

Testarea sistemelor software XC TEST vs Appium

**Care e mai potrivit pentru testarea**

**aplicațiilor mobile ?**

**Profesor : SORINA NICOLETA PREDUT**

Bicu Radu-Florian, Duduman Cristian, Mihai Andrei-Alexandru,

Mihai Dragos-Vasile, Tudor Mihai-Dan, Rus Alexandru

Cuprins

[1. Testarea aplicațiilor Mobile, utilizând Framework-uri 2](#_Toc166415484)

[2. Framework XC Test 2](#_Toc166415485)

[3. Framework Appium 5](#_Toc166415486)

[4. Asemănări 8](#_Toc166415487)

[5. Diferențe 9](#_Toc166415488)

[6. Bibliografie 10](#_Toc166415489)

# Testarea aplicațiilor Mobile, utilizând Framework-uri

Testarea aplicațiilor Mobile joacă un rol foarte important în asigurarea calității și performanței aplicațiilor pe diverse platforme. Echipa noastră a ales să observe diferențele dintre framework-urile Appium și XC Test, acestea două fiind foarte populare în rândul dezvoltatorilor al căror scop e testarea aplicațiilor de mobil. Ne vom concentra pe prezentarea fiecăruia dintre framework-uri, caracteristicile și adecvarea acestora în cadrul testărilor, mai apoi observând diferențele și asemănările dintre cele două.

# Framework XC Test

XCTest este un framework de testare unitară implicit în Xcode, mediul de dezvoltare integrat (IDE) implicit pentru Mac. Acest framework este diferit de XCUITest, care este utilizat pentru testarea automată. De obicei, dezvoltatorii folosesc XCTest pentru testarea unitară în proiectele de nivel enterprise.

## Exemple de utilizare:

* + Testarea funcțiilor matematice: Verificarea corectitudinii operațiior matematice implementate în cod.
  + Testarea logicii de afaceri: Validarea regulilor și logicii specifice aplicației.
  + Testarea conexiunilor la API: Asigurarea comunicării corecte cu serverele externe.

## Limitări:

* + Testarea interfeței grafice: XCTest nu este specializat pentru testarea UI, fiind necesară utilizarea XCUITest sau a altor framework-uri dedicate.
  + Testarea asincronă: Implementarea manuală a așteptărilor este necesară pentru testarea corectă a funcționalităților asincrone.
  + Extensii:
  + Utilizarea framework-urilor de mocking: Permite simularea comportamentului dependențelor externe, facilitând testarea izolată a componentelor individuale.
  + Testarea performanței: Permite măsurarea timpului de execuție și a resurselor utilizate de diverse operații din cod.
  + Generarea automată de teste: Utilizarea instrumentelor dedicate poate automatiza parțial scrierea testelor unitare.

## Avantaje ale XCTest:

1. Automatizarea testelor unitare:
   1. XCTest permite dezvoltatorilor să scrie teste unitare pentru a verifica funcționalitățile individuale ale codului.
   2. Automatizarea testelor unitare reduce efortul manual și asigură că modificările ulterioare nu afectează funcționalitatea existentă.
2. Integrare strânsă cu Xcode:
   1. XCTest este încorporat în Xcode, ceea ce face ușor să începi să scrii teste.
   2. Dezvoltatorii nu trebuie să instaleze sau să configureze nimic suplimentar.
3. Rularea testelor pe simulatoare sau dispozitive reale:
   1. XCTest permite rularea testelor pe simulatoare iOS sau dispozitive reale.
   2. Acest lucru asigură că testezi codul pe platforma reală pe care va rula aplicația.
4. Raportarea detaliată a erorilor:
   1. XCTest furnizează rapoarte detaliate despre erori și eșecuri de testare.
   2. Dezvoltatorii pot identifica rapid problemele și le pot remedia.

