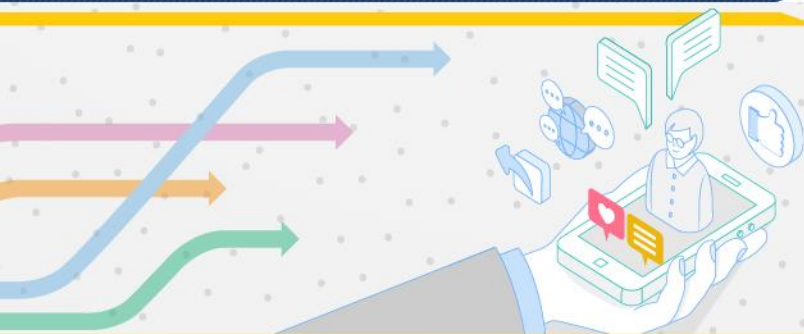


## 애플리케이션 테스트 수행 part 2



# 모바일 테스트

## 학습내용

- 모바일 애플리케이션 테스트 특징과 프레임워크
- 모바일 테스트 도구와 고려 사항

## 학습목표

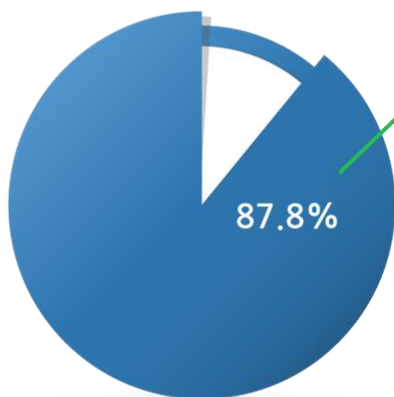
- 모바일 애플리케이션 테스트와 프레임워크의 특징을 설명할 수 있다.
- 모바일 UI 애플리케이션 테스트 자동화 도구와 고려사항을 설명할 수 있다.

## 모바일 애플리케이션 테스트 특징과 프레임워크

### 1 모바일 애플리케이션 사용자



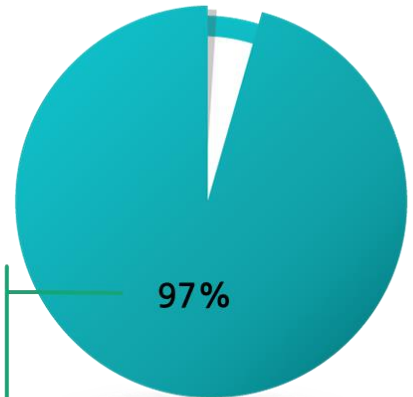
2017년 우리나라 인구에 대한 스마트폰 이용자



[만 3세 이상 인구]

스마트폰  
이용자

스마트폰  
이용자



[10대~50대 인구]

