

# 모바일 어플리케이션의 테스트 자동화 기술 수준 및 사례 연구

한영근\* · 이현노\*\* · 박창환\*\*\*

\*명지대학교 산업경영공학과 · \*\*어니컴(주) · \*\*\*STA테스팅컨설팅

## A Study on Technological Level and Cases of Testing Automation for Mobile Applications

Young-Geun Han\* · Hyun-Noh Lee\*\* · Chang-Hwan Park\*\*\*

\*Dept. of Industrial & Management Engineering, Myongji University

\*\*Onycom · \*\*\*STA Testing Consulting

### Abstract

According to the generalization of mobile device use in our life, the complexity and diversity of mobile applications are increased, and thus the importance of application test to find faults and errors quickly and easily before release grows. This paper reviews the current automatic testing methodologies for mobile device applications and their element technologies. Also, well-known commercial tools for automatic testing are surveyed and case studies of their recent industry applications in domestic and foreign countries are performed.

**Keywords :** Mobile Applications, Automatic Testing

### 1. 서론

스마트폰의 활용이 보편화됨에 따라, 모바일 어플리케이션의 복잡도와 다양성이 크게 증가하였고 다양한 OS 및 해상도를 가진 모바일 디바이스 및 캐리어(통신망)를 지원하고 있어 품질과 신뢰성이 중요시 되고 있다. 이에 따라 구현될 모바일 어플리케이션의 개발 과정에서 결함이 내재될 가능성도 높아지고 있다.

현재 모바일 어플리케이션은 단순한 오락 서비스에 서부터, 금융관련 서비스와 같이 소프트웨어의 기능 오류가 사용자 개인에게 재산적 손해를 미치는 분야로까지 확대되고 있기 때문에, 개발 결과물의 구현 오류를 찾아내기 위한 테스트의 중요성과 비중이 커지고 있다.

모바일 어플리케이션은 그래픽 인터페이스를 통해 기능을 제공하기 때문에, 기능 테스트를 하기 위해서는 모바일 어플리케이션이 제공하는 화면 메뉴에 대하여

직접 이벤트를 발생시키거나 데이터를 입력하여 테스트를 수행하는 사용자 관점의 테스트가 중요하다.

Mike Cohn은 테스트 피라미드(testing pyramid)에서 UI 테스트는 단위 테스트, 사용자 인수 테스트와 비교하여 대상 콘텐츠의 다양성, 모호성, 복잡성으로 구현 및 유지 보수가 어려움을 설명하였다. 그럼에도 UI 테스트는 모바일 어플리케이션 기술의 발전과 더불어 중요성이 점차 높아지고 있다.[1].

이에 따라 회귀 테스트를 자동화 하는 방안에 대한 사람들의 관심이 많아졌고 다양한 모바일 어플리케이션 자동화 기술을 적용한 테스트 제품이 출시되고 있다.

본 연구의 목적은 최신의 자동화 테스트 기술이 적용된 모바일 어플리케이션 자동화 테스트 도구 제품의 자동화 방식, 요소 기술 그리고 적용 사례를 검토하고 비교함으로써 테스트 자동화 기술의 현황을 파악하는데 있다.

†이 논문은 미래창조과학부 2013년도 모바일융합기술센터구축사업의 과제연구비 지원에 의한 것임.

†Corresponding Author: Young-Geun Han,

Dept. of Industrial & Management Engineering, Myongji University,

Ph: 031-330-6454, E-mail: yghan@mju.ac.kr

Received January 17, 2015; Revision Received March 05, 2015; Accepted March 06, 2015.

