

Looking out for Test-driven Development Gotchas



Jason Olson

Staff Software Engineer

@jolson88 www.jolson88.com

Overview



Testing anti-patterns

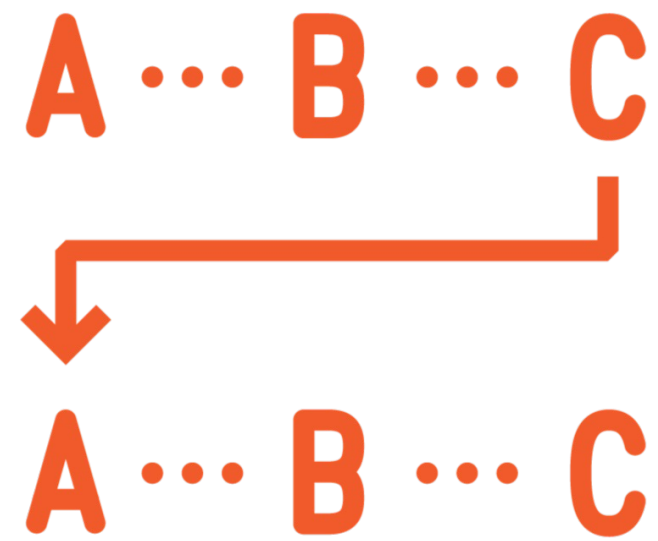
Test-driven development limitations

Common questions



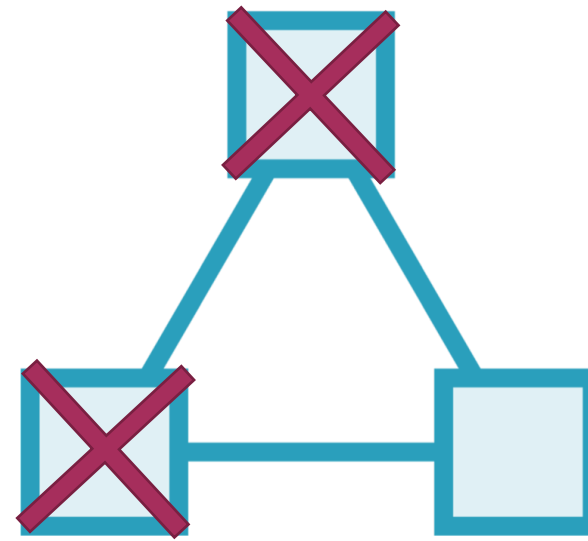
Testing Anti-patterns

Anti-pattern: Dependencies between Tests



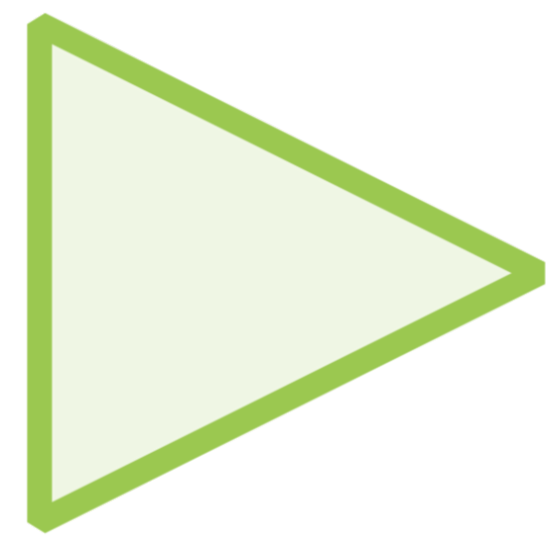
Test Ordering