## Dezavantaje ale XCTest:

1. Limitări în testarea interfeței grafice (UI):
   1. XCTest nu este specializat pentru testarea interfeței grafice.
   2. Pentru testarea UI, este necesar să folosești XCUITest sau alte framework- uri specializate.
2. Complexitate în testarea asincronă:
   1. XCTest nu gestionează automat așteptările asincrone.
   2. Dezvoltatorii trebuie să implementeze manual așteptările pentru a testa corect funcționalitățile asincrone.

## Exemplu de cod XCTest:

1. **import** XCTest

3. **class** MyTests: XCTestCase {

5.

let calculator = Calculator()

7.

XCTAssertEqual(result, 8, "Sum should be 8")

9. }

11. **class** Calculator {

13.

**return** a + b

15. }

17. // Rularea testelor

19. testRunner.runTests(testClass: MyTests.self)

18. let testRunner = TestRunner()

16.

}

14.

func add(a: Int, b: Int) -> Int {

12.

10.

}

8.

let result = calculator.add(a: 5, b: 3)

6.

func testAddition() {

4.

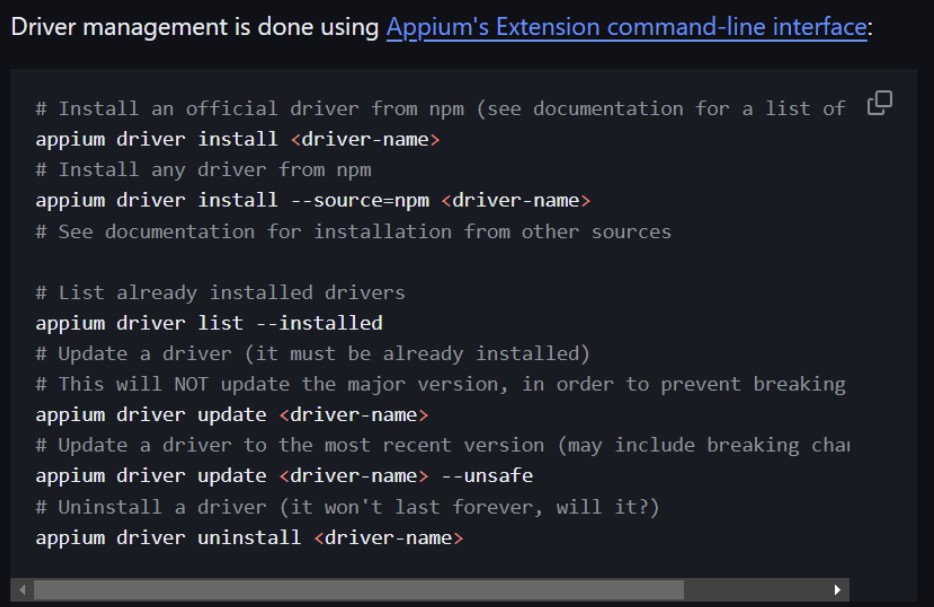
2.

În acest exemplu, am creat o clasă Calculator care are o metodă add(a:b:). Testul testAddition verifică dacă adunarea funcționează corect. XCTest raportează orice eroare dacă suma nu este 8.

