

A tablet is positioned on the left side of the frame, resting on a wooden desk. The screen shows an email inbox with a blue header bar and several email entries. The background is a blurred view of a desk with various items, including what appears to be a laptop keyboard and some papers.

Clean Testing

Clean Architecture with .NET Core

NDC { Oslo } – June 2020

Join the Conversation #CleanTesting @JasonTaylorDev



Jason Taylor

SSW Solution Architect

Developer, Mentor, Speaker & Trainer

.NET 5 (since preview 1.0)

.NET Core (since 1.0)

.NET (since 1.0 in 2002)



[jasontaylordev](#)



github.com/jasontaylordev



jasontaylor.dev



youtube.com/jasontaylordev

Join the Conversation #CleanTesting @JasonTaylorDev

Agenda

Clean Architecture

Clean Testing

Testing Tools

Live Demos

Resources

Clean Architecture

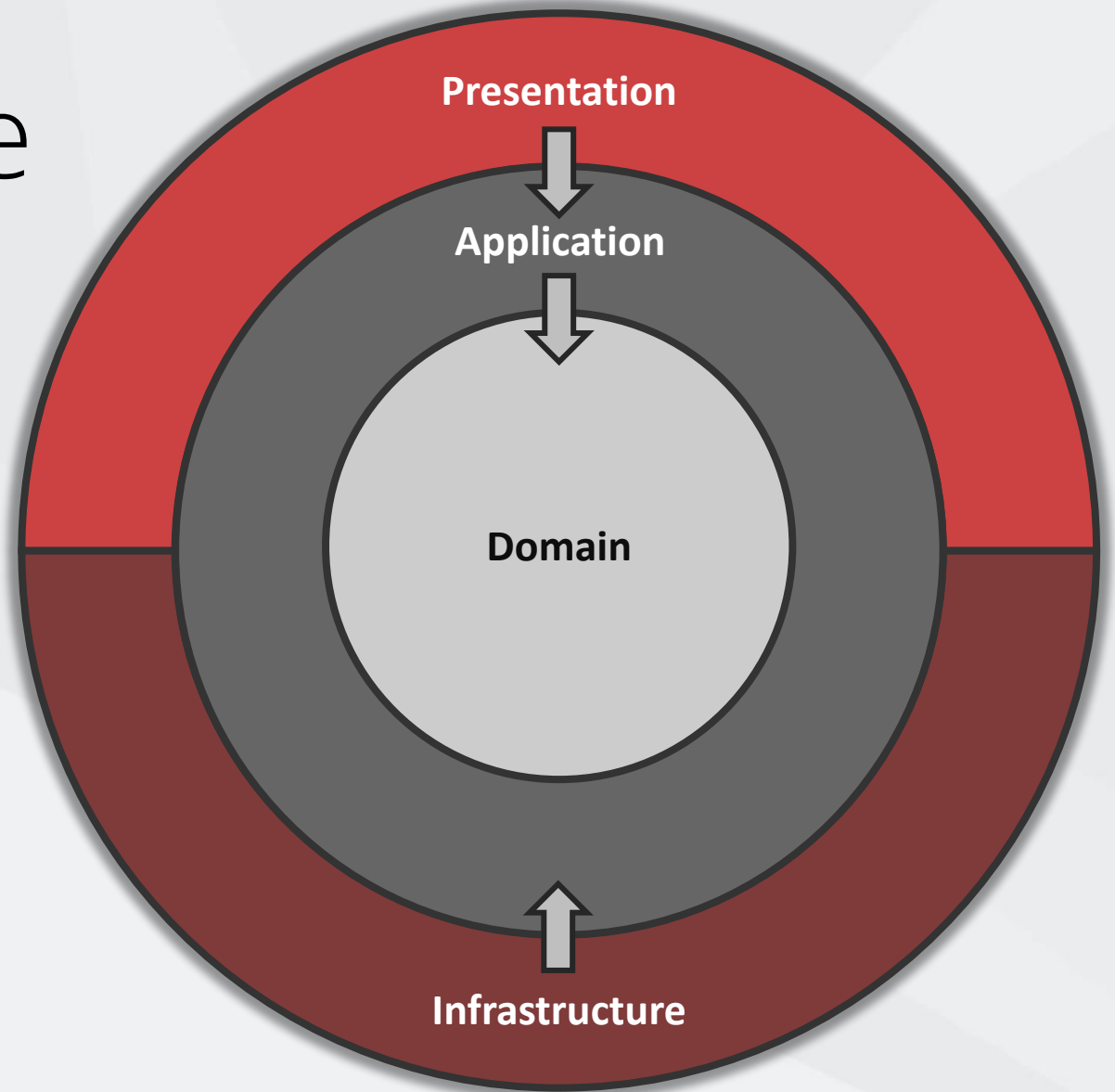
Independent of frameworks

Testable

Independent of UI

Independent of database

Independent anything external



Clean Architecture with ASP.NET Core 3.0

NDC {Sydney} 2019

 youtu.be/5OtUm1BLmG0



Agenda

Clean Architecture

Clean Testing

Testing Tools

Live Demos

Resources

Typical Approach

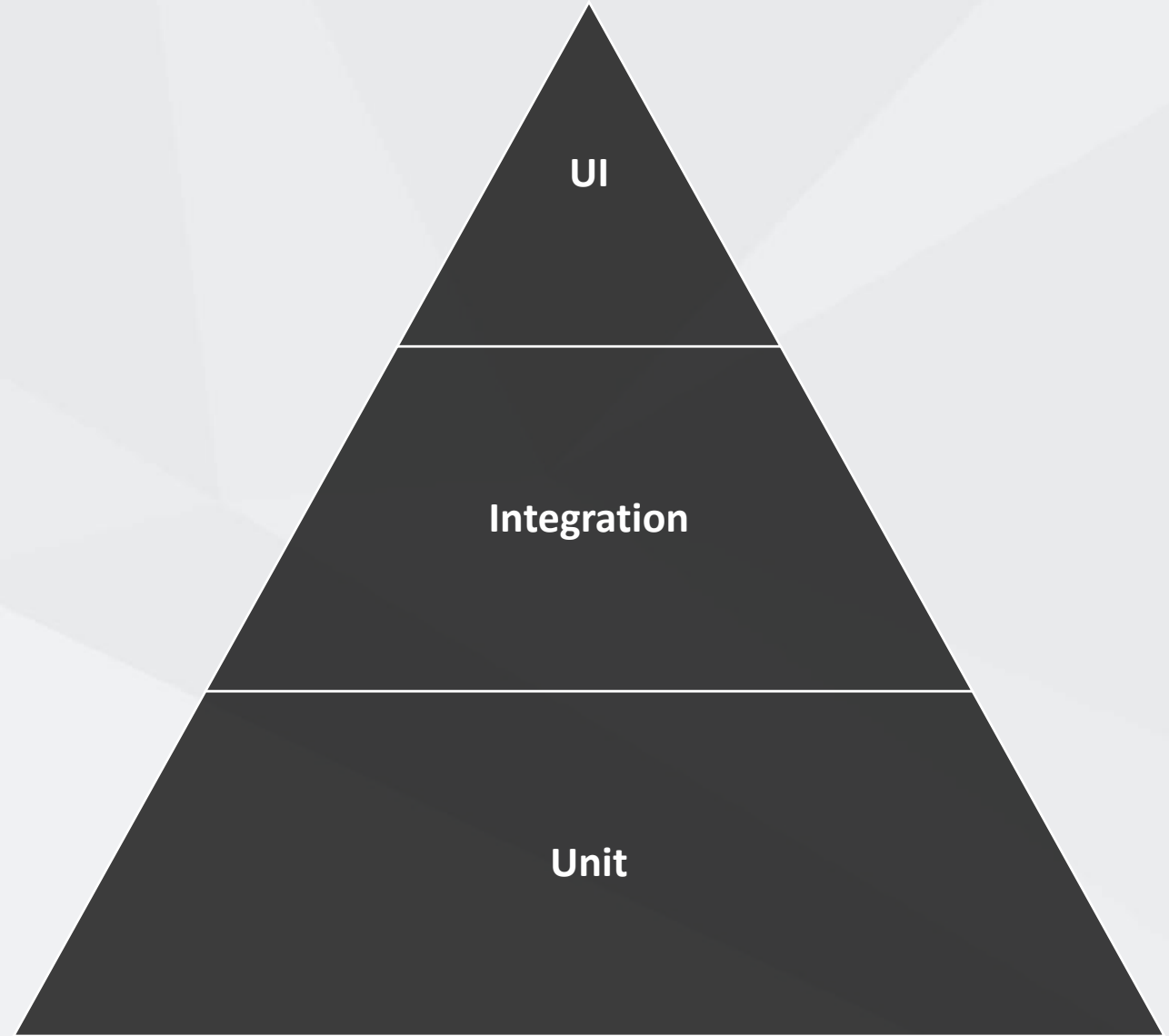
1. Unit tests
2. Integration tests
3. UI tests