Order of tests should not matter



Cascading Failures

Cascading failures can be difficult to track down



Execution

Tests should work regardless of serial vs. parallel execution

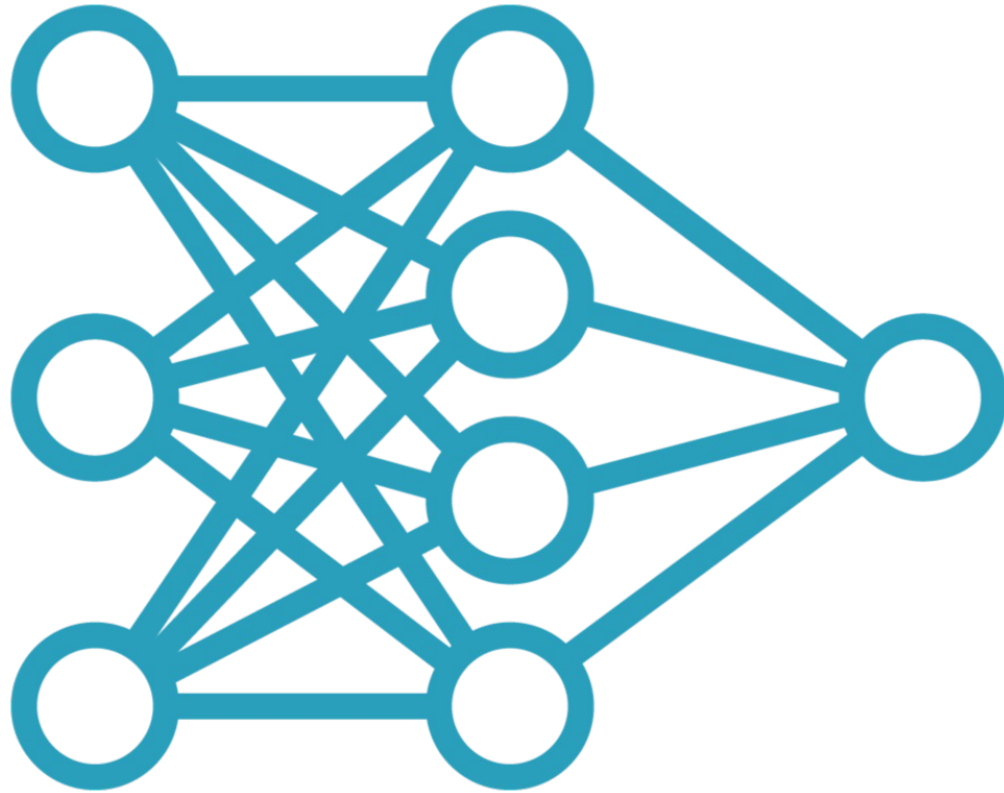
Anti-pattern: Testing Implementation Details

Focus on the “what”, not the “how”

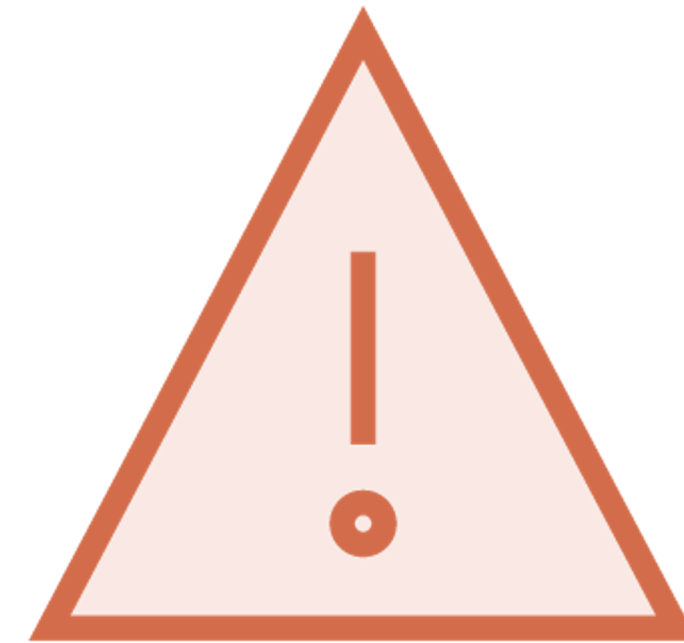
**Testing the “how” leads to brittle code when
refactoring**