## 모바일 애플리케이션 테스트 특징과 프레임워크

### 1 모바일 애플리케이션 사용자

산업 현장에 있는 모든 인구들은 스마트폰으로 모바일 애플리케이션을 사용한다고 볼 수 있음

스마트폰을 보거나 들을 수 없는 장애인 이외에 대부분의 국민들이 스마트폰을 사용하고 있음

스마트폰의 보급화는 일반화되어 있음을 알 수 있음



**모바일 애플리케이션에 대한 테스트의 중요성은 그 어느 때보다 높다고 할 수 있음**

## 모바일 애플리케이션 테스트 특징과 프레임워크

### 2 모바일 애플리케이션 테스트 특징

1

모바일 환경은 핸드폰에 바로 붙여서 테스트하기 보다는 가상의 시뮬레이터 환경 내부에서 먼저 테스트를 함

2

테스트 품질이 어느 정도 완성되면 핸드폰 환경으로 넘어가서 테스트를 함

3

과거 메모리, CPU, Disk 같은 핸드폰 내부의 자원이 제약적이어서 환경을 고려해서 테스트 해야 했음



현재는 고사양 스마트폰이 나오기 때문에 이러한 제약은 거의 없어졌음

## 모바일 애플리케이션 테스트 특징과 프레임워크

### 2 모바일 애플리케이션 테스트 특징

#### 4 빠른 출시 기간으로 Agile한 테스트 기법이 필요

- ➔ 모바일 애플리케이션 개발 기간은 테스트 기간까지 합쳐서 보통 3~4개월 정도를 잡고 있음
- ➔ 이 기간 동안에 개발도 하고 테스트도 해야 하기 때문에 테스트 기간이 절대적으로 부족함
- ➔ 데스크탑 환경과 같이 테스트 시나리오를 만들고 순차적으로 테스트 케이스를 하나하나 테스트를 할 수 없음
- ➔ Agile한 테스트 기법으로 핵심이 되는 기능 중심으로 빠르게 테스트를 해야 함
- ➔ 모바일 테스트에서는 특히 테스터의 경험과 노하우가 중요함
- ➔ 빠른 시간 내에 핵심 기능을 테스트하기 위해서는 테스터의 경험에 많이 좌우됨

## 모바일 애플리케이션 테스트 특징과 프레임워크

### 3 모바일 UI 애플리케이션 테스트 프레임워크

#### 1 셀레니움(Selenium)

브라우저 기반 웹사이트를 테스트하기 위해 만들어진  
오픈 소스 솔루션

UI 테스트 자동화를 위해 서버와 브라우저 통신에 필요한  
기능셋을 정의한 것

##### 기능셋

- RestFul API 형태

##### 각 브라우저

- 기능셋을 구현



## 모바일 애플리케이션 테스트 특징과 프레임워크

### 3 모바일 UI 애플리케이션 테스트 프레임워크

#### 1 셀레니움(Selenium)

##### 테스트 드라이버

- RestFul API를 개발자가 사용하기 쉽도록 함수로 추상화해놓은 것

##### 테스트 개발자

- 드라이버가 제공하는 API를 사용해 웹 애플리케이션 화면을 구성하는 구성요소의 좌표와 속성을 탐색
- 구성요소에 취할 수 있는 액션을 호출하고, 액션 실행에 대한 결과를 체크하는 시나리오로 테스트 케이스를 작성



## 모바일 애플리케이션 테스트 특징과 프레임워크

### 3 모바일 UI 애플리케이션 테스트 프레임워크

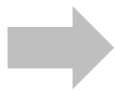
#### 2 에피움(Appium)