Expensive

Slow

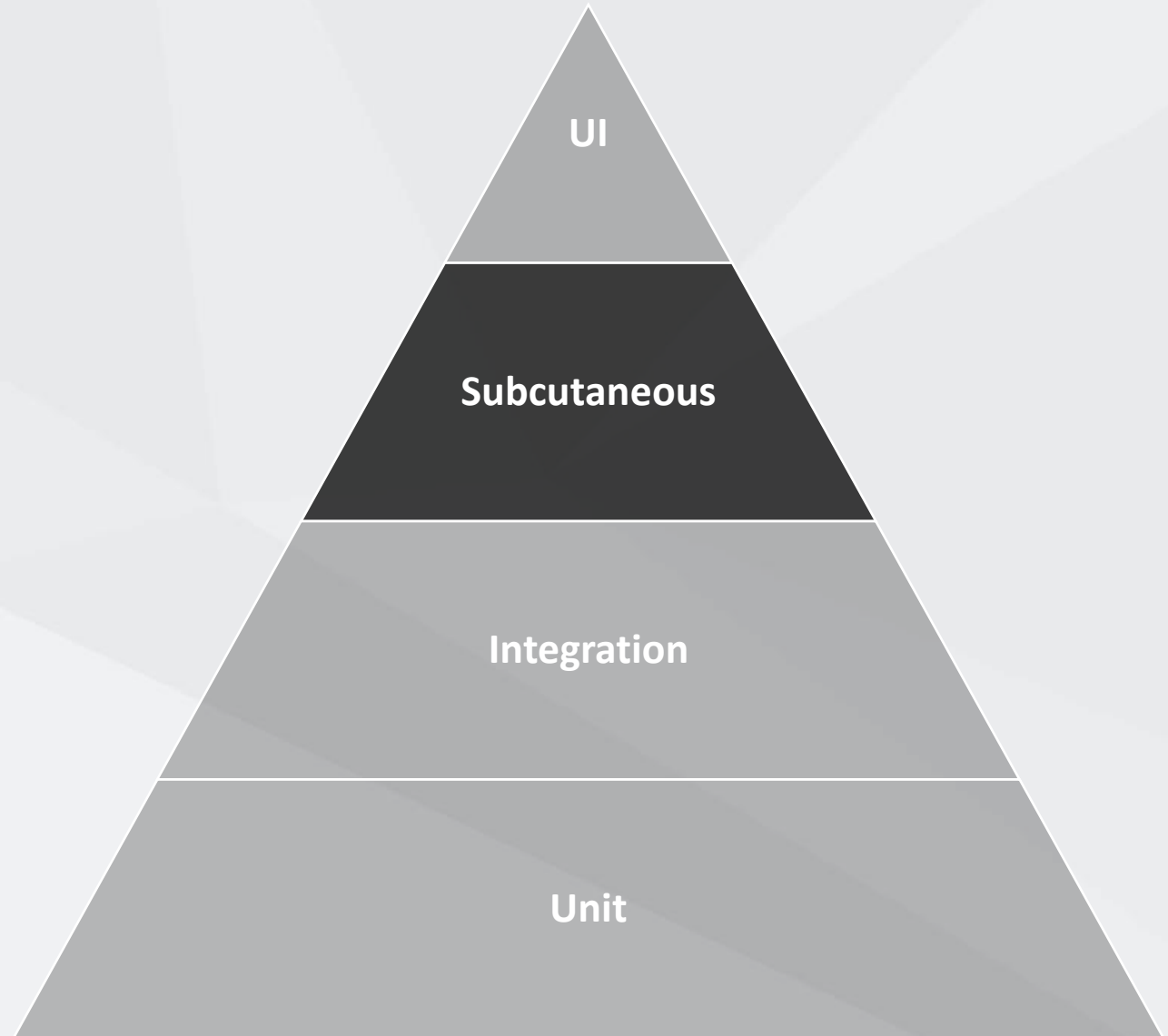
Cheap

Fast



Clean Testing

1. Subcutaneous tests
2. Unit tests
3. Integration tests
4. UI tests



Subcutaneous Tests

A test that operates just below the UI

Verify the basic inputs and outputs of the system

Ideal if you keep logic out of the UI

Reduces the need for UI tests

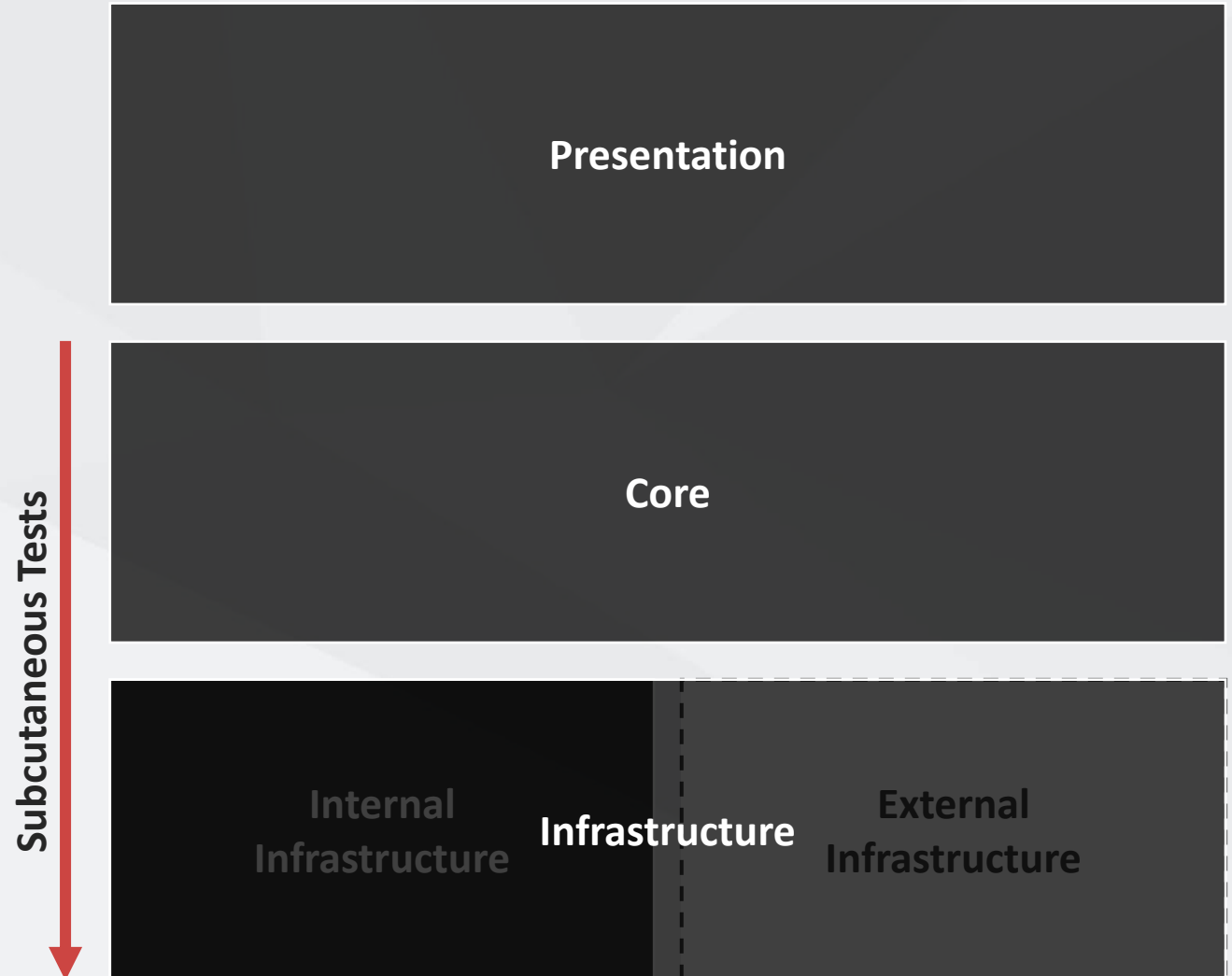
Subcutaneous Tests

Test commands and queries

Include internal services

Mock external services

Keep it simple



Agenda

Clean Architecture

Clean Testing

Testing Tools

Live Demos

Resources

Testing Frameworks



FluentAssertions



Naturally specify the outcome of a test

Improved readability within tests (better assertions)

Clearly explains test failure results (better test results)

Helps to reduce the need to debug tests

Overall, improves productivity and simplifies approach

Moq



Mocking framework for .NET

Very popular and friendly (according to repo)

Supports mocking classes and interfaces

Strongly typed (so no magic strings)

Simple to use, no prior knowledge required

Respawn

