Large language models (LLMs) have shocked the world ever since ChatGPT was released publicly in Fall 2022. They are well-known for their several beneficial abilities such as answering questions, providing grammatical corrections, and even programming assistance. However, are these LLMs programming abilities truly good enough to replace the labor of a human software engineer in the generation of software unit tests? Faults in software unit tests can lead to undetected bugs, causing a decrease in software quality, reliability, and security.

This paper presents an evaluation framework to assess the quality of software unit test generated by LLMs considering 3 factors such as validity, effectiveness, and readability. For each factor, I propose using corresponding metrics. The evaluation framework includes build \& success rate, evaluating unit test failures, line coverage, mutation score, readability of code comments, and readability of the code. Build \& success rate identifies if the unit tests generated are runnable. The evaluation of unit test failures identifies whether the input program had a bug or if the LLM had created a faulty unit test. Line coverage identifies how much of the input program code is being tested. Mutation score identifies how well the unit tests detected mutants or bugs in the input program. The readability of code comments identifies how well the documentation associated with the unit tests can be understood. Readability of the code identifies how easy is the code to understand and maintain. This research illustrates issues and benefits in software unit tests generated by LLMs and proposes future research directions.