# M2 Internship Position: Software Testing Applied to AI Applications

**Supervisors:** Gabriela Gonzalez, Lydie du Bousquet
**Location:** VASCO team, Laboratoire d'Informatique de Grenoble (LIG), Grenoble, France.
**Start date:** Early 2027 **Duration:** 6 months **Funding:** MIAI — EFELIA

**How to Apply:** Send your application to: gabriela.gonzalez-saez@grenoble-inp.fr Interviews will be conducted on a rolling basis until the position is filled. **The position must be filled and confirmed before 26/10.**

## Context

Software testing is the discipline responsible for ensuring the development of high-quality software systems that satisfy both functional and non-functional requirements. Depending on its level of maturity, testing can serve different purposes: from basic debugging (Level 0), to demonstrating correctness (Level 1), identifying failures (Level 2), reducing operational risk (Level 3), and ultimately contributing to improved system design and quality (Level 4).

Large Language Models (LLMs) are becoming foundational components of modern software applications, enabling systems to automate complex tasks [1]. However, while these systems are deployed and maintained as software, their evaluation is still primarily treated as a data-centric problem and is largely limited to Level 1 maturity, where performance is demonstrated through evaluation metrics such as precision, recall, or benchmark-based scores. This limitation prevents systematic support for both the validation of expected properties and the validation of system results in realistic usage scenarios.

Achieving higher levels of testing maturity for LLM-based systems requires not only detecting failures, but also determining whether the test suite sufficiently covers the relevant behaviours of the application. Progressing toward Levels 2 and 3 therefore requires moving beyond aggregate performance metrics toward structured test suites with explicit coverage criteria, capable of revealing untested requirements, interactions, execution paths, and regions of model behaviour.

## Objective

This M2 project focuses on the systematic construction of test suites for LLM applications. The project will adapt classical software-testing strategies, to generate and extend structured test suites. Their adequacy will be assessed using two notions of coverage: **specification-based coverage**, targeting requirements, policies, and execution paths; and **model-representation coverage**, based on input-output attributions, neural representations, and output probabilities using Explainable AI methods.

Our experimental environment is a tool-augmented generation task based on T-bench [2]. T-bench models realistic interactions among users, language agents, domain-specific API tools, and policy constraints, making it suitable for studying AI applications as complete software systems.

The objective is to determine whether these coverage criteria can identify missing or redundant tests and provide a better characterization of test-suite quality for AI-based software components. The expected outcome is a methodology and prototype framework for the systematic construction and evaluation of test suites for tool-augmented LLM applications.

## Tasks

- Run open-source LLMs within a tool-augmented generation framework.
- Implement XAI methods to inspect the internal behaviour of the model when executing each test case.
- Define specification-based and model-representation coverage criteria.
- Evaluate the proposed approach using the T-Bench benchmark.

## Supervision and Working Environment

For the timely development of the internship, supervisors will provide access to open-source LLMs, including example Jupyter notebooks running in the GRICAD or LIG environment. The internship will run alongside M1 and M2 internships on related topics, with opportunities for regular discussion and collaboration.

## References

[1] Weber, I. (2024). Large language models as software components: A taxonomy for llm-integrated applications. arXiv preprint arXiv:2406.10300.
[2] Yao, S., Shinn, N., Razavi, P., & Narasimhan, K. (2024). T-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv preprint arXiv:2406.12045.