모바일 앱 테스트를 위해 셀레니움을 확장해 개발한  
테스트 자동화 프레임워크

앱 설치

삭제



모바일 디바이스를 동작시키기  
위한 기능 추가 정의

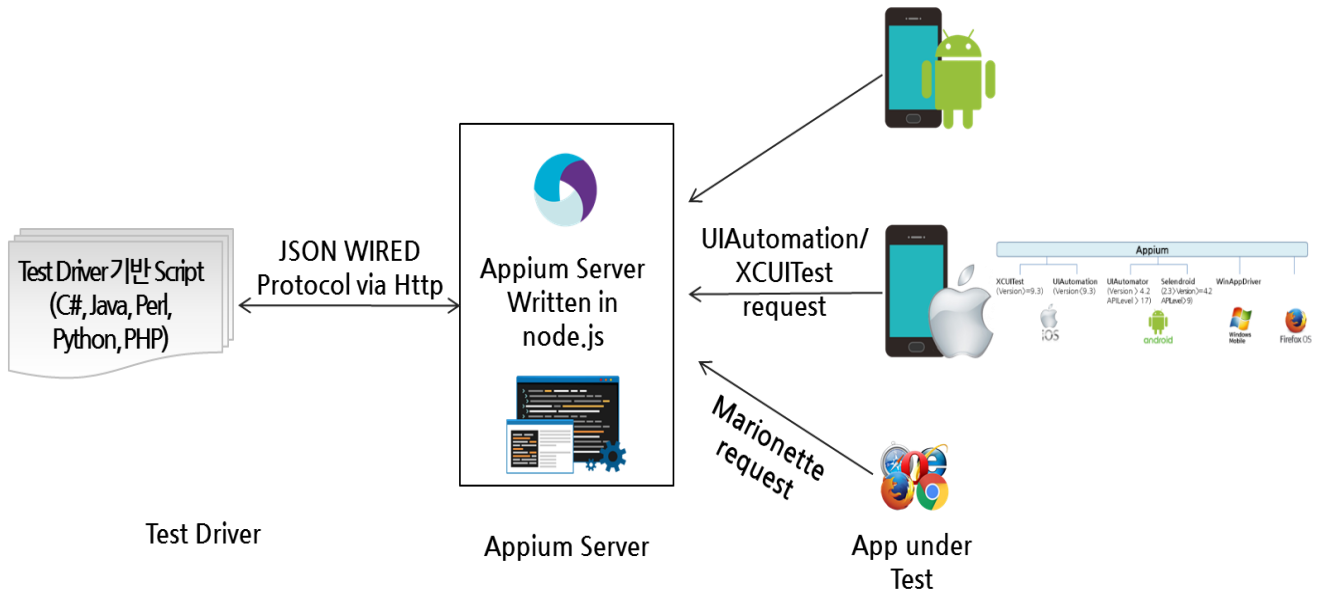
네이티브,  
하이브리드, 웹 등  
모든 종류의  
모바일 앱 테스트  
가능

테스트  
스크립트는  
다양한 언어로  
작성될 수 있음

## 모바일 애플리케이션 테스트 특징과 프레임워크

### 3 모바일 UI 애플리케이션 테스트 프레임워크

#### 2 에피움(Appium)



[에피움 아키텍처]

## 모바일 애플리케이션 테스트 특징과 프레임워크

### 3 모바일 UI 애플리케이션 테스트 프레임워크

#### ② 에피움(Appium)

테스트 스크립트는 에피움 서버와 제이슨 와이어드 프로토콜로 통신함

에피움 서버는 기기에 앱을 설치하고 UI명령어를 전달하는 일련의 과정을 수행하기 위해 플랫폼이 제공하는 툴을 사용함

안드로이드는 디버그 브리지 툴을 제공하며, iOS는 인스트루먼트를 제공함

## 모바일 테스트 도구와 고려 사항

### 1 Android와 iOS의 에피움 기능 지원 비교

에피움  
(Appium)

최근에 나온 모바일 UI 테스트  
프레임워크

iOS와 Android의 지원

모바일 테스트 가능

## 모바일 테스트 도구와 고려 사항

### 1 Android와 iOS의 에피움 기능 지원 비교

결함 유형	iOS	Android
Versions	<ul style="list-style-type: none"> <li>9.0 and up</li> </ul>	<ul style="list-style-type: none"> <li>2.3 and up</li> </ul>
Devices	<ul style="list-style-type: none"> <li>iPhone, iPad Simulator, real iPhone, iPads</li> </ul>	<ul style="list-style-type: none"> <li>Android emulators, real Android devices</li> </ul>
Natives App	<ul style="list-style-type: none"> <li>Debug version .app(simulator), correctly-signed .ipa(real devices)</li> </ul>	<ul style="list-style-type: none"> <li>지원</li> </ul>
Mobile web	<ul style="list-style-type: none"> <li>Mobile safari automation을 통해 지원</li> </ul>	<ul style="list-style-type: none"> <li>Selendroid의 경우 미지원, 4.2와 4.3에서는 공식 Chrome browser, Chromium만 지원</li> <li>4.4 이상에서는 built-in 'Browser' 앱 지원 가능</li> </ul>

## 모바일 테스트 도구와 고려 사항

### 1 Android와 iOS의 에피움 기능 지원 비교

결함 유형	iOS	Android
Hybrid 지원	• 지원	• 지원
한 세션에서 여러 앱 동시 테스트	• 미지원	• 지원
여러 디바이스 동시 테스트	• 미지원	• 지원, 서로 다른 포트 설정으로 가능
벤더 제공 혹은 third- party 앱에 대한 자동화	• 시뮬레이터상에서 벤더 제공 앱만 가능, iOS 10 이상 Home Screen 자동화 가능	• Selendroid의 경우 미지원, 나머지 지원

## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구

구분	자동화 도구	설명
특정 플랫폼에 종속된 도구	Instrument	<ul style="list-style-type: none"> <li>애플이 제공하는 MAC 애플리케이션(iOS 응용 프로그램 개발에 대한 IDE)임</li> <li>iOS앱의 GUI 테스트 자동화를 위해 Xcode에서 추가됨</li> </ul>
	Monkeyrunner	<ul style="list-style-type: none"> <li>안드로이드 장치 또는 원격으로 애플레이터를 제어할 수 있는 프로그램 작성에 API 제공함</li> </ul>
일반 스크립트 기반 도구	Sikuli	<ul style="list-style-type: none"> <li>이미지(스크린샷)를 사용하여 GUI테스트를 자동화하는 Visual 기술임</li> <li>Mac OS X, Windows, Linux, Android 플랫폼 등 모두 지원함</li> </ul>
	Robotframework	<ul style="list-style-type: none"> <li>표나 키워드를 텍스트 형태로 입력하면 이를 읽고 테스트를 자동화하는 스크립트 기반 도구</li> </ul>



## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구

구분	자동화 도구	설명
랜덤 이벤트 생성기	UI/애플리케이션 Monkey	<ul style="list-style-type: none"> <li>Android 장치에서 시스템 레벨의 이벤트뿐만 아니라 클릭 수, 터치, 제스처 등 사용자 이벤트를 임의 순서 스트림으로 테스트 가능한 도구 (Monkeyrunner와는 다름)</li> </ul>
	UIAutomation	<ul style="list-style-type: none"> <li>ios에서 자바 스크립트를 활용하여 터치, 제스처, 시스템 레벨 이벤트와 같은 기능 테스트</li> </ul>
화이트박스 테스트 도구	Android instrument	<ul style="list-style-type: none"> <li>Junit에 기반한 구조, 소스 코딩 로직 테스트</li> </ul>
블랙박스 테스트 도구	Robotium	<ul style="list-style-type: none"> <li>Android 기반 테스트 자동화 도구</li> </ul>

## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구

#### 1 SIKULI(시쿨리)



대표적인 모바일 애플리케이션 테스트 자동화 도구

▶ 이미지 인식을 이용하여 GUI의 테스트를 자동화하는  
비주얼 기술

시쿨리  
스크립트

- GUI요소의 스냅샷을 찍는데 사용할 수 있음
- 일부 참조 스냅샷을 가지고 비교하는데 사용

## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구

#### 1 SIKULI(시쿨리)

시각 중심의 접근 방식 사용

- ▶ 테스트 스크립트 기록기는 대상 애플리케이션 코드의 지식이 없이 테스트 할 수 있음

웹 페이지, 데스크탑 애플리케이션, Android 애플리케이션, 시뮬레이터에서 실행되는 iPhone 애플리케이션을 제어하는데 사용

시쿨리 스크립트는 Jython 스크립트

- ▶ 다른 사용 가능한 Jython 모듈을 사용 확장할 수 있음

## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구

#### ① SIKULI(시쿨리)

적절한 GUI요소로 마우스·키보드 이벤트를 제공하기 위해 이미지 패턴 사용

##### Java.awt.Rotot

- 시쿨리의 구성요소를 사용자에게 스크린의 사용할 위치로 전달

##### C++ 엔진

- 시쿨리의 핵심 이미지 프로세스 엔진인 openCV 즉, 오픈 소스 컴퓨터 비전을 기반으로 함
- 스크린에 이미지 패턴 검색 담당

## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구

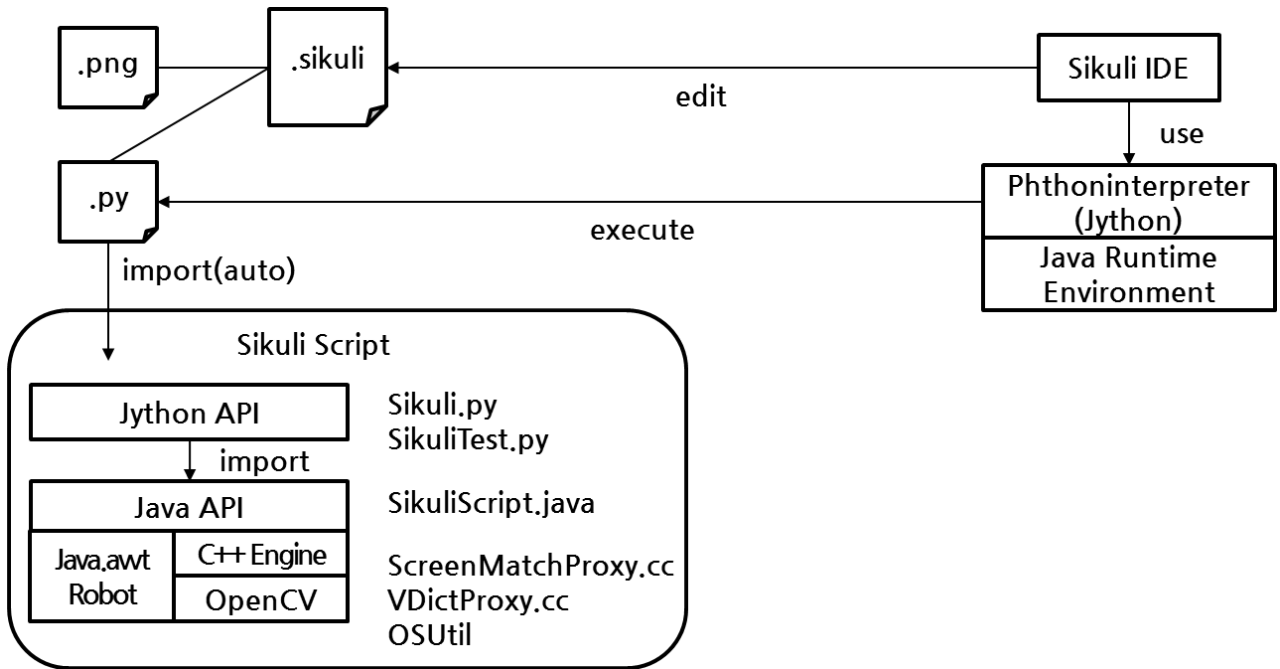
#### 1 SIKULI(시쿨리)

장점	단점
<ul style="list-style-type: none"> <li>• 애플리케이션 코드/설계에 대한 정보가 필요하지 않음</li> <li>• 빠른 자동화</li> <li>• 스크립트가 기존 플랫폼에 의존하지 않음</li> <li>• GUI 테스트 사용</li> <li>• 시쿨리는 일치의 정도를 확인하기 위해 테스트 프로세스 중에 조정할 수 있는 '퍼지로직' 이미지 비교를 제공함</li> </ul>	<ul style="list-style-type: none"> <li>• 크기/위치를 변경하는 경우 이미지를 일치시키는 것이 어려움</li> <li>• VNC/스크립 캐스트를 이용한 바탕 화면에 휴대 전화 화면 전송이 원활하지 않기 때문에 모바일 애플리케이션 성능 시험에 적합하지 않음</li> </ul>

## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구

#### 1 SIKULI(시쿨리)



## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구

#### ② MONKEYRUNNER(Android 용)

##### Monkey Runner 도구

Android 장치를 원격으로 제어하는 API를 제공하는 Android 오픈 소스 프로젝트의 일부

##### MonkeyRunner를 이용

- Android 애플리케이션을 설치, 키를 입력, 사용자 인터페이스의 스크린샷을 이용해 테스트 드라이버의 스크린샷을 저장하는 Python 프로그램을 작성

