LLMs IN SOFTWARE ENGINEERING: A FOCUS ON ISSUE REPORT CLASSIFICATION AND USER ACCEPTANCE TEST GENERATION

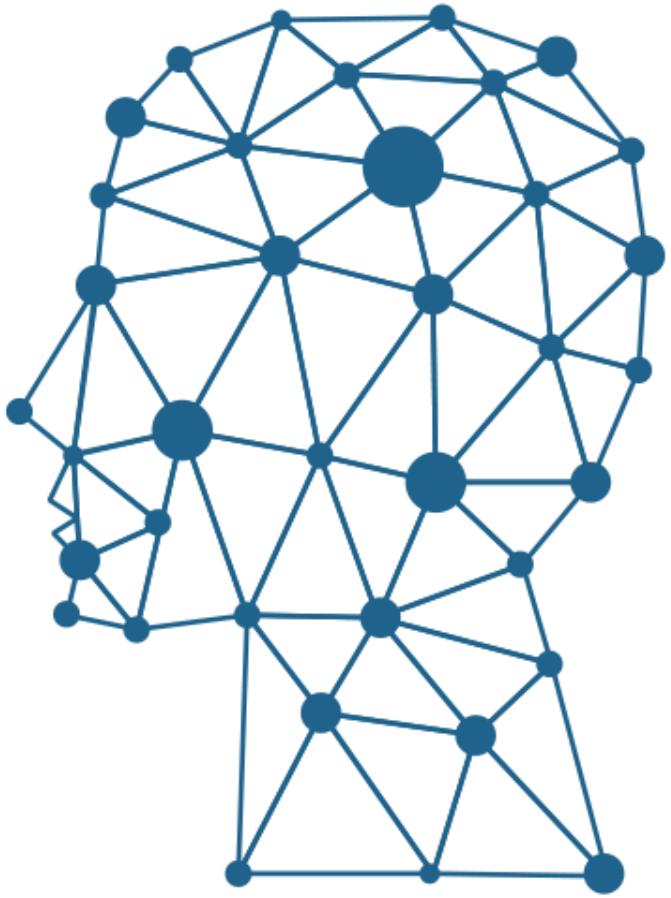
Gabriele De Vito[1], **Luigi L. L. Starace**[2], Sergio Di Martino[2], Filomena Ferrucci[1] and Fabio Palomba[1]

[1] Università degli Studi di Salerno
[2] Università degli Studi di Napoli Federico II

May 29, 2024 – Naples, Italy

LLMs AND SOFTWARE ENGINEERING

- LLMs have demonstrated remarkable performance in **understanding** and **generating Natural Language** and **Source Code**
- In the **Software Engineering** domain, LLMs have the potential for **revolutionizing** traditional practices and improve software development processes [1]



[1] Ozkaya, Ipek. "Application of large language models to software engineering tasks: Opportunities, risks, and implications." *IEEE Software* 40.3 (2023): 4-8.

INVESTIGATED TASKS

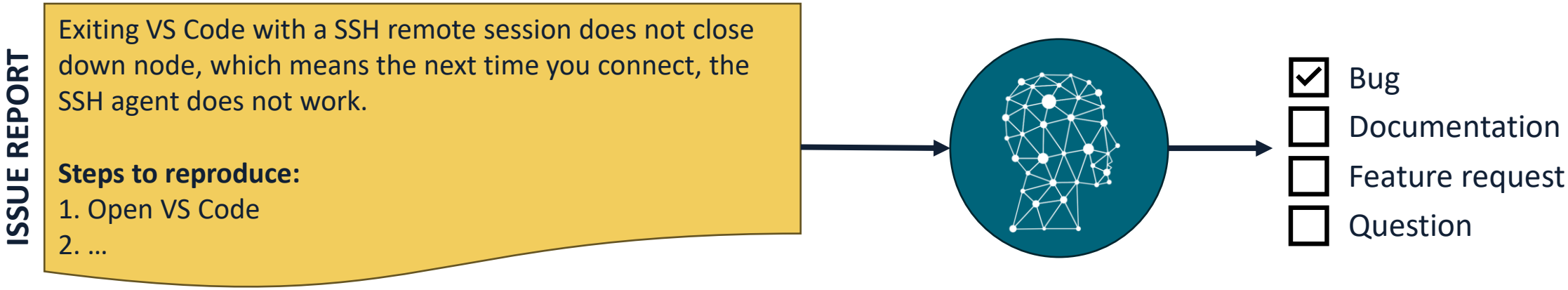
We are currently focusing on two key tasks in Software Engineering:

Issue Report Classification

User Acceptance Test Generation

ISSUE REPORT CLASSIFICATION

- Issue Reports play a crucial role in identifying, documenting and tracking issues and change requests during the Software Lifecycle
- Labelling incoming issue reports (correctly) is fundamental
  - Stakeholders who submit issue reports often misclassify their own issues!
  - Misclassification can lead to misallocation of resources, delays and inefficiencies
  - Automated and effective classification can mitigate these challenges