Anti-pattern: Long-running Tests

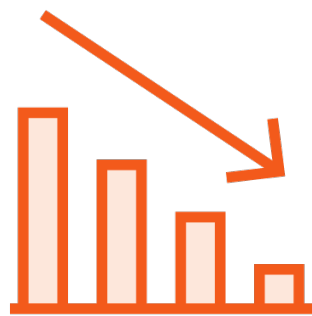


**Warning sign that code might be
too coupled**

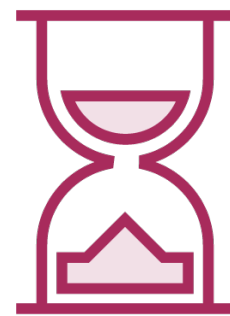


**Warning sign that code might
not be very testable**

Anti-pattern: Long-running Tests



**Lower
Developer
Confidence**



**Slower
Release
Cycles**



**Less
Agile**



Limitations of Test-driven Development



Possible holes in tests

“What does every bug in production have in common?”

- Joe Armstrong, Erlang Co-Creator

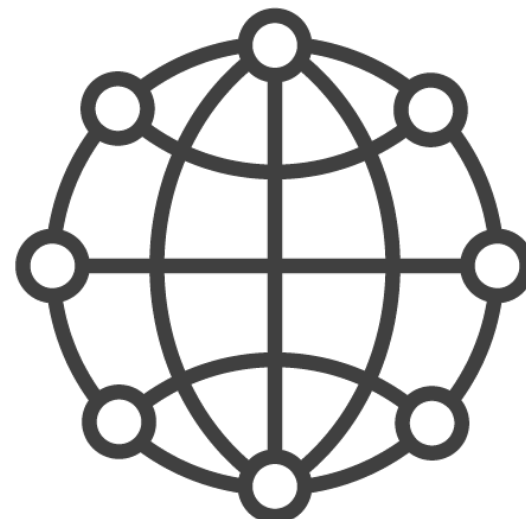




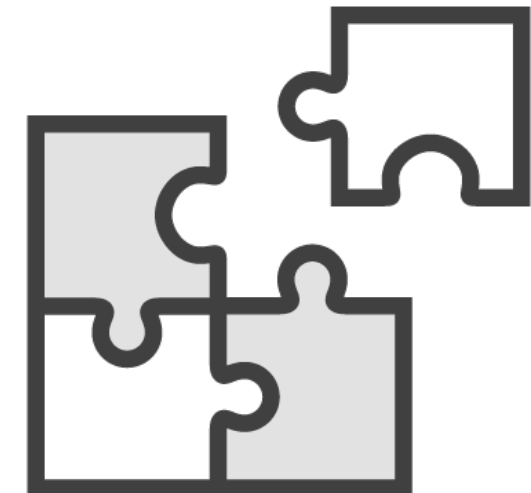
**TDD is not sufficient
by itself**



**Deployment
Verification**



**Network
Changes**



**Integration
Testing**





**Management support
is vital**



Common Questions

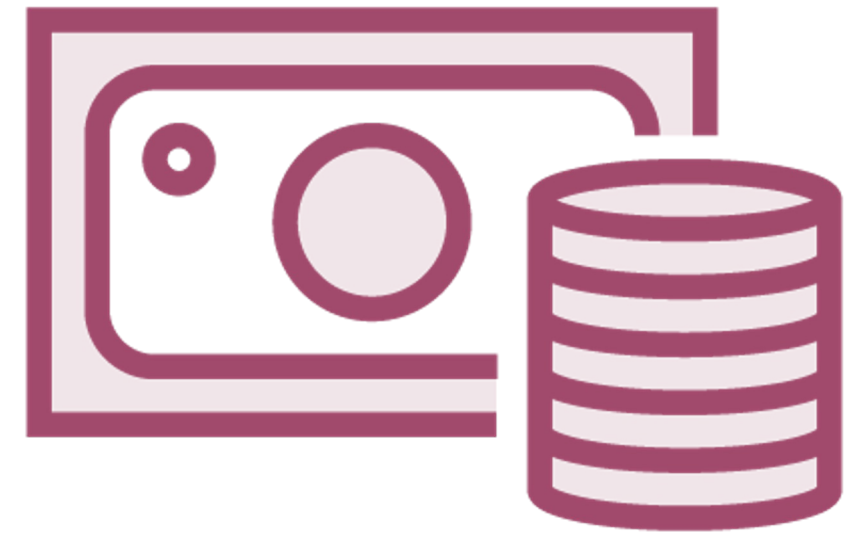
Is Agile Required for TDD?



**Requirements
verification**



**Regression
catching**



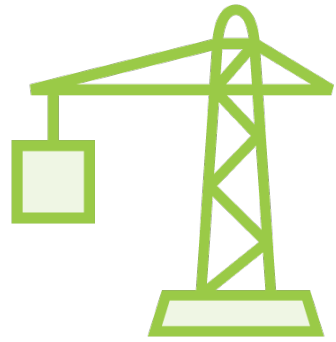
**Lower
maintenance
costss**



Test-driven Development **Shines** within an Agile Process



Do I Need to Write Tests First?



Avoid over-engineering



Momentum



Confidence

Not Always...