##### 스크린샷

- 샘플 기준 스크린샷과 비교해 확인할 수 있음



## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구

#### 2 MONKEYRUNNER(Android 용)

Monkey  
Runner

- 일반적으로 도구를 확장하고 그 위에 프레임워크를 작성하는 방법뿐 아니라 기능 테스트, 재테스트에 대한 기능을 제공함

Monkey  
Runner

Monkey  
Device

Monkey  
Image



테스트 사례들을 만들고 실행시키는 사용자 지정 Python 프로그램을 호출 할 수 있는 API 세트를 제공

## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구


#### ② MONKEYRUNNER(Android 용)

장점	단점
<ul style="list-style-type: none"> <li>• 스크린을 바탕화면으로 보낼 필요 없이 실제 장치를 테스트할 목적으로 사용</li> <li>• 오프라인분석에 대한 스크린샷 생성 가능</li> <li>• 스크린샷은 또한 기본 제공 기능을 이용하여 참조 이미지를 자동화할 목적으로 사용하거나 외부 도구를 사용할 수 있음</li> <li>• 플러그인 아키텍처를 사용하여 확장 가능</li> <li>• 동시에 여러 장치를 제어하는데 사용 가능</li> </ul>	<ul style="list-style-type: none"> <li>• 애플리케이션 코드 정보가 필요함</li> <li>• 예를 들어 사용자는 부가 기능과 뷰의 이름을 알아야 함</li> <li>• MonkeyRunner는 하위 수준 API도구로서 기본적으로 제공되는 GUI/IDE가 없음</li> </ul>

## 모바일 테스트 도구와 고려 사항

### 2 모바일 애플리케이션 테스트 자동화 도구

#### 2 MONKEYRUNNER(Android 용)



```
# Import Monkeyrunner Classes
from com.android.monkeyrunner import MonkeyRunner,
MonkeyDevice
# This class is not documented
from com.android.monkeyrunner.easy import EasyMonkeyDevice
# Connect to Android Device
mydevice = MonkeyRunner.waitForConnection()
# Install the Android Application on connected device
device.installPackage('hscTest.apk')
# Package Name
package = 'com.hsc.TestApp'
# set variable with Activity Name
activity = 'com.hsc.TestApp.LandingPage'
# sets the name of the component to start
runComponent = package + '/' + activity
# Runs the component
device.startActivity(component=runComponent)
# Wait for few seconds
MonkeyRunner.sleep(10)
# Create instance of Easy device
easyDevice = EasyMonkeyDevice(device)
```

## 모바일 테스트 도구와 고려 사항

### 3 모바일 애플리케이션 테스트 시 고려 사항

#### 에뮬레이터 (Emulator) 기반 테스트

- 실제 단말이 아니라 가상의 환경에서 모바일 테스트를 하는 것

▶ 일반적으로 모바일 테스트를 할 경우, 에뮬레이터 기반에서 충분히 테스트 후 운영 장비에 배포

#### 에뮬레이터 테스트의 단점

- 실제 장치의 모든 특징과 기능을 에뮬레이터가 그대로 재연해 주지 못하기 때문에 아무리 에뮬레이터에서 충분히 테스트 했다고 해도 실제 단말에서 예기치 못한 오류가 발생