## 2. 어플리케이션 테스트의 자동화 개요

### 2.1 어플리케이션 자동화 테스트의 정의

어플리케이션 자동화 테스트란 테스트 과정의 반복적이고 시간 소모적인 작업을 효율적으로 자동 처리하여 자원을 절약하고 인적 오류(human error)를 줄일 수 있는 효과적인 방법을 말한다. 테스트를 수행하는 사람은 단순하고 반복적인 테스트 업무에서 해방되어 보다 생산적이고 창의적인 업무에 집중할 수 있으며 개발자들은 테스트 결과를 빠른 시간 안에 피드백을 받을 수 있어 개발의 위험을 최소화 할 수 있다. 이런 장점으로 인해 테스트 자동화는 소프트웨어 개발에 있어 선택사항이 아닌 필수요소로 인정받고 있다[2].

### 2.2 어플리케이션 자동화 테스트의 필요성

어플리케이션 자동화 테스트는 도구를 활용하여 테스트 시간과 인력 등을 절감하고자 할 때, 또는 수동으로 하기 어려운 가상의 작업이 필요하거나, 도구가 제공하는 체계적인 커버리지 달성을 목적으로 도입하는 경우가 많다. 특히 애자일(agile) 개발 환경의 반복적인 빌드-릴리즈는 개별 릴리즈별로 반복적인 테스트가 요구되므로 테스트 자동화 도구의 활용은 거의 필수라고 할 수 있다. 도구를 구입하거나 자체 개발을 통해 테스트를 자동화할 경우에 고려해야 할 사항으로는 적절한 자동화 대상 프로젝트 선정, 자동화를 테스트 전략과 계획에 포함, 자동화 도구 선정에 신중, 도구에서 요구하는 것에 따라 제품 테스트 스펙을 공식(formal)으로 변경하도록 준비, 자동화 테스트 지원 시설(도구) 개발 계획 고려, 테스트 개발 계획 또는 도구 방식에 맞게 진행하는 것을 고려, 상세한 자동화 테스트 지침 필요, 자동화 도구 및 자동화 테스트 교육에 투자 요구, 도구 전문가 참여, 자동화 도구 관련 프로세스의 정의와 적용 등이 있다[3].

### 2.3 어플리케이션 테스트 자동화의 난제

어플리케이션 자동테스트 기술은 자동화 과정에서 경험하게 되는 아래와 같은 어려움을 해결하는 방향으로 진화 발전하고 있다[4].

- 자동화 스크립팅(scripting) 언어 습득 필요 - 기본적인 프로그래밍 스크립팅 역량이 없으면 적용하기가 곤란.
- 어플리케이션에 대한 코드 수준 구조 파악 필요 - 애플리케이션 구조를 이해해야 하고, 코딩 및 수정하는 작업이 필요.

어플리케이션 구조를 이해해야 하고, 코딩 및 수정하는 작업이 필요.

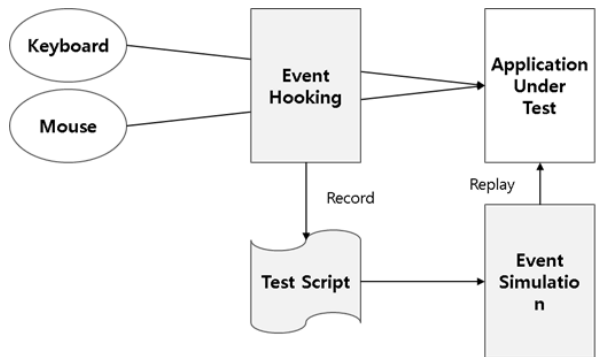
- 유지보수 문제 - 스펙 변경 등으로 애플리케이션의 구성이나 시나리오가 변경될 경우, 직관적으로 찾기 힘들고, 수정이 어려워 유지보수 비용 초래. 시간이 지날수록 그 비용 및 노력은 기하급수적으로 증가.
- 자동화 스크립트 가독성 문제 - 프로그래밍 기술이 없는 사람은 스크립트만 보고는 테스트 시나리오를 이해하기 곤란.
- 통합 관리 문제 - 자동화 집합을 항상 최신 버전으로 유지/관리하기 곤란.

