdebeeck, Coen De Roover and Sergio Di Martino, *E2EGit: A Dataset of End-to-End Web Tests in Open Source Projects,* 2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR), [Published]
[C4]	**Sergio Di Meglio**, and Luigi Libero Lucio Starace, *Towards Predicting Fragility in End-to-End Web Tests,* 22th International Conference on Evaluation and Assessment in Software Engineering (EASE), [Published]
[C5]	**Sergio Di Meglio**, Luigi Libero Lucio Starace and Sergio Di Martino, E2E-Loader: A Tool to Generate Performance Tests from End-to-End GUI-Level Tests, 2025 IEEE Conference on Software Testing, Verification and Validation (ICST), [Published]

Research products

[C6]	Marco De Luca, Sergio Di Martino, **Sergio Di Meglio**, Anna Rita Fasolino, Luigi Libero Lucio Starace, and Porfirio Tramontana., *Rookie Mistakes: Measuring Software Quality in Student Projects to Guide Educational Enhancement,* Euromicro Conference on Software Engineering and Advanced Applications, [Published]
[C7]	**Sergio Di Meglio**, Valeria Pontillo, and Luigi Libero Lucio Starace, *REST in Pieces: RESTful Design Rule Violations in Student-Built Web Apps,* Euromicro Conference on Software Engineering and Advanced Applications, [Published]
[C8]	**Sergio Di Meglio**, Aniello Somma, Luigi Libero Lucio Starace, Fabio Scippacercola, Giancarlo Sperlì, and Sergio Di Martino, *Large Language Models in the Travel Domain: An Industrial Experience,* 37th International Conference on Software Engineering & Knowledge Engineering, [In press]
[C9]	**Sergio Di Meglio**, Luigi Libero Lucio Starace, and Sergio Di Martino, *Starting a New REST API Project? A Performance Benchmark of Frameworks and Execution Environments, IWSM-Mensura, [Published]*
[W1]	Marco De Luca, **Sergio Di Meglio**, Anna Rita Fasolino, Luigi Libero Lucio Starace, and Porfirio Tramontana, *Automatic Assessment of Architectural Anti-patterns and Code Smells in Student Software Projects,* 28th International Conference on Evaluation and Assessment in Software Engineering,(EASE+LEARNER), [Published]
[W2]	Aniello Somma, **Sergio Di Meglio**, Fabio Scippacercola, Ermanno Battista, Sergio Di Martino, and Luigi Libero Lucio Starace, *MYCAMPANIA.TRAVEL: Leveraging Generative AI to Enhance Digital Travel Experiences,* Ital-IA Intelligenza Artificiale Thematic Workshops co-located with the 5th CINI National Lab AIIS [In press]

Research products

[DS1]	**Sergio Di Meglio,** *End-to-End Testing in Web Environments: Addressing Practical Challenges,* Doctoral Symposium, 2025 IEEE Conference on Software Testing, Verification and Validation (ICST),[Published]

PhD thesis: Research Gaps & Objectives #1

🚧 ***Research Gap #1***

- Existing studies focus exclusively on tools or frameworks, offering a **fragmented view** on how these practices are actually **adopted** and **practiced**.
- GUI-level testing research emphasizes technical issues (e.g, **frigility**, **flakiness**)
- Performance testing, on the other hand, focus primarily on **tool comparison**.

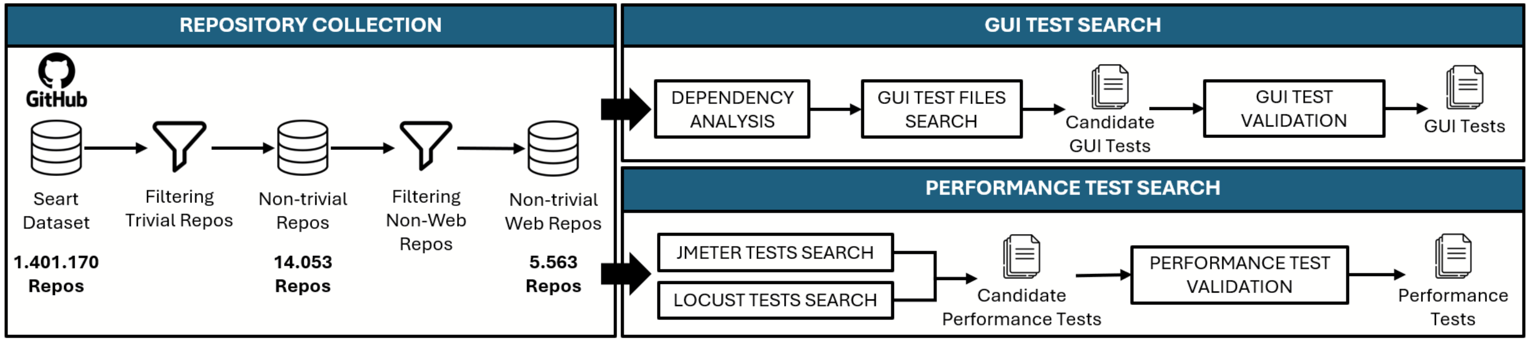
🎯 ***Research Objective #RO.1***

Understanding the **adoption** and **management dynamics** of functional and non-functional End-to-End Testing for web applications.