▶ 실제 장치에서 테스트를 할 수 있는 모바일 테스트 장치가 필요

## 모바일 테스트 도구와 고려 사항

### 3 모바일 애플리케이션 테스트 시 고려 사항

간단한 테스트는 에뮬레이터에서 테스트 함

정확성을 요구하는 테스트는 실제 단말 장치에서 테스트하는 것이 바람직함

## 모바일 테스트 도구와 고려 사항

### 3 모바일 애플리케이션 테스트 시 고려 사항



모바일 테스트는 테스트 장비와 테스터가 멀리 떨어져 있는 경우가 존재함

#### 테스트 장비가 원격에 존재하는 경우

- 테스터가 원격으로 그 테스트 장비에 액세스할 수 있어야 함
- 그렇지 못할 경우 테스터는 매우 번거롭게 테스트 장비에 갔다가 와야 함

#### 테스트 자동화 솔루션이 수동 및 자동 테스트를 모두 지원하는지 확인

- 모바일 소스 코드의 작업 변경 확인 및 기타 임시 테스트를 수행하는 경우에는 수동 테스트가 편리함
- 제품 출시가 더욱 빈번해지고 변화 속도가 더욱 빨리질 경우에는 시간과 비용 절감 차원에서 자동 테스트가 필요함
- 테스트 자동화 솔루션이 수동, 자동 모두 지원해야 함

## 모바일 테스트 도구와 고려 사항

### 3 모바일 애플리케이션 테스트 시 고려 사항



모바일 테스트는 테스트 장비와 테스터가 멀리 떨어져 있는 경우가 존재함

#### 모바일 테스트 환경 설정 시 특수한 환경에서 테스트 해야 할 경우

- 다른 환경은 변경하지 않고 전용망 환경만 구성하여 테스트 해야 함
- 특수한 환경을 구성하기 어려워 모바일 테스트 자동화 도구가 네트워크 가상화 환경도 지원한다면 쉽게 구성이 가능함
- 모바일 테스트 자동화 솔루션 구성이 이러한 네트워크 가상화 환경 지원이 가능한지 확인함



## 모바일 테스트 도구와 고려 사항

### 3 모바일 애플리케이션 테스트 시 고려 사항



모바일 테스트는 테스트 장비와 테스터가 멀리 떨어져 있는 경우가 존재함

자동화 솔루션이 사용자환경에서 발생하는 서비스 중단 및 단절 테스트를 지원하는지 확인

- 모바일 장치에서는 애플리케이션 성능과 기능에 영향을 미칠 수 있는 다양한 동작이 수행됨
- 문자 메시지, 이메일, 전화, 배터리 방전, 새로운 앱 수행 등
- 이러한 다양한 서비스 중단 및 장치 단절 기능 테스트가 가능한지 확인

## 모바일 테스트 도구와 고려 사항

### 3 모바일 애플리케이션 테스트 시 고려 사항



모바일 테스트는 테스트 장비와 테스터가 멀리 떨어져 있는 경우가 존재함

#### 모바일 기능, 성능, 보안 테스트까지 수행

- 모바일 애플리케이션에 대한 기능, 성능, 보안 테스트까지 수행해야 출시 가능함

#### 테스트 스크립트는 재사용

- 모바일 애플리케이션 테스트 투자에 대한 ROI를 높이려면 스크립트 재사용 필요

#### 모바일 테스트 도구를 개발 도구에 통합

- 가급적 통합 개발 도구에 모바일 테스트 도구도 통합 관리하는 것이 효과적임

### 01. 모바일 애플리케이션 테스트 특징과 프레임 워크

- 모바일 애플리케이션 테스트 특징
  - ✓ 시뮬레이션 환경에서 테스트
  - ✓ 자원 제약을 고려한 테스트
  - ✓ 빠른 출시 기간으로 Agile 테스트 필요
  - ✓ 다양한 플랫폼(단말) 환경 존재
- 모바일 UI 애플리케이션 테스트 프레임워크
  - ✓ 에피움(Appium) : 네이티브, 하이브리드, 웹 앱 지원
  - ✓ 셀레니움(Selenium) : Restfull API 제공

### 02. 모바일 테스트 도구와 고려 사항

- 모바일 애플리케이션 테스트 자동화 도구
  - ✓ 특정 플랫폼에 종속된 도구(Instrument, MonkeyRunner)
  - ✓ 일반 스크립트 기반 도구(Sikuli, Robotframework)
- 모바일 애플리케이션 테스트 시 고려 사항
  - ✓ 솔루션이 실제 장치 및 에뮬레이터에서 테스트 지원하는지 확인
  - ✓ 테스트 장치에 원격으로 액세스할 수 있는지 확인