## 3. 어플리케이션 테스트의 자동화 방식

### 3.1 이벤트 시뮬레이션 기반 자동화

이벤트시뮬레이션(event simulation) 기반 자동화 방식은 <Figure 1>과 같이 사용자가 발생시키는 마우스 이벤트와 키보드 이벤트를 녹화하고, 녹화한 이벤트를 재 수행 하는 방식이다. 현재, 대부분의 자동화 도구들이 이 방식을 적용하고 있으며 이 방식을 사용할 경우 자동화 구현이 용이하다는 장점을 가지고 있다. 그러나 녹화 후 재 수행 방식은, 구현대상에 변경이 일어날 경우에는 다시 녹화를 해야 하고, 테스트케이스가 정확히 수행되었는지 수동으로 검증해야 한다. 이 방식의 도구들은 다음과 같은 문제점을 가지고 있다.

첫째, 이 방식은 노동 집약적이다. 효과적인 테스트 케이스를 생성하기 위하여 다양한 시나리오에 의한 테스트를 위해 많은 기록 과정이 필요하며, 테스트 대상 변경 시 같은 작업을 반복해야 한다.



<Figure 1> Automatic testing based on event simulation

또한 녹화시의 사소한 실수가 테스트 수행 시 치명적으로 나타나는 경우가 발생한다.

둘째, 이 방식은 단순히 마우스 이벤트와 키보드 이벤트의 내용만을 녹화할 뿐 테스트 케이스가 어떤 작업이며, 어떤 내용인지에 대한 명세를 따로 기술해야 한다. 만약, 이러한 명세가 없다면 많은 양의 테스트 케이스를 관리하기가 어렵다.

셋째, 이 방식에서 테스트 스크립트를 수정하는 것에 많은 노력이 필요하다. 만약, 테스트 대상의 행동 명세가 변경되면 처음 시스템의 실행 동안 기록된 테스트 스크립트가 의미가 없는 경우가 발생한다. 시스템의 변화를 맞추기 위하여 테스터는 시스템 명세에 맞추어 테스트 스크립트를 수정하거나 다시 테스트 케이스를 녹화해야 한다[5].

테스터의 수동적인 조작을 기록하여 테스트를 자동화하는 방식은 매력적으로 보이지만, 이러한 접근 방식은 대규모의 테스트를 자동화할 때는 적절치 않다. 저장된 스크립트는 각 스크립트마다 특정 데이터와 행위를 선형적으로 표현하고 있다. 따라서 이런 종류의 스크립트는 미처 예상하지 못한 이벤트에 취약할 수 있다[6].

### 3.2 키워드 기반 자동화

키워드(keyword) 기반 테스트 자동화는 대부분의 어플리케이션 유지보수 환경에서 뛰어난 해결책을 제공한다. 그 이유는 테스트 설계 시 비용과 시간을 줄일 수 있을 뿐 아니라, 재사용성이 높고, 유지보수가 쉽기 때문이다.

이 방식에서는 테스트 설계 시 테스트 대상이 되는 어플리케이션의 기능들을 간단한 텍스트로 표현되는 키워드와 이와 관련된 파라미터 값을 사용해서 표현한다. 이렇게 조합해서 만든 키워드 페어(pair)들을 일련의 시퀀스(몇 가지 아이템을 어떤 기준에 따라 공간적 또는 시간적으로 순서를 정하여 배열하는 것)로 구성하여 테스트케이스를 작성한다. 키워드는 테스트 케이스에서 필요할 때마다 사용할 수 있기 때문에, 테스트 설계 및 자동화를 구축할 때 추가 작업 없이 재사용할 수 있고, 기 구축된 테스트 케이스나 자동화 셋(set)에는 영향 없이 키워드를 업데이트 할 수 있는 장점이 있다[7].

다음의 <Table 1>은 간단한 키워드의 예이고, <Table 2>는 더 복잡한 키워드의 예이다.

