**Software Testing of Generative AI Systems: Challenges and Opportunities**

**Aldeida Aleti, Faculty of Information Technology, Monash University Melbourne, Australia**

**Aim of the Research Study**

This paper by explores the emerging challenges and opportunities in testing generative AI systems, models that are capable of producing creative outputs such as text, images or code. Unlike traditional software, these systems produce variable and most of the time non-deterministic outputs, making it difficult to define expected results or correctness. Aldeida Aleti aims to identify where conventional software testing falls short and to propose directions for building new testing methodologies suited for generative AI.

**Methodology of the Study**

The study is an analytical review and it shows an overview of existing software testing approaches and compares them to the unique characteristics of generative AI systems. The paper outlines a variety of challenges such as lack of test oracles, instability of outputs and ethical considerations. It also reviews how current practices in AI testing, evaluation and verification attempt to address them.

**Results of the Study**

The analysis reveals that current testing techniques are largely inadequate for generative AI. Key challenges include defining what a correct output looks like, ensuring consistency across diverse inputs and validating performance across ethical and fairness dimensions. The study identifies promising directions, such as leveraging large-scale benchmarks, incorporating user feedback, and integrating explainability to support testing.

**Implications for Research and Practice**

For research, the paper calls for the development of new testing methodologies tailored to generative AI, including formal definitions of quality, correctness, and diversity. It also suggests interdisciplinary collaboration between software engineering, AI, and human-computer interaction. For practitioners, the study recommends incorporating evaluation techniques like user-centered testing, adversarial input generation, and fairness audits into development pipelines. These approaches can help bridge the gap between system behavior and human expectations in real-world use.