Test Automation in iOS

Chris Woodard, Tampa Bay Cocoaheads

Software Testing

Software Testing

- QA testing before shipping
- Test while developing
- Regression testing while fixing bugs
- Performance testing
- Security testing

Testing Cycle

- Set up initial conditions
- Perform test
- Record results
- Evaluate against expected results

Manual Tests

Testers

- Developer
- QA Tester
- Beta Tester
- Customers (!)

Test Plan

- Paper
- Spreadsheet
- Google Docs
- Etc.

When?

- At end of development - waterfall
- Throughout development - “agile”
- As you’re developing

Automated Tests

Types of Test

- Unit - single piece of code
- Integration - multiple pieces of code
- UI/User Flow - whole app

Unit & Integration Tests

- Need access to classes in system under test
- Need to test async code
- Use XCTAssert* macros to test expected conditions

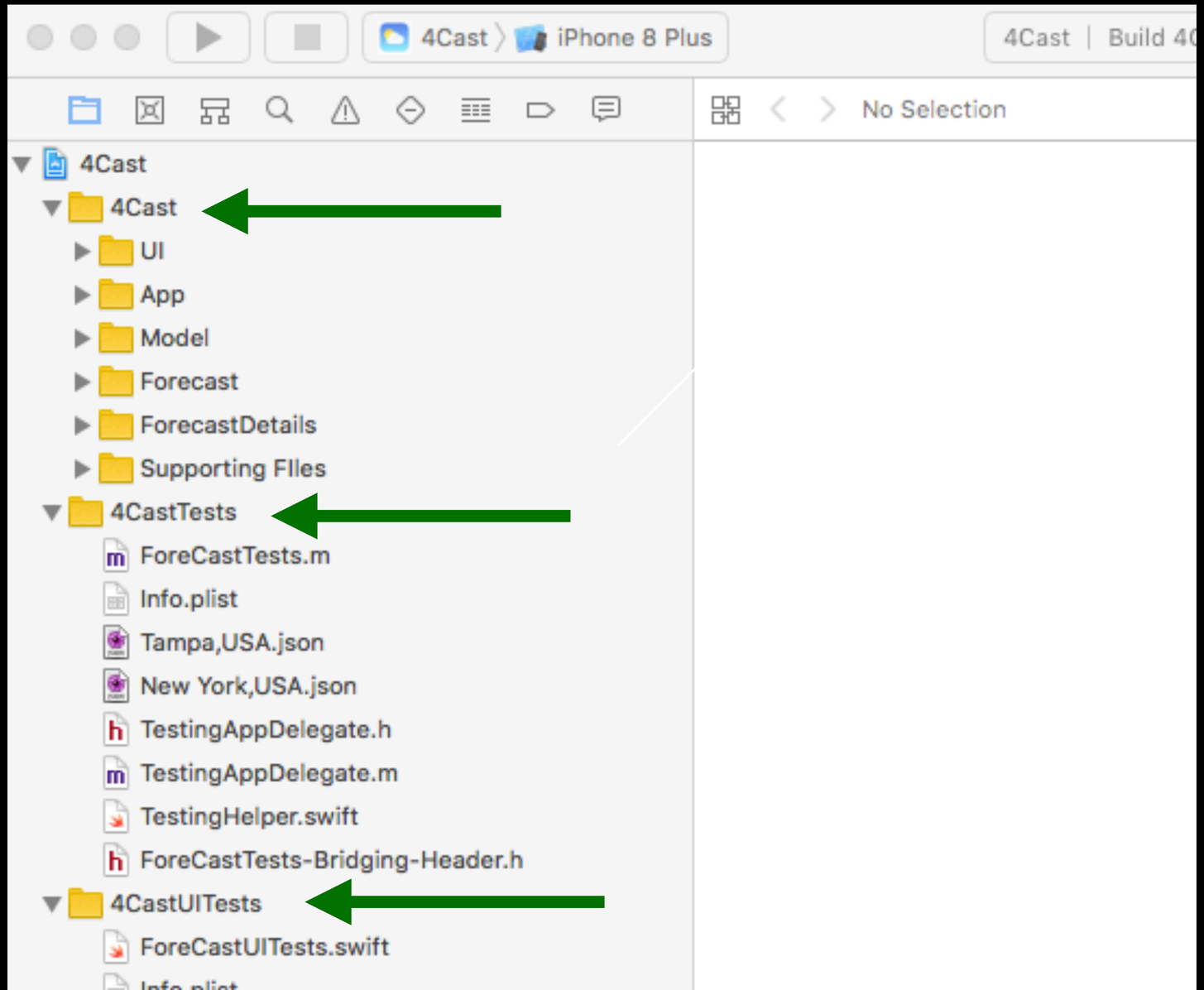
Unit & Integration Tests

- Initial conditions must be reproducible. Use mocks.
- Design (or refactor) SUT code to use switchable mocks.

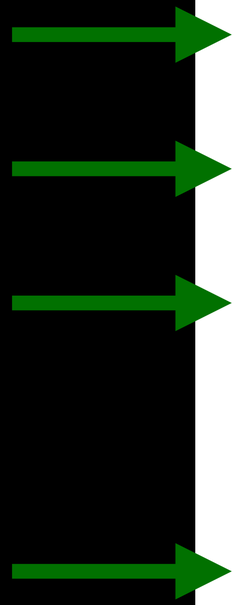
Mocks

- Object or method that delivers canned data in place of live data
- JSON, Images, etc.
- Where the mock goes is important.

Automated Tests in iOS



Unit & Integration Tests in iOS



```
-(void)testWeatherForTampaFL {
    XCTestExpectation *expectation = [self expectationWithDescription:@"asynchronous
        request"];
    __block NSDictionary *theForecast = nil;
    __block NSError *anythingHappened = nil;

    [self.restClient forecastForCity:@"Tampa,USA" completion:^(NSDictionary *results,
        NSError *err) {
        theForecast = results;
        anythingHappened = err;
        [expectation fulfill];
    }];

    [self waitForExpectationsWithTimeout:10.0 handler:nil];

    XCTAssertNotNil(theForecast, @"Unable to get the forecast");
    XCTAssertNil(anythingHappened, @"Got error: %@", anythingHappened);
}
```

Demo

Unit & Integration Tests

- Best used for TDD
- Can be used for “library testing”
- Benefit from helper methods

UI Automation Tests

UI Automation Tests

- Performed by XCUITest class
- Simulates user using app
- Needs knowledge of UI and annotations.

UI Automation Tests

- XCUITest uses UIAccessibiilty
  - Accessibility IDs
  - Accessibility values
- Operates the app with them

Demo

Wrapup

- Unit & integration testing pays off
- Use TDD for Unit/integration
- Designing for testability matters
- Mocks rock

Resources

- *Apple Documentation on Xcode Testing* - https://developer.apple.com/library/content/documentation/DeveloperTools/Conceptual/testing_with_xcode/chapters/01-introduction.html
- *Apple Documentation on UI Testing* - https://developer.apple.com/library/content/documentation/DeveloperTools/Conceptual/testing_with_xcode/chapters/09-ui_testing.html#//apple_ref/doc/uid/TP40014132-CH13-SW1Accessibility IDs
- https://www.raywenderlich.com/150073/ios-unit-testing-and-ui-testing-tutorial
- https://bitbar.com/the-basics-of-xcuitest-and-using-xcode-ui-test-recorder/
- https://medium.com/@johnsundell/getting-started-with-xcode-ui-testing-in-swift-ac7b1f5101e5
- https://github.com/sciprojguy/4Cast