<Table 1> Example of keyword 1

Object	Action	Data
Textfield (username)	Enter text	User Name

출처:

[Http://en.wikipedia.org/wiki/Keyword-Driven\\_testing](http://en.wikipedia.org/wiki/Keyword-Driven_testing)

<Table 2> Example of keyword 2

Object	Action	Data
Textfield (domain)	Enter text	<domain>

출처:

[Http://en.wikipedia.org/wiki/Keyword-Driven\\_testing](http://en.wikipedia.org/wiki/Keyword-Driven_testing)

스크립트의 재사용과 유지보수의 용이성을 가지고 있는 키워드 기반 방식에도 키워드로 구분되지 않는 플래시 및 정보를 숨겨 키워드로 식별되지 않는 영역에 대한 테스트 자동화가 불가능하다는 단점이 있다. 이렇게 되면 회귀 체크리스트에서 자동화 할 수 있는 영역은 한정되고 결국엔 테스트 자동화 커버리지는 낮아질 수밖에 없다.

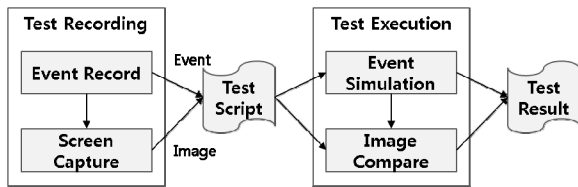
이 방식에서는 스크립트 언어에 익숙하지 않은 테스터라도 키워드를 이용하여 테스트를 정의할 수 있는데, 키워드는 테스트 대상 어플리케이션에 맞게 테일러링 될 수 있다.

### 3.3 이미지 인식 기반 자동화

기존의 이벤트시물레이션 기반 자동화는 구현이 쉬워 GUI 테스트에서 많이 사용되지만 모든 사용자의 키보드/마우스 이벤트의 기록이 필요하고 이에 대한 명세가 필요하다. 그리고 테스트 스크립트를 수정하는 많은 노력이 필요하다. 사용자 이벤트의 기록과 재현에 중점을 두어 테스트 결과에 대한 검증을 사람이 직접 수행해야 한다.

이에 반해 이미지인식 기반 자동화는 <Figure 2>와 같이 컴퓨터 비전을 이용하여 화면에 표시되는 GUI 요소를 하나의 이미지로 가정하고 이미지 내 특정영역을 저장하고 저장된 이미지를 기반으로 화면 내 해당 이미지의 존재 여부를 비교하는 방식으로 테스트를 자동화한다. 스크립트 작성과 인식이 직관적이며, 어플리케이션 변경 시 유지보수가 용이하다. 특히, 화면의 사

이즈가 상대적으로 적으며, 비교적 정형화된 구성을 갖는 모바일 어플리케이션의 경우 테스트 결과의 신뢰성이 상당히 높아 모바일 어플리케이션 자동화 테스트 분야에는 가장 많이 선택되는 방식이다.



<Figure 2> Automatic testing based on image recognition

테스트 결과 검증을 위해 예상 이미지와 실제 이미지를 비교하는 방식을 취하고 있는데, 초기의 이미지인식 기반 테스트 자동화 도구들은 이미지 비교의 속도를 빠르게 하기 위해, 이미지의 컬러패턴을 단순화한 그레이 이미지를 사용하여, 복잡하거나 유사한 이미지와 복잡한 컬러 패턴의 경우 이미지 비교 결과에 오류가 발생하는 문제가 존재하였다.

최근에 인식알고리즘의 발전으로 이러한 단점이 많은 부분 보완 되었으며, 문자열 검증을 위한 OCR모듈을 내장하여 데이터 드리븐 테스트까지 지원하는 모델로 발전하고 있다.

이 방식의 장점은 다음과 같다.