PhD thesis: Research Contribution #1

- Empirical studies on E2E Testing are **limited** by the **lack** of public **datasets**.
- To address this limitation we developed **E2EGit**, a dataset of **non-trivial** web repositories adopting GUI-level *and/or* Performance Testing, obtained through a **large-scale mining** process on GitHub.

Framework	#repo	#test
Selenium	87	10.464
Playwright	197	18.175
Puppeteer	20	302
Cypress	187	14.733
JMeter	72	244
Locust	13	27



REPOSITORY COLLECTION

GitHub

Seart Dataset
1.401.170 Repos

Filtering Trivial Repos

Non-trivial Repos
14.053 Repos

Filtering Non-Web Repos

Non-trivial Web Repos
5.563 Repos

GUI TEST SEARCH

DEPENDENCY ANALYSIS

GUI TEST FILES SEARCH

Candidate GUI Tests

GUI TEST VALIDATION

GUI Tests

PERFORMANCE TEST SEARCH

JMETER TESTS SEARCH

LOCUST TESTS SEARCH

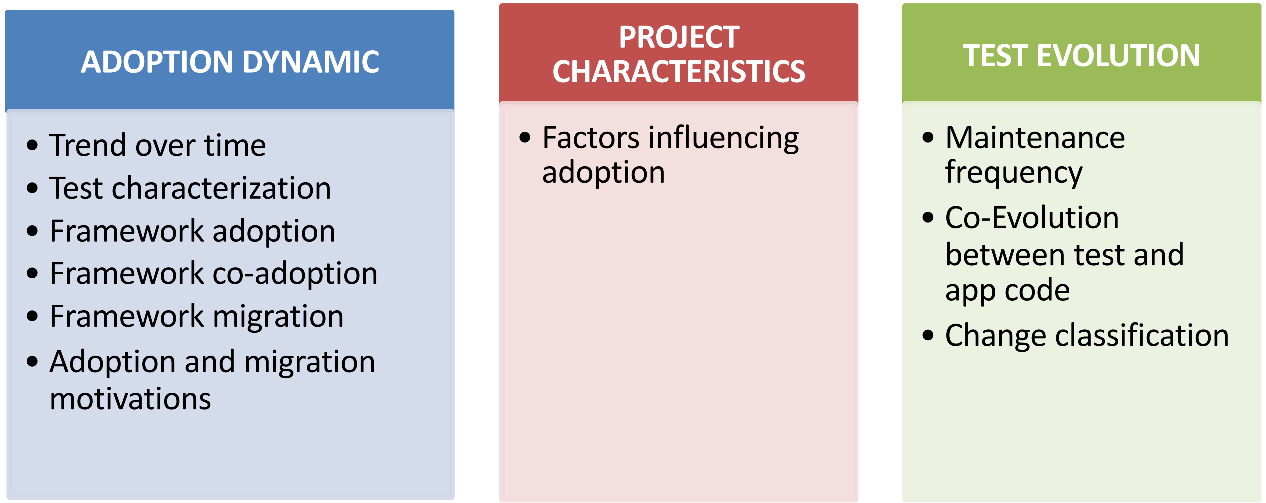
Candidate Performance Tests

PERFORMANCE TEST VALIDATION

Performance Tests

PhD thesis: Research Contribution #1

- Based on the E2EGit dataset, two distinct empirical analyses were conducted:
  - Functional (GUI-level) Testing
  - Non-functional (Performance) Testing
- These analyses were organized into **three complementary aspects:**



ADOPTION DYNAMIC

- Trend over time
- Test characterization
- Framework adoption
- Framework co-adoption
- Framework migration
- Adoption and migration motivations

PROJECT CHARACTERISTICS

- Factors influencing adoption

TEST EVOLUTION

- Maintenance frequency
- Co-Evolution between test and app code
- Change classification

