Creating High Value Tests That Cover Your Riskiest Code

Rusty Divine
SOFTWARE CONSULTANT

@CornerPosts cornerpostsoftware.com

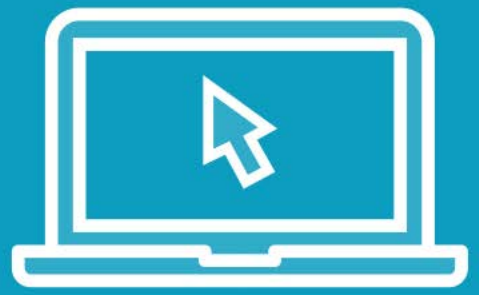
Core Concepts

Write testable code

Test riskiest code

Know when to stop

Demo



Code review of poorly tested code

Tips for Testable Code

Elements of Testable Code

Ask for what you need

Avoid “new”

Avoid mutable global state

Beware static methods

Add seams

Composition over inheritance

SOLID design

Single responsibility

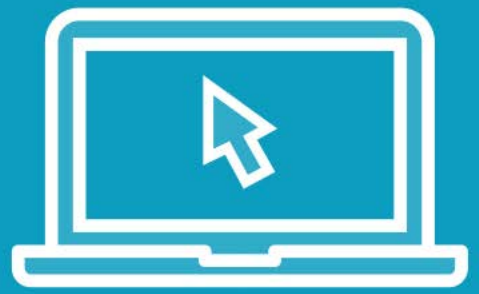
Open for extension, closed for modification

Liskov substitution principle

Interface segregation

Dependency inversion

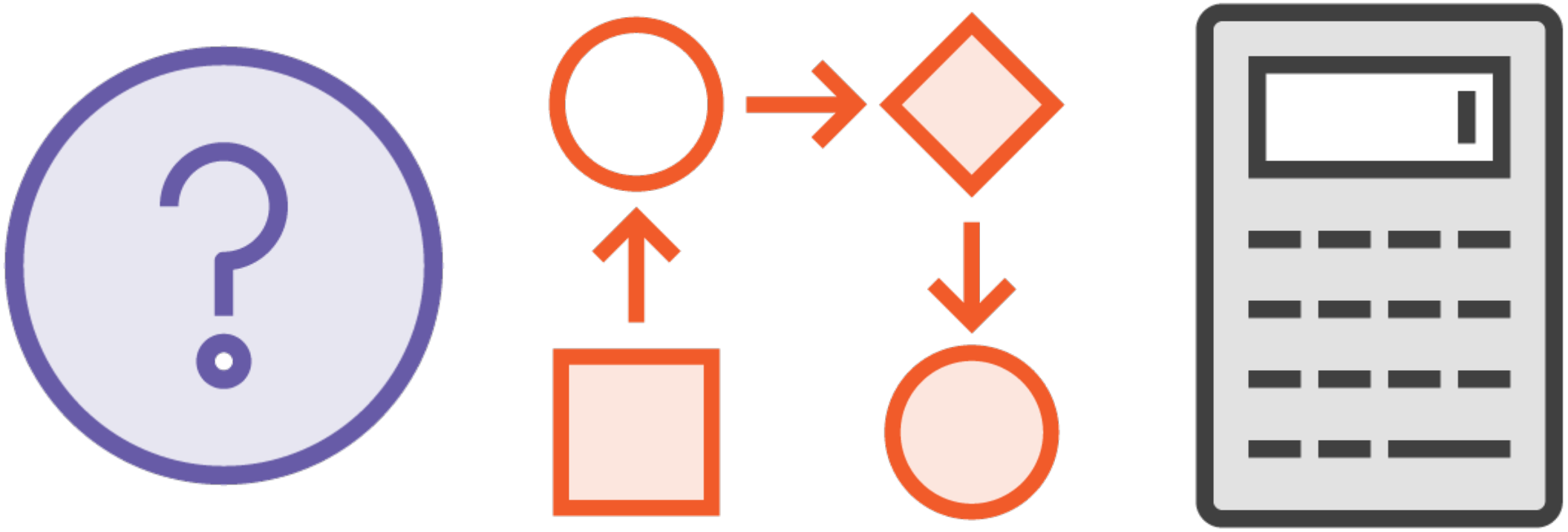
Demo



SOLID testable code

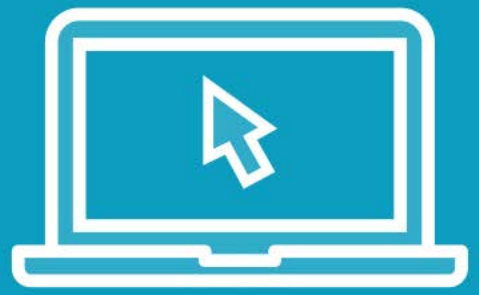
Focus on High Value Tests

Risk-driven Testing



Value by Test Type

Demo



Value, state, and interaction tests

How Much Test Coverage Is Enough

Context

Framework or public API

Line of business application

Service layer

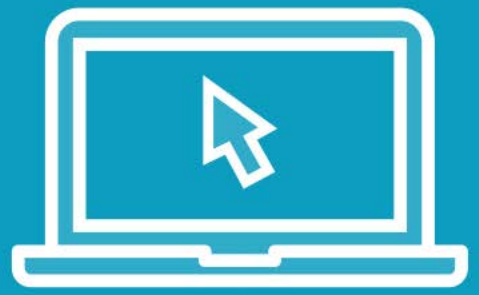
Do I Have Enough Test Coverage?

Business rules

Happy & sad scenarios

Text & anything the compiler would catch

Demo



More than 100% coverage

Summary

Context guides coverage

Strive for SOLID testable code

Think, don’t just stop at X%