- 화면 구성 오브젝트의 인식 여부와 상관없이 언제나 테스트 가능
- 어떠한 웹과 어플리케이션에서도 테스트 가능
- 화면 구성에 대한 전문적인 지식이 없이도 누구나 손쉽게 테스트 스크립트 작성 가능
- 가상(버추얼) 데스크탑 환경에서도 테스트 가능
- 다양한 모바일 디바이스 환경에서도 동일하게 테스트 가능
- 키보드 보안이 있는 경우에도 문제없이 테스트 가능

이 방식의 단점은 다음과 같다.

- 각각의 마우스나 키보드입력의 이벤트를 입력할 경우, 매번 해당 이미지를 특정하여 주어야 하므로 스크립트 작성시 시간이 많이 소요
- 이미지, 글자폰트, 배경화면이 변경될 경우, 다시 스크립트를 작성
- 동일 이미지가 화면 내 여러 개 존재할 경우, 테스트 진행에 문제 발생
- 검증포인트 설정 시, 이미지로만 검증이 가능
- 검증포인트 설정 시, 문자 비교를 해야 할 경우, OCR 기능이 반드시 필요

### 3.4 객체 인식 기반 자동화

이 방식은 네이티브(native) 어플리케이션이나 웹에서 화면을 구성하고 있는 각각의 오브젝트(텍스트, 이미지, 메뉴, 버튼 등 화면에 표시되는 모든 것)의 정보를 기반으로 화면에 오브젝트의 존재 여부 및 구성값 등을 비교하여 테스트 시나리오를 자동화 하여 실행하는 것이다.

이 방식의 특징과 장점은 다음과 같다.

- 마우스 클릭 또는 키보드 입력만으로 레코딩을 통한 손쉽고 빠른 스크립트 작성이 가능하며 레코딩된 테스트 스크립트를 바로 재생 가능
- 검증 포인트 작성 시, 다양한 종류의 항목(문자, 이미지, 위치 등)을 검증 포인트로 설정이 가능
- 이미지, 폰트, 배경색 등의 변화에 상관없이 오브젝트의 변경이 없다면 스크립트 재사용 가능
- 다양한 브라우저 환경에서도 동일하게 테스트 가능

이 방식의 단점은 다음과 같다.

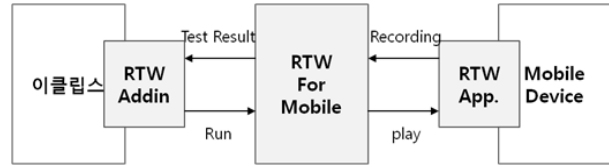
- 화면 구성 오브젝트에 관한 지식이 어느 정도 있어야 재촬영성 높은 스크립트 작성 가능
- 제품에 따라 오브젝트 인식율에 차이가 있으므로 제품 선택이 중요
- 오브젝트가 랜덤하게 바뀌는 화면에서는 regular expression 등의 추가 작업이 필요
- 키보드 보안이 있는 경우, 레코딩이 되지 않아 매뉴얼 코딩이 필요한 경우 발생

### 3.5 자동화 테스트 방식의 비교

어플리케이션 자동화 테스트의 기반 기술인 테스트 방식을 <Table 3>에 요약하여 비교하였다. 각 자동화 테스트 방식은 장단점이 존재하며, 최근의 어플리케이션 자동화 테스트는 각 방식의 장점을 살려 자동화 테스트 적용의 어려움을 해소할 목적으로 한 가지 방식이 아닌 다수의 방식을 지원하는 방향으로 진행되고 있다.

<Table 3> Summary of testing methods

method	feature	adv. & disadvantage
Event Simulation based Testing	Record and execute mouse and keyboard events from user acts.	Test results should be checked manually. Maintenance according to app modification is almost impossible.
Key-word based Testing	Based on understanding app structure, do mapping structural elements to keywords and transmit events to keywords.	Testing is impossible for flash apps and when app structure is not understood. This evolved to object recognition method recently, which recognizes keywords and object information automatically on screen.
Image Recognition based Testing	Save a specific region in image and based on it detect whether or not the image exists on screen.	Its testing domain is the broadest. Script writing time is long because of specifying each image.
Object Recognition based Testing	Detect whether or not objects exist on screen and compare its numeric value, based on each object data of native app or web.	There is a difference in recognition rate according to implementation skill. Recognition is difficult when objects are constructed dynamically. Script reusability is maximized if one has object organization knowledge.