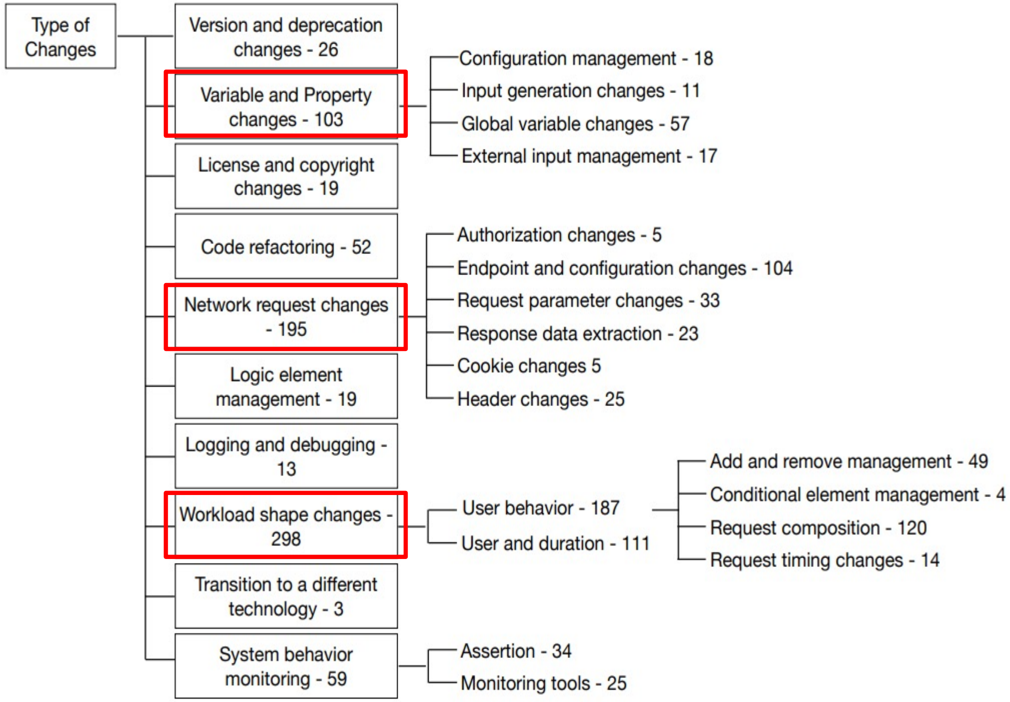
PhD thesis: Research Contribution #1

Key findings related to Performance Testing:

- Performance testing is **rare**, found only ~2% of non-trivial web app.
- Among adopters: 23% introducted it immidiately, 43% withitin 3 years and 34% later.
- Tests are generally **simple**, simulating **single-user behavior** with **few requests**, **low concurrency**, and **short execution times**.

PhD thesis: Research Contribution #1

- **Performance tests are generally long-lived** (mean lifespan* 1,787 days)
- 53% **has never changed** after creation.
- others change **rarely** (mean ≈ 3 vs. 4,428 app code changes).
- 👉 *taxonomy of performance tests changes.*



- Type of Changes
  - Version and deprecation changes - 26
  - Variable and Property changes - 103
    - Configuration management - 18
    - Input generation changes - 11
    - Global variable changes - 57
    - External input management - 17
  - License and copyright changes - 19
  - Code refactoring - 52
  - Network request changes - 195
    - Authorization changes - 5
    - Endpoint and configuration changes - 104
    - Request parameter changes - 33
    - Response data extraction - 23
    - Cookie changes 5
    - Header changes - 25
  - Logic element management - 19
  - Logging and debugging - 13
  - Workload shape changes - 298
    - User behavior - 187
      - Add and remove management - 49
      - Conditional element management - 4
      - Request composition - 120
      - Request timing changes - 14
    - User and duration - 111
  - Transition to a different technology - 3
  - System behavior monitoring - 59
    - Assertion - 34
    - Monitoring tools - 25

* removed or modified more than 66%

PhD thesis: Research Contribution #1

Key findings related to GUI-level testing:

- **Higher adoption** ~10% but **delayed**, increasing after 2018, aligned with new browser automation frameworks releases.
- Cypress and Playwright dominate, while Selenium and Puppeteer are less common.
- Migrations mainly target playwright, driven by **flakiness**, **deprecated** apis, and tool **unification**.

PhD thesis: Research Contribution #1

- GUI tests are generally **durable** (mean lifespan: 637 days) and require **few modifications** (mean: 5.8 changes).
- **Co-evolution** between tests and application code was **statistically confirmed** in 324 repositories.
- **Assertions** and **selector** updates are the most frequent maintenance actions.
- Maintenance is typically handled by a small, **specialized group** who also contribute to the application code.