Test-driven Development
Is About Delivering **Value**



Summary



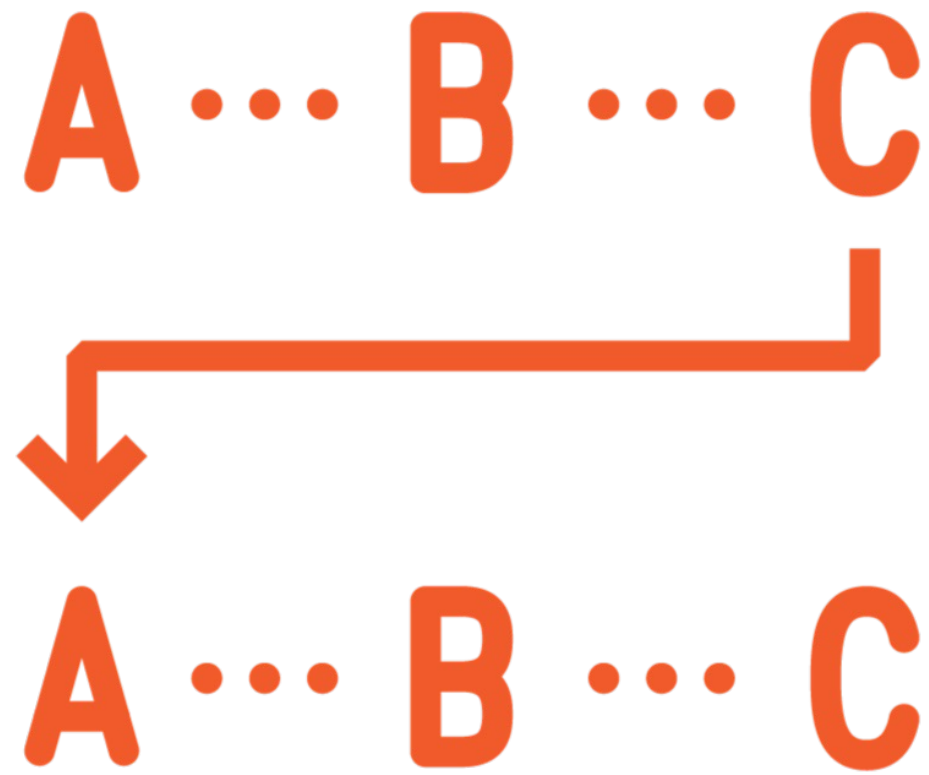
Testing anti-patterns

Test-driven development limitations

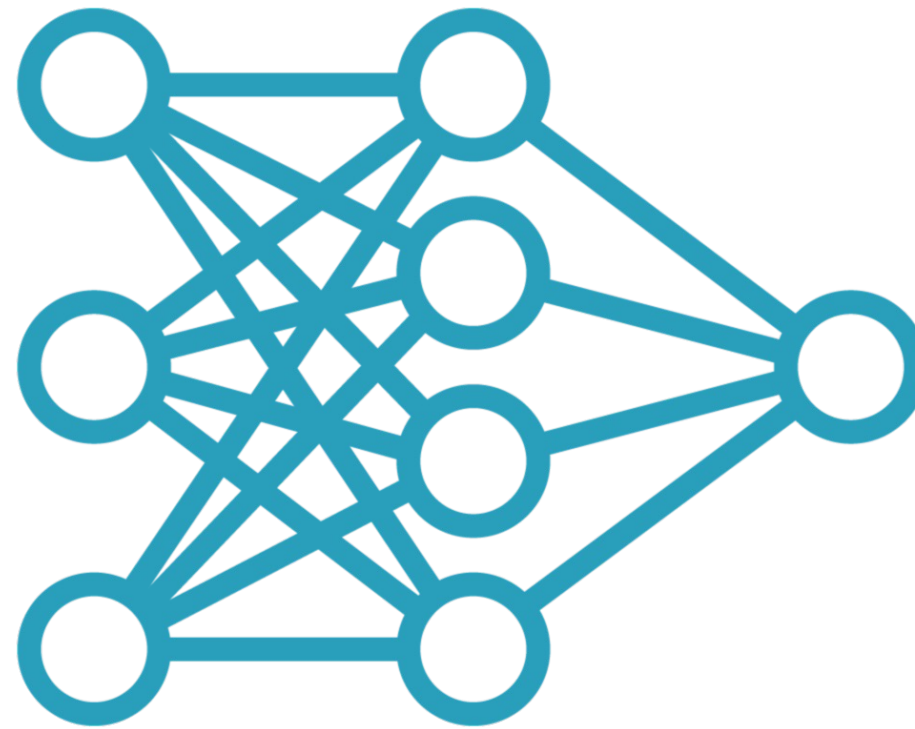
Common questions



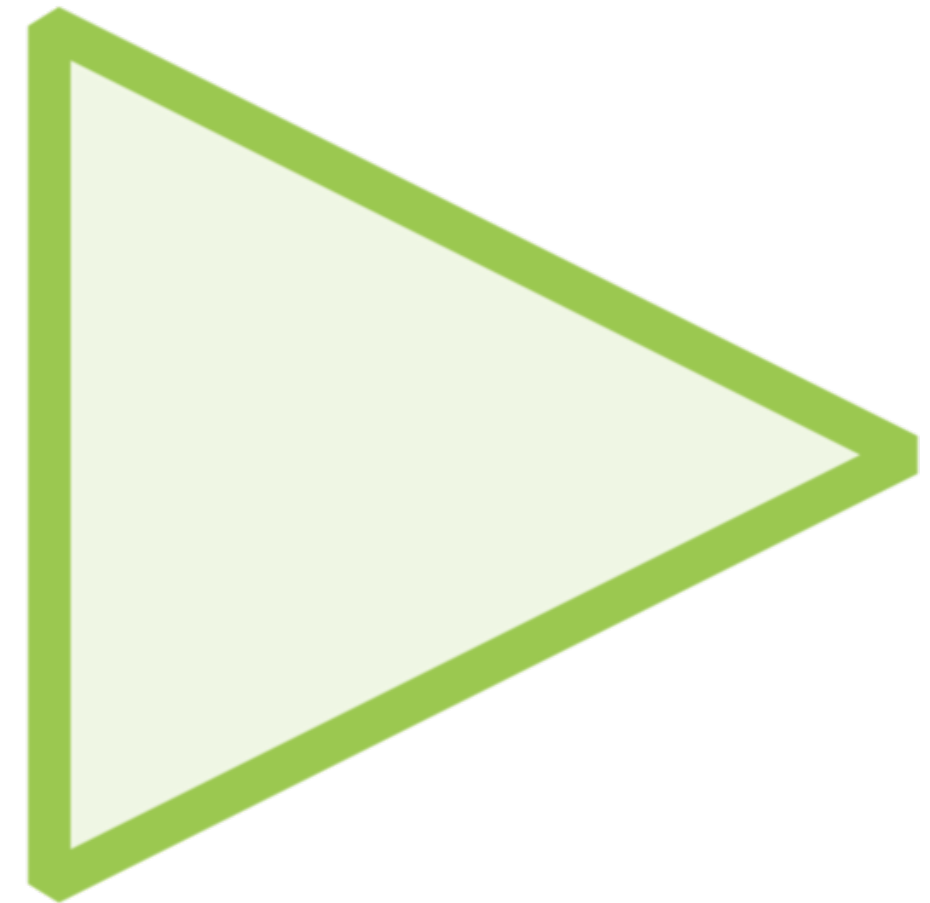