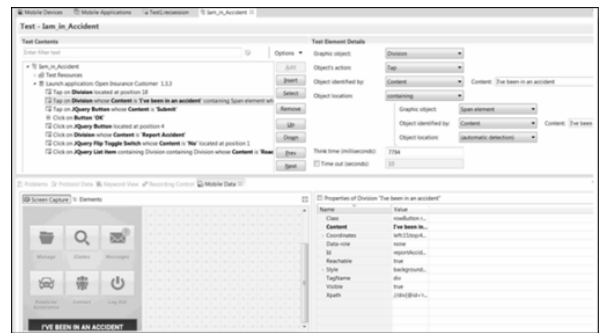


<Figure 4> Configuration of RTW Mobile

이클립스 Addin은 모바일 어플리케이션의 사용자 인터페이스 및 성능을 테스트하고 사용자의 PC 브라우저에서 웹 어플리케이션의 사용자 인터페이스를 테스트하기 위한 클라이언트의 통합 모듈이다. RTWM 클라이언트(PC)는 테스트 케이스를 생성하고, 검증 포인트를 작성하며, 테스트 케이스 실행 리포트를 만든다. RTWM 클라이언트(Device)는 RTW에서 레코딩과 플레이가 필요한 어플리케이션을 업로드하고, 높은 수준의 UI 제스처와 디바이스 이벤트를 캡처하고 모바일 상에서 테스트 결과를 확인할 수 있다.

RTW Mobile의 주요 기능은 다음과 같고, <Figure 5>는 실행화면의 예이다.

- 모바일 어플리케이션 테스트 스크립트 작성
- 모바일 어플리케이션 테스트 스크립트 실행
- 데이터 드리븐 테스트지원(스프레드시트, ODB)
- 실행 단계별 세부 보고서 기능 제공
- 모바일 자체 내에서 스크립트 실행 가능
- 결합 관리 도구(예: RTC 등) 연동을 통한 원 클릭 결합 등록
- 테스트 관리 도구 (예: RQM 등) 연동을 통한 테스트 실행 및 결과 자동 등록



<Figure 5> Screen example of RTW Mobile

#### 4. 어플리케이션 테스트의 자동화 도구

##### 4.1 Rational Test Workbench Mobile

이 제품은 미국의 IBM사에서 2013년 출시한 제품으로서 객체 인식 방식의 모바일 어플리케이션 GUI 테스트 자동화 지원 도구이다. <Figure 4>는 이 도구의 구성을 보여준다.

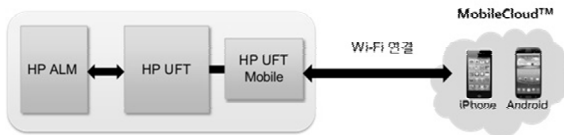
RTW Mobile의 특징과 장점은 다음과 같다.

- Android 및 iOS 지원
- 네이티브, 하이브리드, 웹 어플리케이션 모두 테스트 가능
- Record & Play 방식으로 손쉽게 스크립트 생성 및 실행 가능

- 키워드 방식의 테스트 스크립트 지원으로 마우스 클릭만으로 스크립트 생성과 수정 가능
- 빌트인 된 체크포인트로 간편하게 PASS /FAIL 판단 요소 등록 가능
- Rational Integration Tester와 연동되어 원격 실행, 문제점을 이메일과 모바일로 자동 통지, 문제 해결 조치를 할 수 있도록 구성 가능

### 4.2 Unified Functional Testing Mobile

이 제품은 미국의 Hewlett-Packard사에서 출시한 것으로서, 클라우드 환경을 통한 단말기 제공 서비스와 통합된 모바일 어플리케이션 테스트 자동화 지원 도구이다. <Figure 6>은 이 도구의 구성을 보여준다.



