STUNIR Testing Guide

Test Suites Overview

Suite	Purpose	When to Run	Target Time
Smoke	Quick sanity checks	Every commit	< 30s
Regression	Prevent bug recurrence	Every commit	< 2min
Contract	API/format compatibility	Every commit	< 1min
Visual	Output format consistency	PRs	< 1min
Fuzz	Find edge cases	Scheduled/Manual	< 30min
Mutation	Test quality	Weekly	~1hr
Load	Performance	Manual	~5min
Chaos	Failure handling	Manual	~5min

Running Tests

Quick Start

```
# Run smoke tests (fast sanity check)
pytest tests/smoke/ -v

# Run all standard tests
pytest tests/ -v --ignore=tests/fuzz --ignore=tests/load --ignore=tests/chaos

# Run with coverage
pytest tests/ -v --cov=tools --cov=manifests
```

By Suite

```
# Smoke tests (< 30s)
pytest tests/smoke/ -v --timeout=30

# Regression tests
pytest tests/regression/ -v

# Contract tests
pytest tests/contracts/ -v

# Visual regression tests
pytest tests/visual/ -v

# Fuzz tests (requires hypothesis)
pytest tests/fuzz/ -v

# Chaos tests
pytest tests/chaos/ -v

# Load tests (requires locust)
locust -f tests/load/locustfile.py --headless -u 10 -t 60s

# Mutation tests (requires mutmut)
python tests/mutation/run_mutation_tests.py
```

Installing Testing Dependencies

```
# Core testing
pip install -e .[dev]

# Additional testing tools
pip install -e .[testing]

# Or install individually
pip install hypothesis mutmut locust pytest-snapshot
```

Test Quality Thresholds

Coverage

- Overall: > 80%
- Critical modules: > 90%

Mutation Score

- tools/ir_emitter: > 80%
- tools/security: > 85%
- manifests/: > 75%

Performance (p95)

- compute_sha256: < 10ms
- canonical_json: < 50ms
- emit_ir: < 200ms

Writing New Tests

Smoke Test Template

```python
import pytest

class TestFeatureSmoke:
    @pytest.mark.timeout(5)
    def test_basic_functionality(self):
        # Quick check that core feature works
        pass
```

Regression Test Template

```python
class TestRegBug123:
    \"\"\"
    Regression test for Bug #123.

    Bug: Description of the bug
    Fix: What was done to fix it
    \"\"\"
    def test_bug_123_fixed(self):
        # Test that verifies the bug is fixed
        pass
```

Fuzz Test Template

```python
from hypothesis import given, strategies as st

@given(st.text())
def test_handles_any_input(data):
    # Function should not crash
    result = function_under_test(data)
    assert result is not None or result is None
```

CI Integration

Workflow	Trigger	Blocks PR
smoke-tests.yml	Push, PR	Yes
regression-tests.yml	Push, PR	Yes
contract-tests.yml	Push, PR	Yes
fuzz-testing.yml	Schedule	No
mutation-testing.yml	Schedule	No