PhD thesis: Research Gaps & Objectives #2

🚧 ***Research Gap #2***

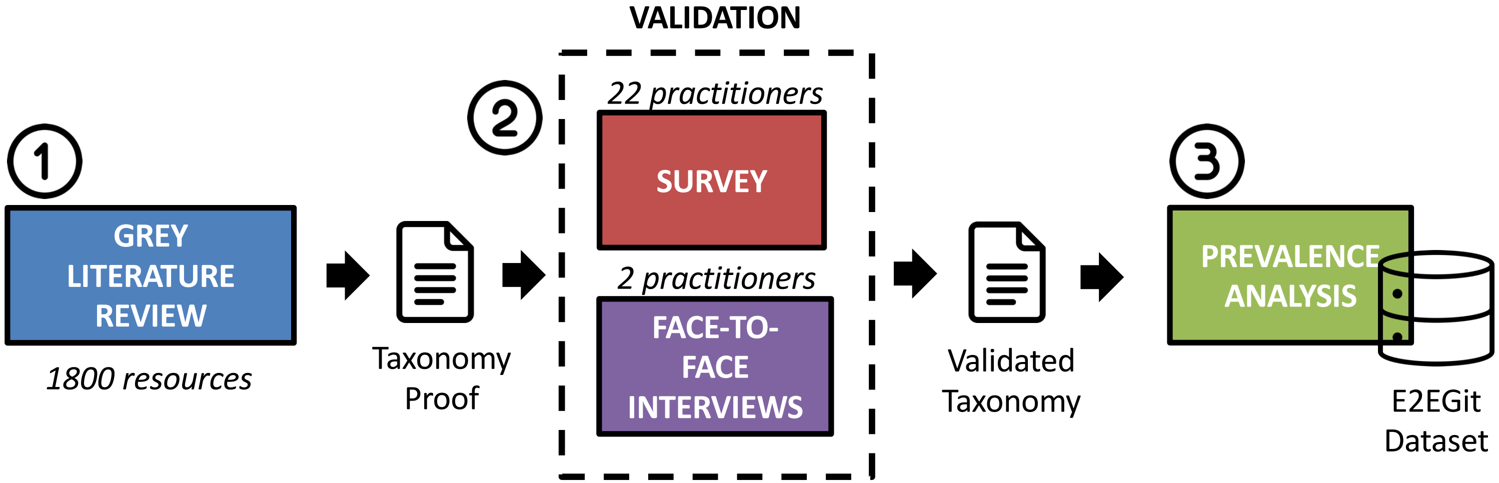
- Extensive research exists on **bad practices** and **test smells** across various contexts (Unit, Integration, GUI-level testing).
- **No systematic effort** has yet focused on Performance Testing.
- Such bad practices can **negatively impact maintenance, reliability, and adoption** of tests.

🎯***Research Objective #RO.2***

Systematically identify, **characterize**, and **assess** bad practices in performance testing, and how they’re **perceived** by practitioners.

PhD thesis: Research Contribution #2

- To achieve RO.2, we conducted a **systematic**, **feedback-driven** process designed to **identify**, **validate**, and refine a **taxonomy** of bad practices in web performance testing.



- The resulting taxonomy comprises **29 bad practices** spanning the **design**, **execution**, and **analysis** phases of performance testing.

PhD thesis: Research Contribution #2

- Examples of representative bad practices are: ***Unrealistic (simplistic) user behavior***, ***Improper data correlations management***, ***No assertion or response validation***.
- Preliminary prevalence analysis focused on **5 automatically detectable bad practices** on E2EGit dataset, revealing that such issues are **widespread** (932 violations out of 244 tests) and **persistent.**
    - Only **3** test were free of any violations.
    - Test history analysis showed that bad practices are **frequently introduced and rarely removed.**

PhD thesis: Research Gaps & Objectives #3

🚧 ***Research Gap #3***

- From RO.1, GUI-level testing is **more common** than performance testing.
- From RO.1,2 and literature review, **workload generation** emerge as major challenges in performance testing.
- Most existing workload generation approaches are **log-based**, which suffer from **late availability**, **high manual effort**, and rapid **obsolescence** as application evolve.

🎯 ***Research Objective #RO.3***

Develop a methodology to **automatically generate** Non-functional (**performance**) End-to-End Web **tests** directly from existing Functional (GUI-level) test suites.