<Figure 6> Configuration of UFT Mobile

HP ALM(Application Lifecycle Management)은 테스트 공정 관리와 테스트 실행 스케줄링 및 리포팅을 담당하고, HP UFT(Unified Functional Testing)는 웹 및 윈도우 어플리케이션 테스트 자동화를 수행하고, HP UFT Mobile은 rooting 또는 jail-breaking 없이 모바일 환경 지원을 위한 VNC 서버 기능을 제공한다. MobileCloud™는 PerfectoMobile사의 모바일 디바이스 제공 클라우드 서비스이다.

UFT Mobile의 주요 기능은 다음과 같고, <Figure 7>은 실행화면의 예이다.

- 모바일 어플리케이션 테스트 스크립트 작성
- 모바일 어플리케이션 테스트 스크립트 실행
- 데이터 드리븐 테스트지원 (스프레드시트, ODBC) 단순 보고서 기능 제공
- 에러복구 기능 제공
  - Email me, SMS me, Call me, Power cycle, Share, Injection audio
- 테스트관리도구 및 성능테스트도구 연계 가능



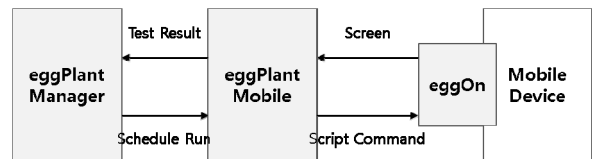
<Figure 7> Screen example of UFT Mobile

UFT Mobile의 특징과 장점은 다음과 같다.

- Mobile 디바이스 제공 클라우드 서비스와 테스트 자동화 제품이 통합된 서비스
- 작성된 스크립트를 다른 모바일 OS에서도 실행이 가능하게 OS별 이미지 라이브러리 제공
- HP UFT에 Add-in 형태로 통합되어 HP 테스트 제품군과 연동

### 4.3 eggPlant Mobile/eggOn

이 제품은 2008년 미국의 TestPlant사에서 출시한 것으로서, VNC 프로토콜 기반의 이미지 인식 및 OCR 인식 기술이 적용된 모바일 어플리케이션 Visual GUI 테스트 자동화 지원 도구이다. <Figure 8>은 이 도구의 구성을 보여준다.



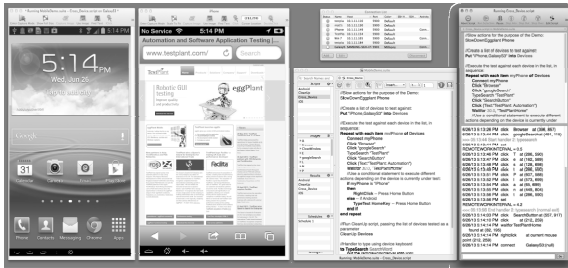
<Figure 8>Configuration of eggPlant Mobile

eggPlant Manager는 테스트 실행 스케줄링 및 리포팅을 담당하고, eggPlant Mobile은 자동화 스크립트 작성과 단독 실행을 수행하며, eggOn은 rooting 또는 jail-breaking없이 모바일 환경을 지원하기 위한 VNC 서버 기능을 수행한다.

eggPlant Mobile의 주요 기능은 다음과 같고, <Figure 9>는 실행화면의 예이다.

- 모바일 어플리케이션 테스트 스크립트 작성
- 모바일 어플리케이션 테스트 스크립트 실행
- 데이터 드리븐 테스트지원 (스프레드시트, ODBC) 단순 보고서 기능 제공
- 에러 복구기능 제공
- 디버깅 툴 제공

- 테스트관리도구 및 성능테스트도구 연계 가능



<Figure 9>Screen example of eggPlant Mobile

eggPlant Mobile의 특징과 장점은 다음과 같다.

