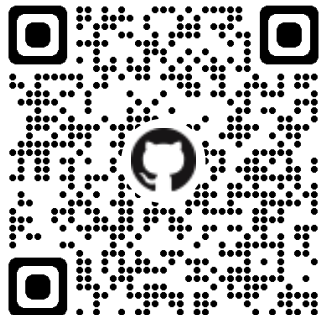


pytest-inline: An Inline Testing Tool for Python

Yu Liu¹, Zachary Thurston², Alan Han², **Pengyu Nie**¹, Milos Gligoric¹, Owolabi Legunsen²



ICSE 2023



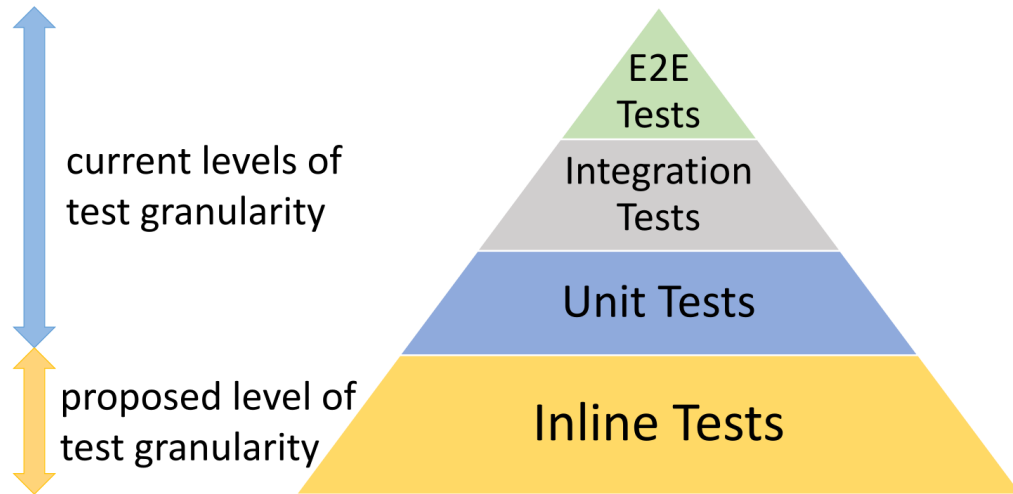
[pytest-dev/pytest-inline](https://github.com/pytest-dev/pytest-inline)

partially
supported by

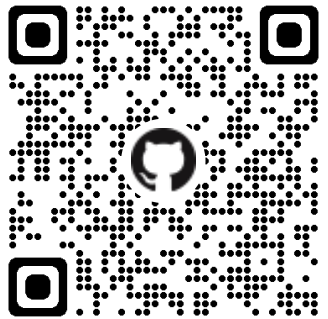


CCF-1652517
CCF-2019277
CCF-2045596
CCF-2107291
CCF-2217696

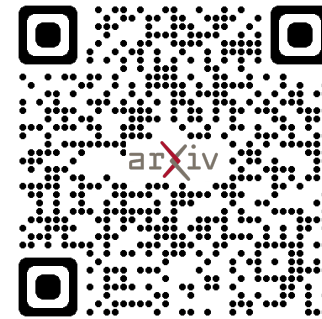
Inline Testing



- **Inline testing:** new granularity of testing for checking individual program statements
 - regular expression
 - string manipulation
 - bit manipulation
 - collection handling (list comprehension etc.)



[pytest-dev/pytest-inline](https://github.com/pytest-dev/pytest-inline)



Liu et al., Inline Tests. In ASE 2022.

Inline Testing Example

```
def get_assignment_map_from_checkpoint(tvars, init_c):  
    ...  
    for var in tvars:  
        name = var.name  
        m = re.match("^(.*):\\d+$", name)  
        itest().given(name, "a:0").check_eq(m.group(1), "a")  
    ...  
    declare          assign          assert
```

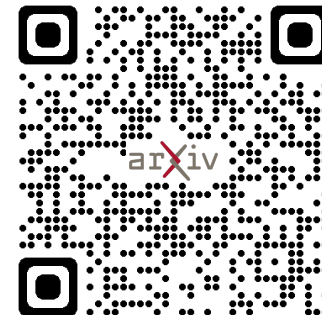
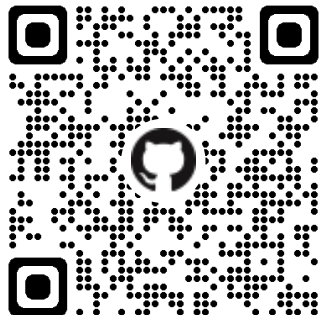
statement under test

inline test

→ m = re.match("^(.*):\\d+\$", name)
→ itest().given(name, "a:0").check_eq(m.group(1), "a")

Evaluation on 87 inline tests in 31 Python projects

- **fast**: <0.1s/test
- **low overhead**: 0.012
- **no overhead when disabled**



Integration with pytest

- `pytest-inline`: a pytest plugin that simplifies running inline tests
- `$ pip install pytest-inline`

Feature	Examples
name	<code>itest(test_name="check_match_name")</code> <code>itest("check_match_name")</code>
parameterized test	<code>itest(parameterized=True).given(name, ["a:0", "a:1:1"])</code> <code>.check_eq(m.group(1), ["a", "a:1"])</code>
repetition	<code>itest(repeated=2)</code>
tag	<code>itest(tag=["regex"])</code> <code>\$ pytest --inlinetest-group="regex"</code>
disabled test	<code>itest(disabled=True)</code>



Integration with `pytest` – New Features

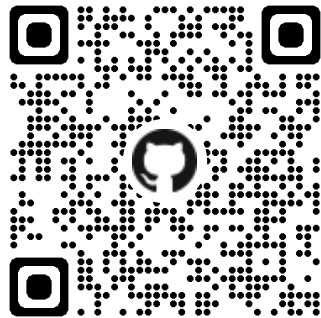
- 5 features added in `pytest-inline` compared to our ASE'22 prototype

Feature	Examples
timeout	<code>itest(timeout=5)</code>
assumption	<code>itest().assume(platform.system() == "Linux")</code>
execution order	<code>\$ pytest --inlinetest-order="first" --inlinetest-order="second"</code>
parallel execution	<code>\$ pip install pytest-xdist</code> <code>\$ pytest -n auto</code>
more oracles	<code>check_neq</code> , <code>check_none</code> , <code>check_not_none</code> , <code>check_same</code> , <code>check_not_same</code>



Conclusions

- **pytest-inline**: a pytest plugin for running inline tests in Python
- Fast, low overhead, easy to write
- Open-sourced and hosted by pytest-dev organization



<https://github.com/pytest-dev/pytest-inline>

```
$ pip install pytest-inline
```

Yu Liu, Zachary Thurston, Alan Han, **Pengyu Nie**, Milos Gligoric, Owolabi Legunsen

pynie@utexas.edu

joining Waterloo CS as assistant professor starting from 9/1