

Automated UI testing, for example Appium tests - benefits:

- Managers understand them
- A few minor things

Unit-style tests over UI tests:

- Faster, even with simulator required
- Finer grained tests, more scenarios
- Many sensors unsuceptible to UI tests. Example:
 - Camera
 - Network drops
 - accelerometer
- No app navigation required
- More modular, loosely-coupled code
- Unrealistic to drive code with UI tests

Tricks for testing

Apple does not care about supporting TDD. Apple cares about APPEARING to be supporting TDD.

- Mocking
 - Trigger: Derived classes with log fields for testing
 - Action: Homegrown mocking library
 - Benefits: Mocks simpler and more consistent, less aggravating
 - Dangers: Misspellings, leaks of production functionality
 - Code not available since it's closed source (doh!)
 - Use dictionary for objects/functions
 - Use another dictionary to mock return values and the functions they are associated with
- Quarantine (Dan's name for this)
 - Trigger: Required superclass, required methods, or required instance fields
 - Action: Quarantine requirements in tiny class, delegate everything else to tame class
 - Benefits: much more testable: one hand (mock) washes the

- other
 - Dangers: Circular references
- Injected Singleton
 - Trigger: Static sharedInstance
 - Action: Instance field in AppDelegate, initialized in .application(...)
 - Benefits: Injectable, mockable and testable
 - Dangers: Forgotten cleanup killing distant test
 - Example: Wrap CMMotionManager
 - open static left sharedInstance = CMMotionManager()
- Constructor Seam
 - Trigger: Collaborator construction with important dependencies not known at subject construction time
 - Action: Injectable initialized closure instance field
 - Benefits: Get your mocks into play, even late
 - Dangers: Unclear and confusing when unfamiliar or badly named; forgotten test
- Delegate Classes
 - Trigger: Instance field of type that needs a delegate
 - Action: Define separate class for delegates; do not add protocol to containing class
 - Benefits: functionality is isolated and clear; delegates can be mocked; can be tested with minimal setup; no switch statement
 - Dangers: circular references
-