

F.I.R.S.T. principles

- Principles, not rules, that we may follow when writing tests:
 - Fast
 - Independent
 - Repeatable
 - Self-validating
 - Thorough

Fast:

- Unit tests should be fast
 - Faster tests – faster feedback

Isolated/Independent:

- Tests should be isolated from:
 - Other tests
 - External environment
 - No shared state with other tests
 - The order in which tests run should not matter
 - Contradiction with the F(fast) principle:
 - Individual tests take more time to setup

Repeatable:

- Same result with the same input:
 - Challenge: Random/Date values – we will often mock these
- Example: test that writes to a database:
 - It should always clean up
- In contradiction with the Fast principle:
 - More setup and teardown operations

Self validating:

- After the test is finished, it's results should be clear:
 - Pass/fail

Thorough:

- Cover all the cases/paths/scenarios
 - Hard to think at all of them from the beginning
- Happy cases, bad paths, edge cases
- Invalid output
- Large values
- 100% code coverage – not a great indicator

Illustration:

- A tester walks into a bar
 - Runs into a bar
 - Crawls into a bar
 - Dances into a bar
 - Jumps into a bar
 - And orders:
 - A beer
 - 2 beers
 - 0 beers
 - 1 million beers
 - -1 beers
 - A lizard beer
- Testing complete 😊

Illustration:

- A real customer walks into a bar
 - Asks where the bathroom is
- System crash