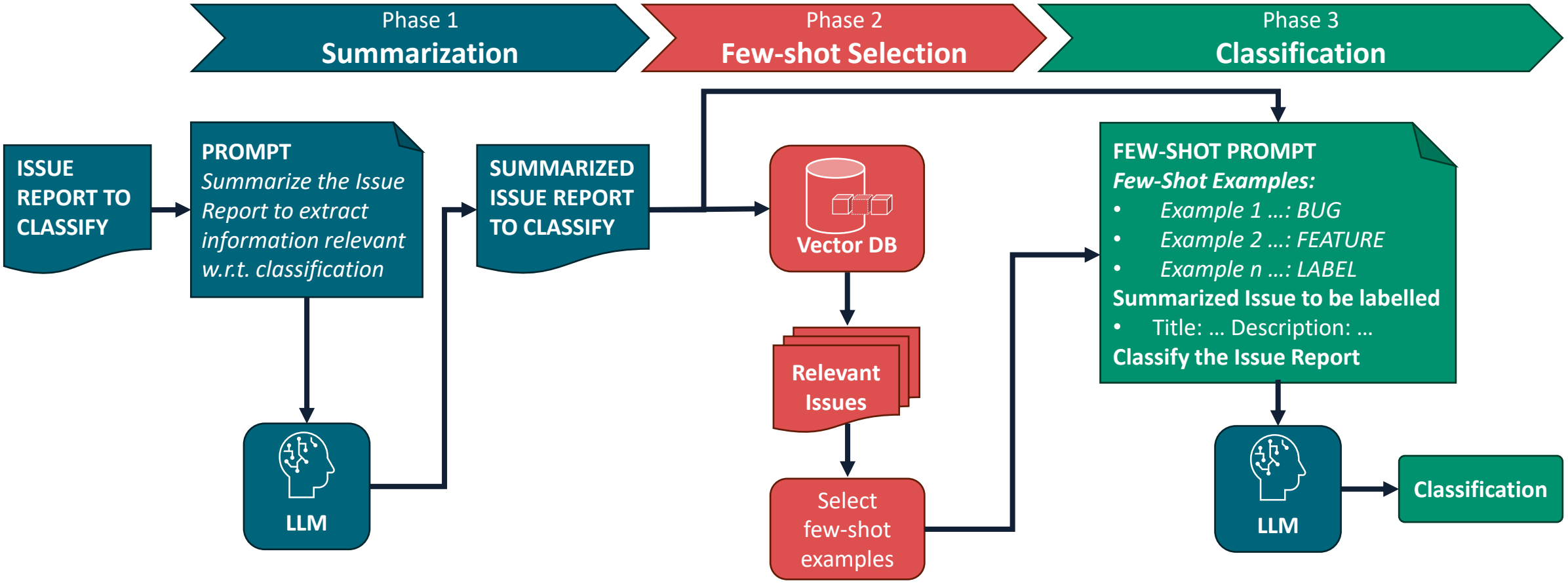


ISSUE REPORT CLASSIFICATION: STATE OF THE ART

- Many machine-learning based approaches have been proposed in the literature (e.g.: [1])
- Recently, Colavito et al. [2] proposed an approach leveraging GPT-like LLMs to automate issue labelling.

[1] R. Kallis, et al., Ticket tagger: Machine learning driven issue classification, in: Proc. of the IEEE Int. Conf. on Software Maintenance and Evolution (ICSME), IEEE, 2019, pp. 406–409.
[2] G. Colavito, F. Lanubile, N. Novielli, L. Quaranta, Leveraging GPT-like LLMs to automate issue labelling. To appear in the Mining Software Repositories (MSR) proceedings (2024).

ISSUE REPORT CLASSIFICATION: APPROACH



ISSUE REPORT CLASSIFICATION: ASSESSMENT

We plan to assess the effectiveness of the approach leveraging datasets of real-world issue reports

- A well-known dataset is the NLBSE dataset [1] (1.4M issues)
- A subset of that dataset (400 issues) was also manually verified by Colavito et al. [2]

[1] R. Kallis, et al., The NLBSE'23 tool competition, in: Proceedings of The 2nd Intern. Workshop on Natural Language-based Software Engineering (NLBSE'23), 2023
[2] G. Colavito, F. Lanubile, N. Novielli, L. Quaranta, Leveraging GPT-like LLMs to automate issue labelling. To appear in the Mining Software Repositories (MSR) proceedings (2024).