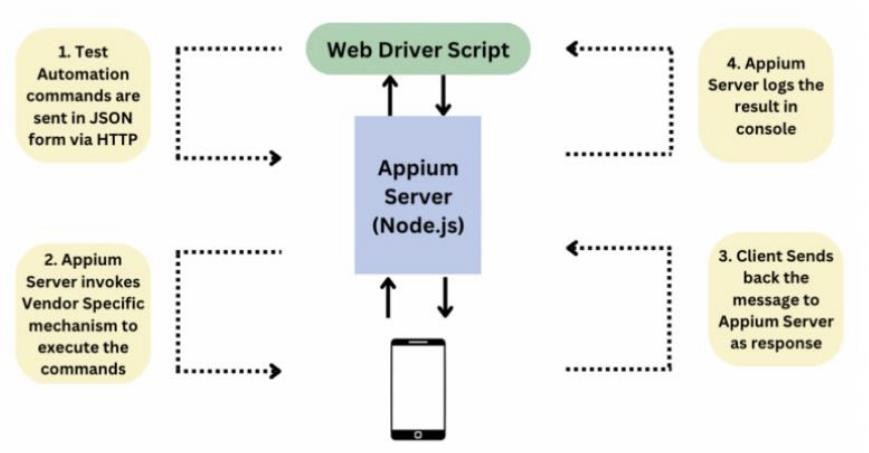
În concluzie, XCTest este un instrument puternic pentru testarea unitară în dezvoltarea de aplicații mobile și poate fi folosit eficient pentru a asigura calitatea și funcționalitatea corectă a codului.

# Framework Appium

Appium este un framework open source care ofera posibilitati bazate pe WebDriver pentru automatizarea multor tipuri de aplicatii (mobile, desktop, IoT). E modular si extensibil, suporta multe limbaje de programare si poate fi integrat in intregi ecosisteme software: Driverele adauga suport pentru platforme specifice: Appium suporta dezvoltare pe multe platforme: iOS, Android, macOS, Windows. Instalarea de drivere din CLI:



Client-urile fac posibila scrierea testelor intr un limbaj de programare ales de catre utilizator Plugin urile fac posibila extinderea functionalitatilor date de catre Appium[[1]](#footnote-1)



Appium permite automatizarea in contextul uneoi sesiuni, clientul deschizand o sesiune conform bibliotecii clientului (java, js, php etc..) care trimite apoi un

request POST /session catre server. Aceasta cerere e insotita de un JSON care descrie capacitatile dorite . Dupa, serverul incepe sesiunea de automatizare si raspunde cu un ID al sesiunii care e folosit in continuare pentru a trimite alte

comenzi sesiunii. Cum “initiaza actiuni” / face asserturi? Pentru iOS, foloseste API ul UIAutomation pentru a interactiona cu interfata userului. Fisierul bootstrap.js

functioneaza ca un server TCP care trimite comenzi care sa actioneze pe device ul iOS (bineinteles, cu ajutorul API ului Apple UIAutomation). Pentru Android e o procedura similara (se foloseste acelasi API).

Particularitati extra care fac Appium asa de popular: - Ofera QA testers control intreg asupra API-urilor din back-end prin codul de testare. Asta ajuta in a evalua defectele care vin din back-end - Real time monitoring - Permite testarea pe

emulatoare/simulatoare sau environment-uri care replica device-uri reale. (sursa: Browserstack appium).

Exemplu de cod:

1. **const** {remote} = require('webdriverio');

3. **const** capabilities = {

5. 'appium:automationName': 'UiAutomator2',

7. 'appium:appPackage': 'com.android.settings',

9. };

8. 'appium:appActivity': '.Settings',

6. 'appium:deviceName': 'Android',

4. platformName: 'Android',

2.

|  |
| --- |
| 10. |
| 11. **const** wdOpts = { |
| 12. hostname: process.env.APPIUM\_HOST || 'localhost', |
| 13. port: parseInt(process.env.APPIUM\_PORT, 10) || 4723, |
| 14. logLevel: 'info', |
| 15. capabilities, |
| 16. }; |
| 17. |
| 18. async function runTest() { |
| 19. **const** driver = await remote(wdOpts); |
| 20. **try** { |
| 21. **const** batteryItem = await driver.$('//\*[@text="Battery"]'); |
| 22. await batteryItem.click(); |
| 23. } **finally** { |
| 24. await driver.pause(1000); |
| 25. await driver.deleteSession(); |
| 26. } |
| 27. } |
| 28. |
| 29. runTest().**catch**(console.error); |

Testul gaseste item-ul Battery din lista si il apasa. Pauzeaza pentru un moment pentru efect vizual Incheie sesiunea de Appium.[[2]](#footnote-2)

# Asemănări[[3]](#footnote-3)

Platforme Suportate:

1. XCTest: Este conceput pentru a testa exclusiv aplicații native iOS și macOS.
2. Appium: Susține testarea aplicațiilor pe diverse platforme, inclusiv iOS, Android și platforme hibride.

