Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Test-Driven Development with .NET MAUI

Class Duration

35 hours of live training delivered over 5 days.

Student Prerequisites

- Professional C# programming experience
- Working knowledge of .NET MAUI fundamentals: XAML, MVVM, and Shell navigation (see *Build Cross-Platform Apps with .NET MAUI*)
- Familiarity with Git
- No prior testing experience required

Target Audience

C# developers and teams building .NET MAUI apps who want the confidence of a test-first workflow. Ideal for teams shipping business-critical mobile and desktop apps where regressions are expensive — field tools, data-collection apps, and line-of-business clients — and for organizations that require automated test coverage as part of their delivery pipeline. This is the test-driven companion to our *Build Cross-Platform Apps with .NET MAUI* course.

Description

Mobile apps are notoriously hard to test after the fact — platform APIs, navigation, and device state weave through everything. Test-driven development inverts the problem: by writing tests first, you are forced into the architecture that makes MAUI apps testable at all. This five-day course teaches that discipline hands-on with .NET 10. Day one grounds students in TDD itself — red-green-refactor, test organization with xUnit, and the economics of fast feedback — and establishes the MAUI testability architecture: thin views, MVVM view models, and platform services behind interfaces. Day two applies TDD to the heart of a MAUI app: driving view models test-first with CommunityToolkit.Mvvm, testing commands, validation, and async flows, and faking time, connectivity, and other environmental dependencies. Day three tackles the hard parts: abstracting and mocking platform APIs (geolocation, sensors, camera, permissions) with NSubstitute, testing navigation against Shell abstractions, and TDD for offline data — SQLite repositories and sync logic under test. Day four moves up the

pyramid: integration tests for the data and service layers, device-based test execution, and automated UI testing with Appium on emulators and real devices — including what belongs in UI tests and what does not. Day five puts it in the pipeline: CI workflows that build and test Android and iOS targets, device test strategies, coverage and mutation testing as honesty checks, and applying AI coding assistants to test generation without surrendering the TDD discipline.

Learning Outcomes

- Practice red-green-refactor fluently and organize xUnit test suites that stay fast and maintainable.
- Structure MAUI apps for testability: thin views, MVVM view models, and platform services behind interfaces.
- Drive view models test-first: observable state, relay commands, validation, and async flows.
- Isolate tests with mocks, stubs, and fakes using NSubstitute, including time, connectivity, and randomness.
- Abstract and mock platform APIs — geolocation, sensors, camera, permissions — for deterministic tests.
- Test navigation logic against Shell abstractions without launching the app.
- Apply TDD to local data: SQLite repositories, migrations, and offline sync logic.
- Write integration tests for service and data layers, and run test suites on devices and emulators.
- Automate UI tests with Appium, and decide what belongs at each level of the test pyramid.
- Build CI pipelines that compile, test, and report on Android and iOS targets.
- Use coverage and mutation testing to keep the suite honest.
- Apply AI coding assistants to test generation while preserving test-first discipline.

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Software Requirements

.NET 10 SDK, Visual Studio 2026 (Windows) or VS Code with the .NET MAUI extension (Windows or Mac), Android SDK with an emulator, Appium, and Git. A Mac with Xcode is required only for the iOS labs; all other labs run on Windows or Mac.

Training Topics

TDD Foundations

- Red-green-refactor: the discipline and the payoff
- Test organization with xUnit: fixtures, theories, and naming
- The test pyramid for mobile apps
- The economics of fast feedback: what makes suites slow and flaky

Architecting MAUI for Testability

- Thin views, testable view models: where logic belongs
- Dependency injection in MauiProgram as the seam for testing
- Platform services behind interfaces
- Humble objects at the platform boundary

Test-First View Models

- Driving CommunityToolkit.Mvvm view models from failing tests
- Observable properties, relay commands, and CanExecute logic
- Validation rules test-first
- Async operations: testing loading states, cancellation, and errors

Isolation Techniques

- Mocks, stubs, and fakes with NSubstitute
- Faking time, connectivity, and environmental state
- Test data builders and object mothers
- Avoiding over-mocking: testing behavior, not implementation

Mocking the Platform

- Abstracting geolocation, sensors, camera, and share
- Permission flows under test: granted, denied, and degraded paths
- Connectivity-dependent logic: offline and flaky-network scenarios
- Testing platform-conditional code

Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Navigation Under Test

- Shell navigation behind an abstraction
- Testing route logic, parameters, and guards
- Modal flows and result passing

TDD for Local Data and Sync

- SQLite repositories test-first
- In-memory and on-disk test strategies
- Migrations under test
- Offline-first sync logic: conflict cases as test cases

Integration and Device Testing

- Service-layer integration tests with real HTTP via test servers
- Running test suites on emulators and devices
- Automated UI testing with Appium: selectors, waits, and stability
- What belongs in UI tests — and what does not

CI/CD for Tested MAUI Apps

- Pipelines that build and test Android and iOS targets
- Emulator strategies and device farms in CI
- Coverage reporting and mutation testing as honesty checks
- Test results as release gates

AI-Assisted Testing

- Generating tests with AI coding assistants — within the TDD loop
- Reviewing AI-generated tests for real assertions
- Coverage-gap analysis with AI tools