USER ACCEPTANCE TEST (UAT) GENERATION

- UATs are essential to ensure that software systems meet the specified requirements and work as intended for end users
- Traditionally, creating UATs involves translating user requirements into **testable, formalized scenarios**
  - Time-consuming; requires domain knowledge; error-prone.

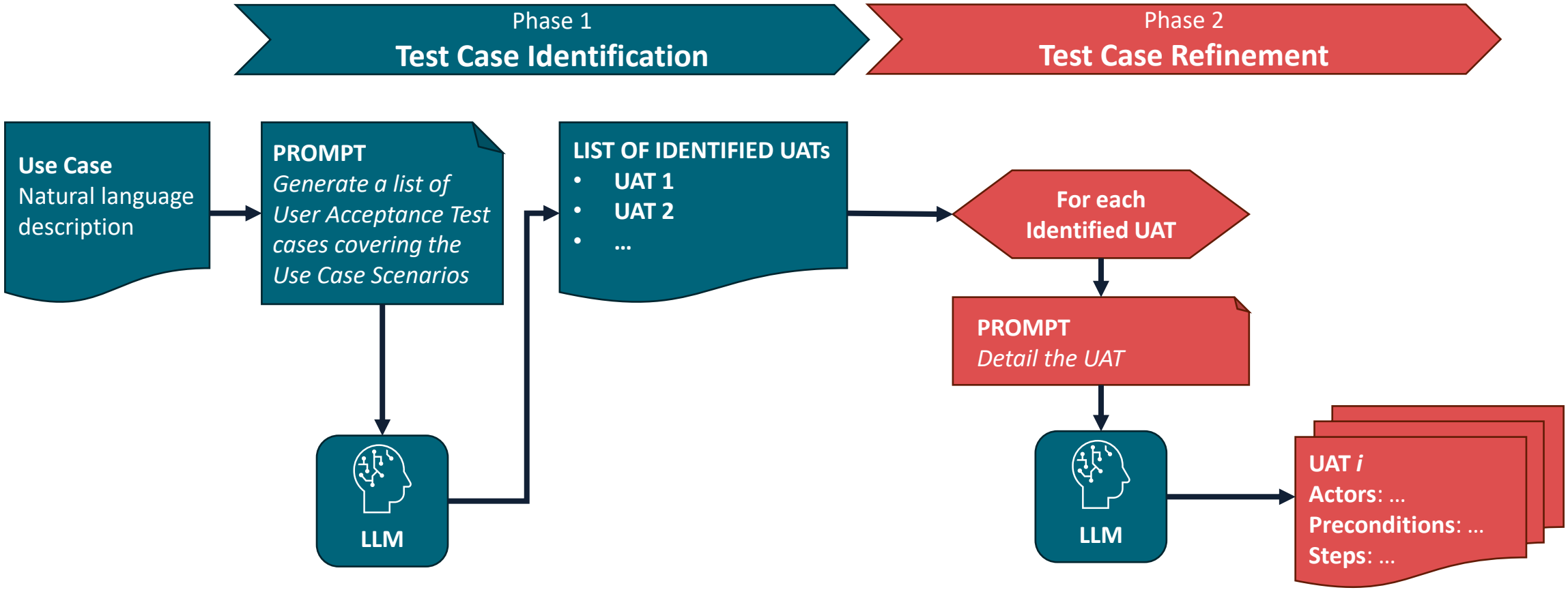
UAT GENERATION: STATE OF THE ART

Existing approaches (e.g.: [2, 3]) exhibit some limitations:

- **Limited scalability** on complex systems
- **Limited generalizability** to different domains or types of software systems

[2] J. Fischbach, et al., Specmate: Automated creation of test cases from acceptance criteria, in: IEEE 13th Int. Conf. on Software Testing, Validation and Verification (ICST), 2020, pp. 321–331.
[3] C. Wang, et al., Automatic generation of acceptance test cases from use case specifications: An NLP-based approach, IEEE Transactions on Software Engineering 48 (2022) 585–616.