Limbaje de Programare:

1. XCTest: Este strâns integrat cu limbajele native de programare Swift și Objective-C.
2. Appium: Permite testarea în multiple limbaje de programare, precum Java, Python, Ruby, JavaScript etc.

Integrare în Medii de Dezvoltare:

1. XCTest: Vine integrat în Xcode, mediul de dezvoltare furnizat de Apple.
2. Appium: Poate fi integrat cu diverse medii de dezvoltare și IDE-uri fără a depinde de un anumit IDE.

Flexibilitate și Scalabilitate:

1. XCTest: Furnizează un set solid de funcționalități pentru testarea unitară și a performanței, dar poate fi mai puțin flexibil în ceea ce privește testarea pe mai multe platforme sau în cloud.
2. Appium: Este mai flexibil și scalabil, permitând testarea pe multiple platforme și dispozitive, inclusiv în cloud.

Simularea Interacțiunii Utilizatorului:

1. XCTest: Are capacități limitate pentru simularea interacțiunii utilizatorului.
2. Appium: Permite simularea interacțiunii utilizatorului cu elementele interfeței utilizatorului (UI) ale aplicației.

Suport pentru Framework-uri de Testare:

1. XCTest: Este un framework nativ care poate fi integrat cu alte framework- uri de testare.
2. Appium poate fi încorporat în diferite framework-uri de testare, cum ar fi Selenium, XCTest și altele.

# Diferențe[[4]](#footnote-4)

XC Test și Appium au unele diferențe semnificative între ele în ceea ce privește arhitectura, compatibilitatea cu platformele și altele.

Vom analiza diferențele dintre acestea împărțite pe mai multe criterii:

1. Limbaj de programare
   1. XC Test este scris în limbajul SWIFT
   2. Appium are o gamă mult mai largă de limbaje de programare (Java, Python, Ruby, etc)
2. Compatiblitatea platformei
   1. XC Test este dedicat exclusiv pentru testarea aplicațiilor IOS si macOS
   2. Appium poate fi folosit atât pentru testarea aplicațiilor pe iOS, cât și pe Android
3. Instrumente de dezvoltare
   1. XC Test – Fiind integrat în Xcode, face ușoară scrierea și rularea testelor
   2. Appium – Utlizatorii trebuie să configureze medii diferite de testare, nefiind integrat în niciun IDE

# Bibliografie

Github Appium, *Appium Documentation*, <https://github.com/appium/appium?tab=readme-ov-file>, 12.05.2024

Appium Documentation, *Write a Test (JS)*, <https://appium.io/docs/en/2.0/quickstart/test-js/>, 12.05.2024

ChatGPT, *“Asemănări între XC Test și Appium”*, <https://chatgpt.com/>, 03.05.2024

ChatGPT, *“Diferențe între XC Test și Appium”*, <https://chatgpt.com/>, 03.05.2024

1. Github Appium, *Appium Documentation*, <https://github.com/appium/appium?tab=readme-ov-file>, 12.05.2024 [↑](#footnote-ref-1)
2. Appium Documentation, *Write a Test (JS)*, <https://appium.io/docs/en/2.0/quickstart/test-js/>, 12.05.2024 [↑](#footnote-ref-2)
3. ChatGPT, *“Asemănări între XC Test și Appium”*, <https://chatgpt.com/>, 03.05.2024 [↑](#footnote-ref-3)
4. ChatGPT, *“Diferențe între XC Test și Appium”*, <https://chatgpt.com/>, 03.05.2024 [↑](#footnote-ref-4)