

# Assessing and Testing SOLID Code



**Steve Smith**

Founder and Principal Architect, NimblePros

@ardalis | [ardalis.com](http://ardalis.com)

# Overview



**Quick Code Review and Analysis**

**Adding Some Unit Tests**

**Adding Some Integration Tests**

**What Next?**





## Assessing Code Quality Over Time

- Maintainability
- Complexity
- Size / Lines of Code



# The Importance of **Unit** Testability

# **Unit Testable Code...**

- Is loosely coupled**
- Minimizes side effects**
- Avoids global state changes**
- Supports configurable state**
- Is observable**
- Has many seams**
- Is modular**
- Has clear, predictable behavior**



# Unit Testable Code

shares many characteristics  
with

# High Quality Code



# Demo



## Unit Testing SOLID Code



# The Importance of **Integration** Testing

**Integration Tests ensure  
your system works with its  
dependencies and that its  
internal modules work with  
one another.**



# Should I unit test or integration test my controllers / endpoints?



# **ASP.NET**

## **Integration Tests**

### **With**

#### **WebApplicationFactory**

**Nearly as fast as unit tests**

**More resilient to refactoring**

**Can test**

- Routes
- Middleware
- Filters
- Model binding / validation

**Prefer over unit tests for controllers/endpoints**

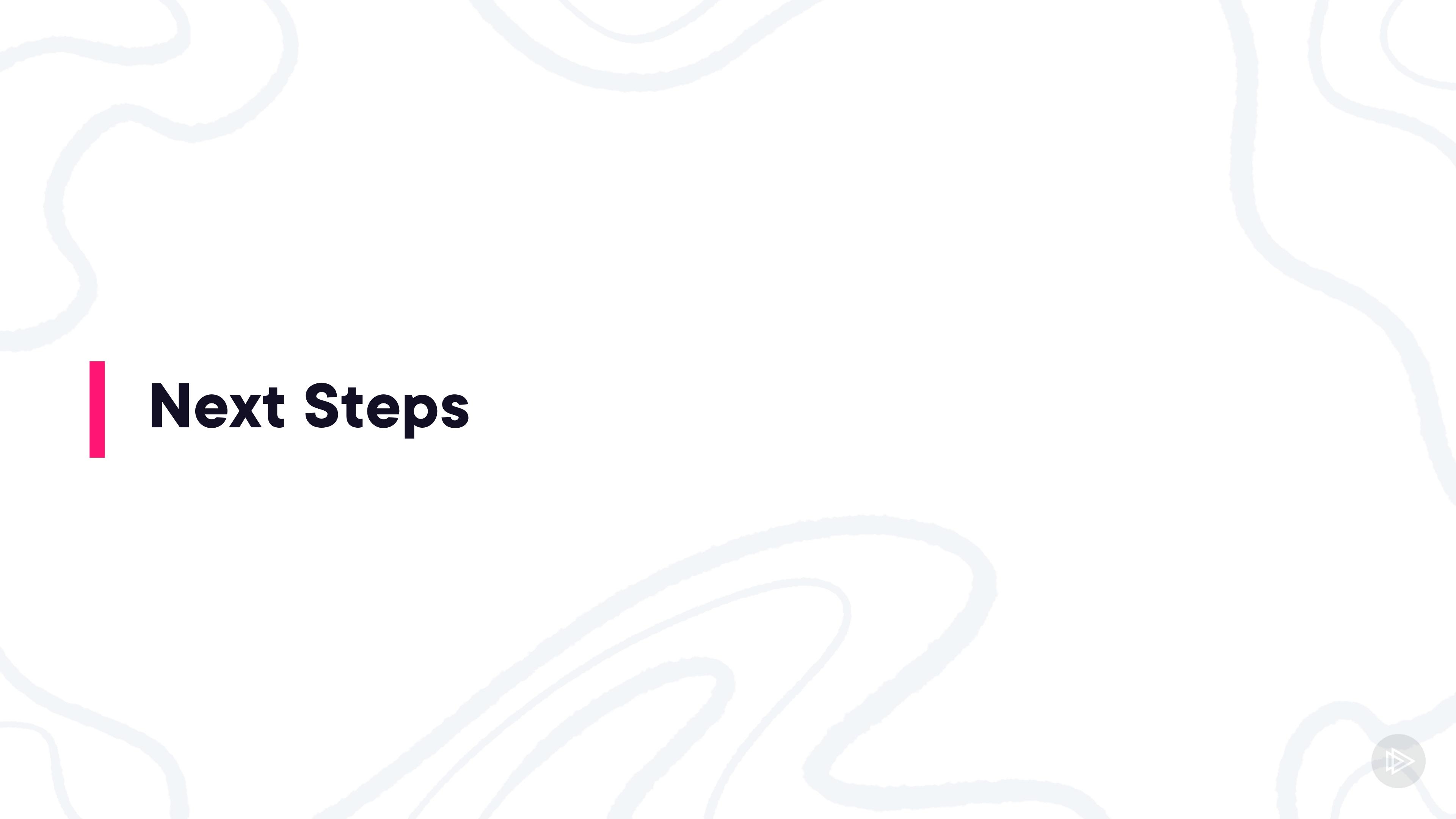


# Demo



## Integration Testing SOLID Code





# Next Steps



# My Next Steps

**Introduce an O/RM**

**Write More Unit Tests**

**Enhance Model Types with More Behavior**



# Summary



Evaluating Progress

Unit Tests

Integration Tests

Next Steps



# Where to Go Next



**Steve Smith**

Founder and Principal Architect, NimblePros

@ardalis | [ardalis.com](http://ardalis.com)