# Bringing It Together with Guidelines, Checklists, and Resources



Rusty Divine
SOFTWARE CONSULTANT

@CornerPosts cornerpostsoftware.com





# Are My Tests Effective?

- □ You trust them
- □ You maintain them
- □ You actually read them
- □ They don't get in your way

#### Code Review Guidelines



#### Benefit of Guidelines

Common ground

Power of a checklist

Structure for improvement





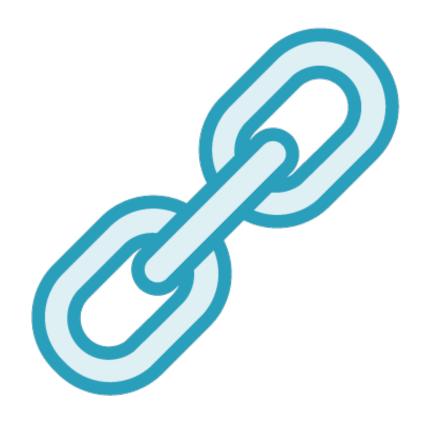
# Unit Testing Checklist

- □ Test name describes the scenario
- □ Contains arrange, act, assert
- Stays within one project layer
- □ Is a state, value, or interaction test
- □ Fakes all dependencies
- Mocks at most one dependency
- □ Favors the public API
- □ Asserts against one object
- □ Favors builder over setup methods



## Resources





#### **Google Testing Blog:**

https://testing.googleblog.com/search/label/TotT

Suggested articles:

**Don't Overuse Mocks** 

**Testing State vs. Interactions** 

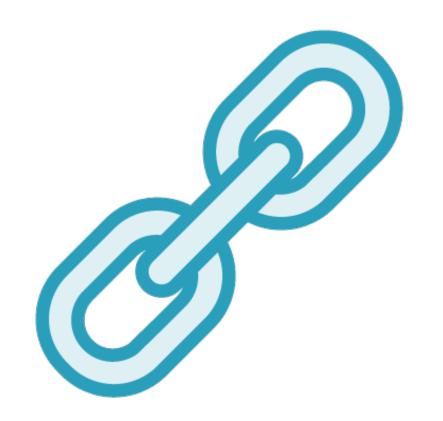
Prefer Testing Public APIs Over Implementation-Detail Classes

**Test Behaviors not Methods** 

**Effective Testing** 

**Measuring Coverage** 



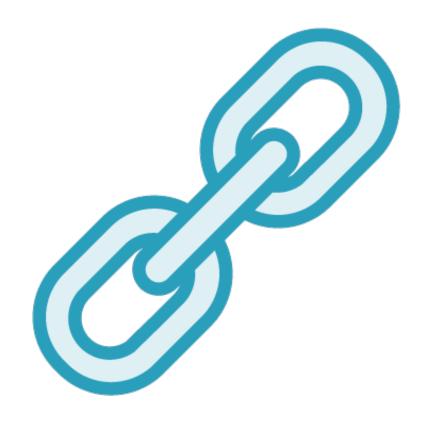


The Art of Unit Testing: with examples in C# 2<sup>nd</sup> Edition, by Roy Osherove

http://amzn.to/2elq1v6

http://artofunittesting.com/





My unit testing links: <a href="http://osmy.in/UnitTestLinks">http://osmy.in/UnitTestLinks</a>

My code review guidelines: <a href="http://osmy.in/UnitTestGuidelines">http://osmy.in/UnitTestGuidelines</a>



# Summary



Context matters, opinions vary; strive for more effective unit tests

Team agreement and training is vital

Clear & simple tests, focused on riskiest code, that don't inhibit refactoring