Intelligent database cleaner for integration tests

Instead of deleting data or rolling back transactions

Resets database to a clean checkpoint

Ignore tables / schemas, e.g. __EFMigrationsHistory

Supports SQL Server, Postgres, MySQL, ...

<https://github.com/jbogard/Respawn>

Agenda

Clean Architecture

Clean Testing

Testing Tools

Live Demos

Resources

Clean Architecture Solution Template

.NET Core Template Package

ASP.NET Core 3.1

Angular 9

Entity Framework Core 3.1

ASP.NET Core Identity 3.1



Using the Solution Template

A screenshot of a Windows Command Prompt window. The title bar reads "C:\ Command Prompt". The command history shows the user running two commands: "dotnet new -i Clean.Architecture.Solution.Template" and "dotnet new ca-sln". The output of the second command is "The template 'Clean Architecture Solution' was created successfully." The prompt is currently at "C:\Code\CaTodo>". In the bottom right corner of the terminal area, there is a purple icon consisting of a gear with three cubes inside it.

```
C:\Code\CaTodo>dotnet new -i Clean.Architecture.Solution.Template
C:\Code\CaTodo>dotnet new ca-sln
The template "Clean Architecture Solution" was created successfully.
C:\Code\CaTodo>
```

Demo: Clean Testing



Create tests that are easy to write and maintain with
NUnit, FluentAssertions, Moq, and Respawn

Agenda

Clean Architecture

Clean Testing

Testing Tools

Live Demos

Resources

Resources

Clean Testing Demo Code

github.com/jasontaylordev/ndc-oslo-2020-clean-testing

Clean Architecture Solution Template

github.com/jasontaylordev/cleanarchitecture

Contoso University

<https://github.com/jbogard/ContosoUniversityDotNetCore-Pages>

Fixie Demo

<https://github.com/fixie/fixie.demo>

Summary

Clean tests are simple to create, maintain, and understand

Configured to run production services against a real database

Testing tools such as Moq and FluentAssertions improve simplicity and productivity

Respawn is an amazing database cleaner for integration tests

Better test coverage and higher confidence

Thank you!



jasontaylordev

info@ssw.com.au

www.ssw.com.au

Sydney | Melbourne | Brisbane | Adelaide