Testing Anti-patterns



**Dependencies
between tests**



**Testing
implementation
details**



Long-running tests

Test-driven Development Limitations



Possible holes in tests

Test-driven development won't solve all your problems by itself...



TDD is not sufficient by itself



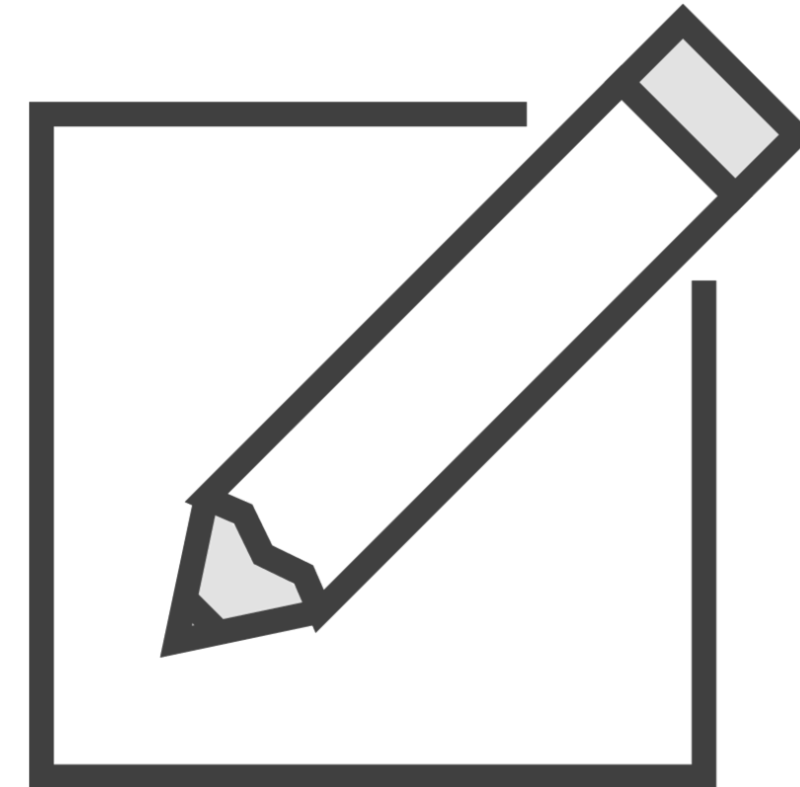
Management support is vital



Common Questions



Is Agile required to practice TDD?



Do I need to write my tests first?

Test-driven Development
Is About Delivering **Value**

