Looking out for Test-Driven Development Gotchas

Course Outline

Software development challenges

What is test-driven development?

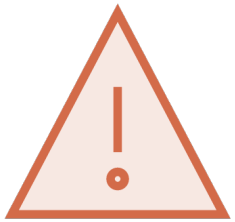
Different ways of testing applications

Test-driven development in action

Strategies/techniques for testing code

Test-driven development gotchas

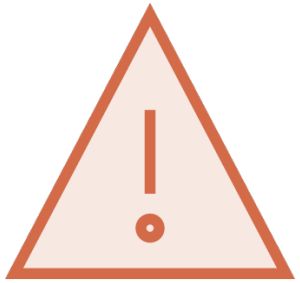
Test-Driven Development Gotchas



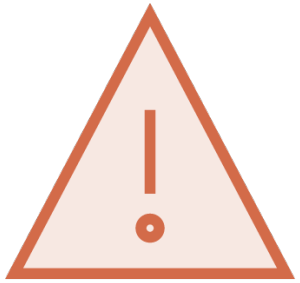
Anti Patterns

TDD Limitations

Common Questions



Testing Anti-patterns

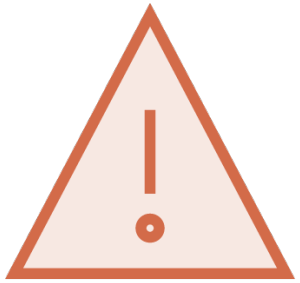


Dependencies Between Tests

Execution order of tests should not matter

Interdependent tests cause cascading failures and false positives

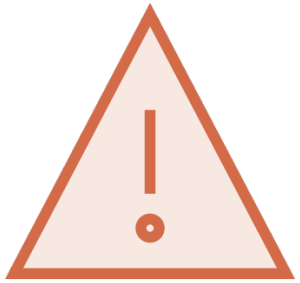
Serial execution versus parallel execution



Testing Implementation Details

Tests should focus on the “what” not the “how”

Testing implementation details leads to brittle tests that break when refactoring



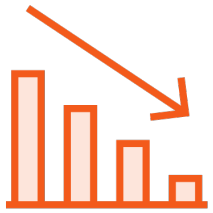
Slow-running Tests

Prevents rapid red/green/refactor cycles

Warning-sign that code might be too coupled

Warning-sign that code might not be very testable

Long-running Tests



Lower Developer Confidence

Slower Release Cycles

Less Agile

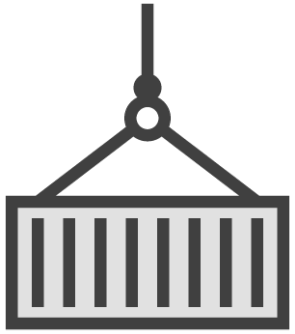
Limitations of Test-Driven Development

Possible holes in tests

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

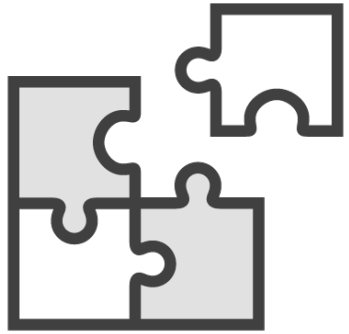
- Joe Armstrong, Erlang Co-Creator

TDD is not sufficient by itself



Deployment Verification

Network Changes



Integration Testing

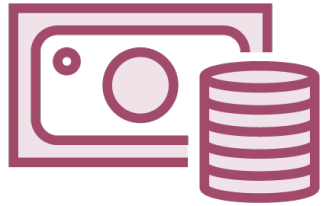
Management support is **vital**

Common Questions

Is Agile Required for TDD?

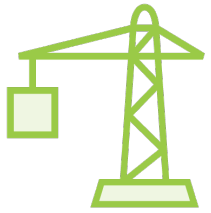
Requirements Verification

Regression Catching



Lower Maintenance Costs

Do I Need to Write Tests First?



Avoid Over-Engineering

Momentum

Confidence

Test-Driven Development Is About Value

Recap - What Is Test-Driven Development?

Test

"A procedure intended to establish the quality, performance, or reliability of something, especially before it is taken into widespread use."