PhD thesis: Research Gaps & Objectives #4

🚧 ***Research Gap #4***

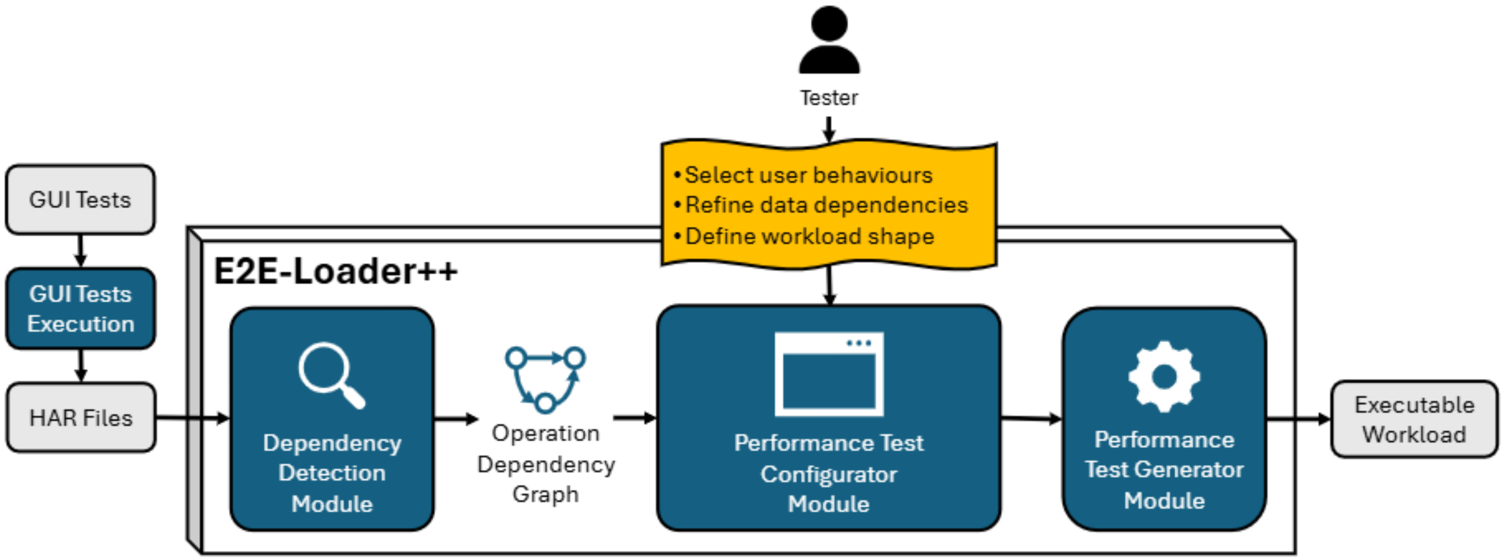
- Understanding the **real-world effectiveness and practical benefits** of the methodology and tool in industrial settings

🎯 ***Research Objective #RO.4***

Assess the **practical impact** of the proposed methodology in an **industrial environment.**

PhD thesis: Research Contribution #3

- Key idea behind our methodology is to **automatically generate** performance testing **workloads** by **leveraging existing GUI-level tests.**
- This was implemented in the **E2E-Loader++** tool.


Tester
•Select user behaviours
•Refine data dependencies
•Define workload shape
GUI Tests
GUI Tests Execution
HAR Files
E2E-Loader++
Dependency Detection Module
Operation Dependency Graph
Performance Test Configurator Module
Performance Test Generator Module
Executable Workload

PhD thesis: Research Contribution #3#4

- **2** industrial
- **1** open-source

- Pre-existing GUI-Level tests
- Expert-validated references workloads

- 6 software engineers

The industrial evaluation targets **four main aspects**:

- **Accuracy:** Assessing how precisely *E2E-Loader++* detects data dependencies.
- **Equivalence:** Evaluating whether *E2E-Loader++* generates workloads comparable to references ones.
- **Efficiency:** Measuring the reduction in **time and manual effort** achieved through its use.
- **Usability:** Understanding practitioners’ **perception and user experience** with the tool.

PhD thesis: Research Contribution #3#4

- **1 warm-up phase :** 120 min
- **2 counterbalanced session:** 4h each
- **Post-task survey**

- **Accuracy**: F1-Score 0.85, **17% improvement** over previous work baseline
- **Equivalence:** Tool-generated workloads achieve **statistical comparable** quality to reference workloads.
- **Efficiency: 62% faster** completion time and **37% fewer user interactions** (clicks, scroll etc..)
- **Practitioner perception:** SUS (System Usability Score) score 77.1 (above average usability), strong acceptance and intention to use.

Thanks for the attention!

Contact:

sergio.dimeglio@unina.it
sergio@fervento.com