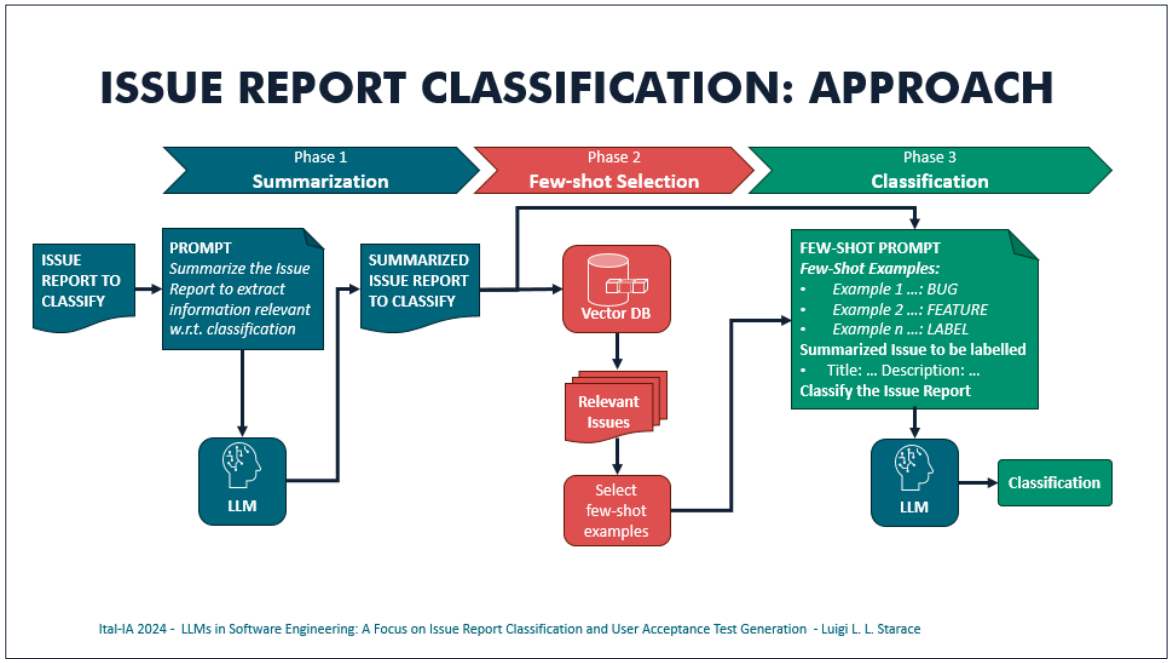
LLM–BASED UAT GENERATION: APPROACH

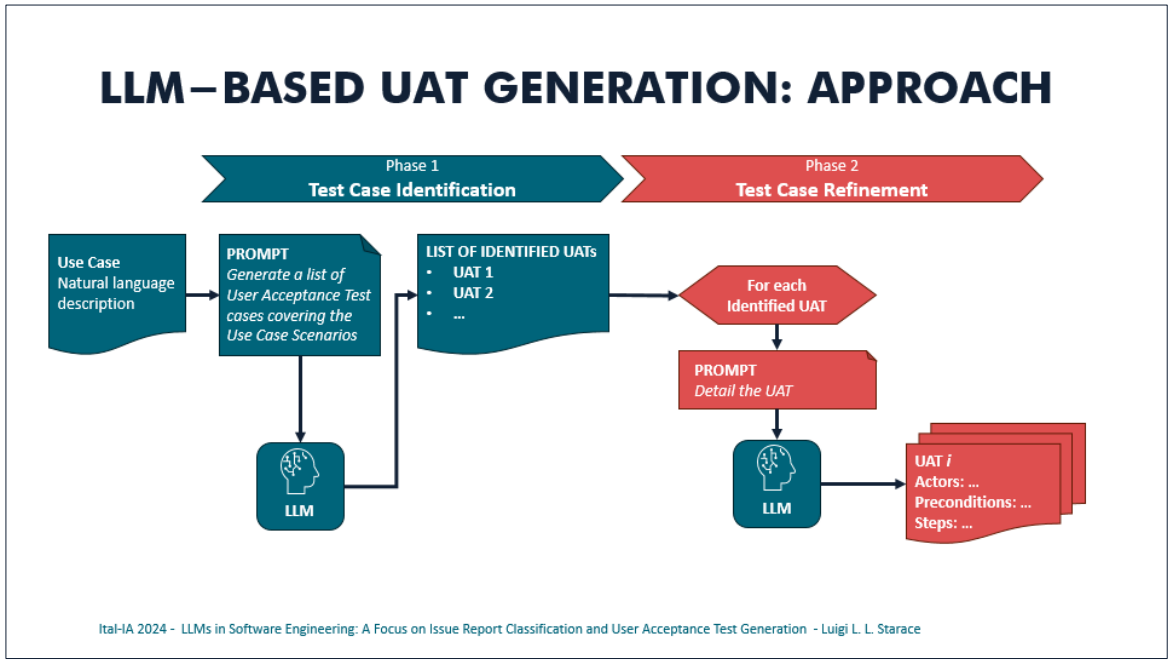


LLM–BASED UAT GENERATION: ASSESSMENT

- We plan to conduct and empirical evaluation involving Software Engineering Professionals
- We will compare UAT specifications defined by Professionals with those generated automatically by our approach, considering:
  - Completeness
  - Correctness
  - Clarity and Understandability

LLMs IN SOFTWARE ENGINEERING: A FOCUS ON ISSUE REPORT CLASSIFICATION AND USER ACCEPTANCE TEST GENERATION





Luigi Libero Lucio Starace
luigiliberolucio.starace@unina.it
https://luistar.github.io