Satisfies Requirements

Responds Correctly to all Input

Acceptable Performance

Test-Driven Development

"A **software development process** that relies on the repetition of a **very short development cycle**: requirements are turned into very specific test cases, then the software is improved to pass the new tests, only." - Wikipedia

"Red – Green - Refactor"

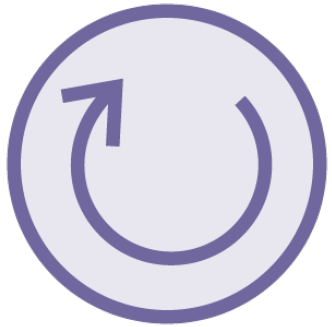
Maintenance accounts for **65%** of all software development costs!

Customer Focus

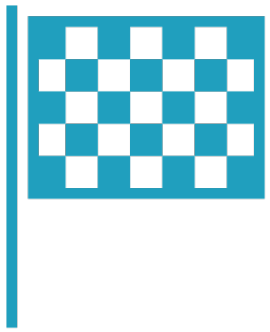
Avoid Over-Engineering

Confidence And Momentum

Next Steps



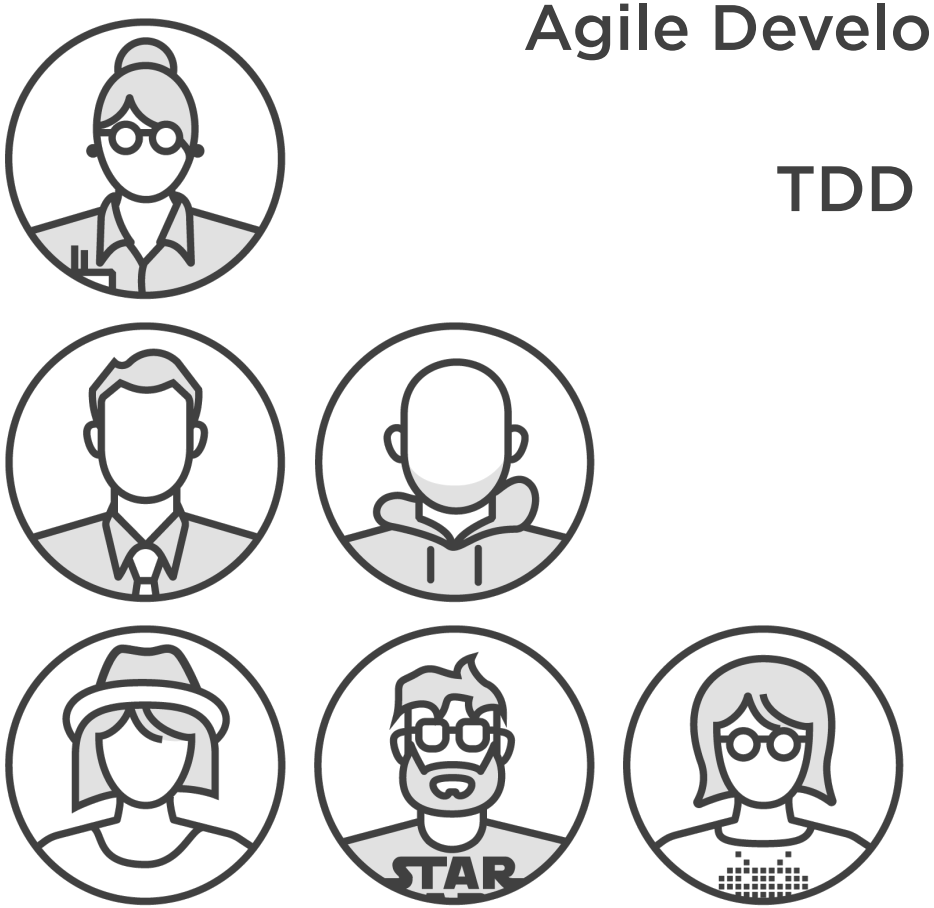
Continuous Improvement



From Legacy to Supported

Share Your Journey

Related Topics



Agile Development

TDD for Your Language

Refactoring

Continuous Integration

Continuous Deployment

Summary

Software development challenges

What is test-driven development?

Different ways of testing applications

Test-driven development in action

Strategies/techniques for testing code

Test-driven development gotchas