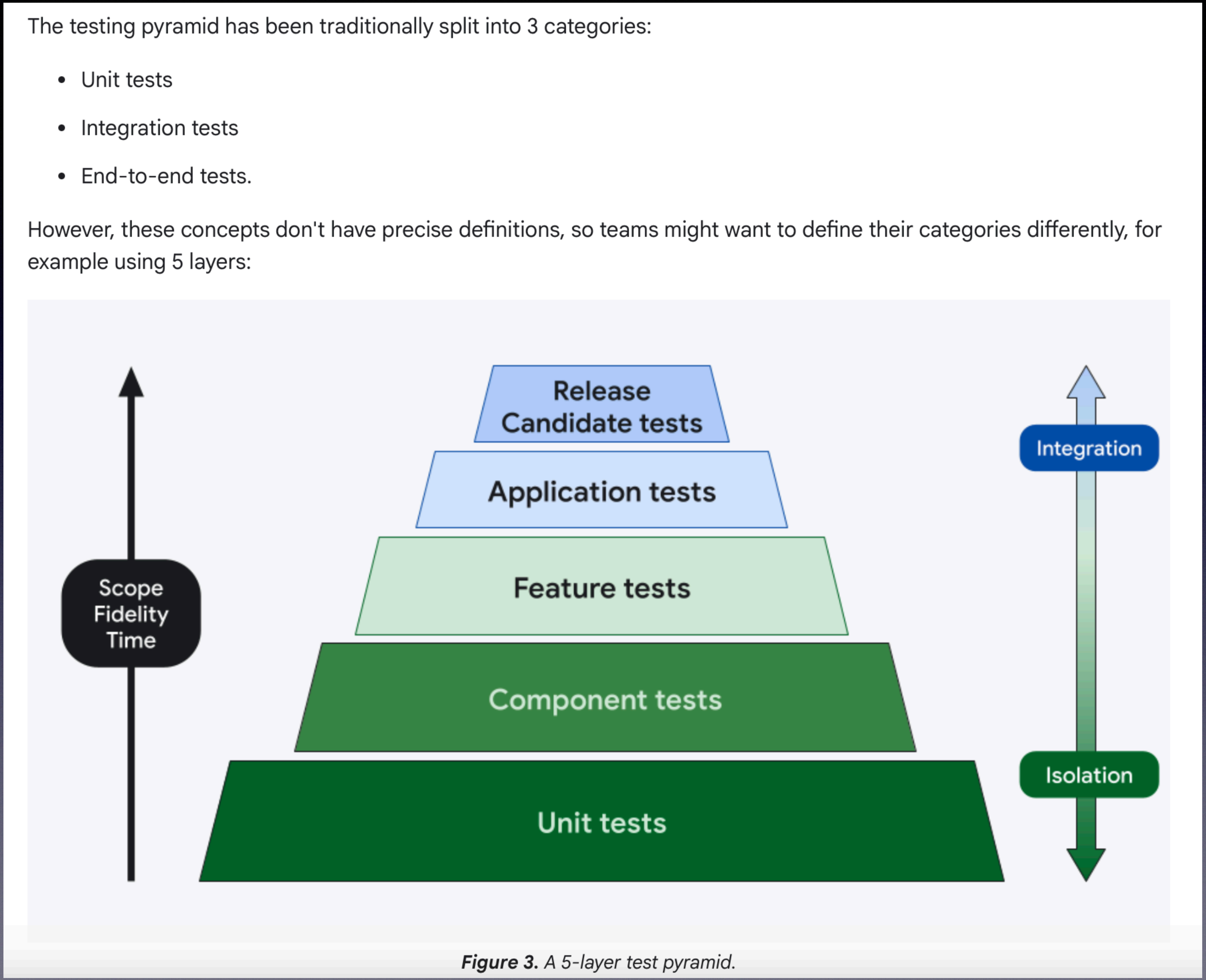
Mobile Apps UI Testing

Android

UI testing strategy



The testing pyramid has been traditionally split into 3 categories:

- Unit tests
- Integration tests
- End-to-end tests.

However, these concepts don't have precise definitions, so teams might want to define their categories differently, for example using 5 layers:

Figure 3. *A 5-layer test pyramid.*

Screenshot tests (device/emulator)

UI tests (device/emulator)

Robolectric (JVM)
Screenshot tests (JVM)

Android Mobile Applications

- Native
- Hybrid
- Mobile Web Apps
- Cross-Platform Apps

Android Mobile Applications

- Native apps written in Kotlin, Java, or any other language that compiles to Android’s bytecode (e.g., C++)
- Hybrid apps use web technologies (HTML, CSS, JavaScript) within a native wrapper
- Mobile Web apps are accessed through a mobile browser
- Cross-Platform apps - Flutter (Dart lang), React Native (JavaScript or TypeScript)

Most Popular

- Large-scale complex native apps: Android SDK + Jetpack Compose UI framework (Kotlin lang)
- Cross-platform development: Flutter UI framework (Dart lang)

UI Test Automation

- Native: Espresso, UI Automator, Compose, Appium
- Hybrid: Appium, WebDriverIO + Appium
- Mobile Web Apps: Playwright, Cypress, Selenium
- Cross-Platform Apps: Flutter Integration Test (Flutter), Detox (React Native), Appium

Native UI: Test Frameworks

- Espresso (paired with UI Automator if needed)
- UI Automator (suitable for cross-app functional UI testing across system and installed apps)
- Compose (required to be paired with Espresso or UI Automator to test legacy Views)
- Appium

How to choose

- Jetpack Compose UI: Compose testing framework
- XML Views: Espresso
- Mixed (Compose + XML): Espresso + Compose
- Interacting with system apps or UI outside your app: UI Automator

Cloud Platforms for Android UI Testing
Device farms/Emulators

- Firebase Lab
- BrowserStack
- Sauce Labs
- AWS Device Farm

Visual UI testing (screenshot testing)

- JVM: Roborazzi (Robolectric), Paparazzi (Layoutlib) testing libraries
- Device/Emulator: Shot
- Cloud-based (device farms/emulators): Applitools, BrowserStack (Percy), Firebase Test Lab + Shot

Different types of UI tests

- End-To-End tests
- Atomic UI tests/Fake data UI tests
- Visual UI/Screenshot tests
- Accessibility tests

Keep Test Automation Effective

- Right tools
- Right test coverage
- Simplicity
- Framework structure
- Minimize third-party dependencies
- Simplify reporting and logs
- Build with long-term maintenance in mind