- 다양한 모바일 디바이스 테스트 가능(6종: Windows Mobile, Android, iOS, Symbian S60, Nokia SmartPhone, Blackberry)
- 자체 개발한 자연어와 유사한 "SenseTalk" 언어로 사용 편리
- 위치변경과 무관하게 이미지를 인식하는 특허기술 보유
- 이미지 검색 시 원본이미지에서 크기가 변동되어도 검색/비교가 가능한 Auto-Scaling 기능 제공
- 작성된 스크립트를 다른 모바일 OS에서도 실행이 가능하도록 OS별 이미지그룹화기능 제공
- 데이터 드리븐 테스트를 위한 ODBC(open database connectivity) 지원

#### 4.4 TestComplete Mobile

이 제품은 미국의 SmartBeart사에서 1999년 처음 출시한 것으로서, 오브젝트 및 이미지 인식 방식의 모바일 어플리케이션 GUI 테스트 자동화 지원 도구이다. <Figure 10>은 이 도구의 구성을 보여준다.



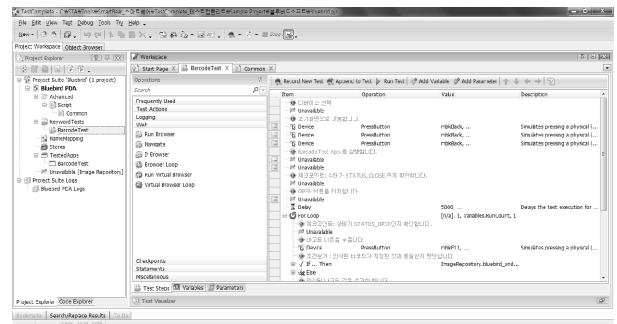
<Figure 10>Configuration of TestComplete

3개의 구성 모듈 중 Desktop모듈은 Windows 기반

의 어플리케이션 테스트를 수행하고, Web모듈은 다양한 브라우저 기반의 웹 테스트를 실행하며, Mobile모듈은 Android 및 iOS 모바일 디바이스들을 테스트한다. TestCompletePlatform은 스크립트의 레코딩과 실행 등을 수행하는 공통 플랫폼이다.

TestComplete Mobile의 주요 기능은 다음과 같고, <Figure 11>은 실행화면의 예이다.

- 모바일 어플리케이션 테스트 스크립트 작성
- 모바일 어플리케이션 테스트 스크립트 실행
- 데이터 드리븐 테스트지원 (스프레드시트, ODBC)
- 사용자 함수 구현을 통한 기능 확장 가능
- 실행 단계별 세부 보고서 기능 제공
- TestExecute를 통한 원격지 테스트
- 결합 관리 도구(예: JIRA 등) 연동을 통한 원 클릭 결합 등록
- 테스트 관리 도구 (예: HP QC, QAComplete 등) 연동을 통한 테스트실행 및 결과 자동등록



<Figure 11>Screen example of TestComplete

TestComplete Mobile의 특징과 장점은 다음과 같다.

- Android 및 iOS 지원
- Android/iOS 공통 개발플랫폼인 Xamarin 지원
- 네이티브, 하이브리드, 웹 어플리케이션 모두 테스트 가능
- 오브젝트 인식 방식과 이미지 인식 방식 모두 지원하여 어떠한 경우에도 테스트 가능
- Record & Play 방식으로 손쉽게 스크립트 생성 및 실행 가능
- 키워드 방식의 테스트 스크립트 지원으로 마우스 클릭만으로 스크립트 생성과 수정 가능
- 키워드 방식의 테스트 스크립트는 다양한 스크립트 언어(Visual Basic, Java, C# 등)로 변환하여 보다 세부적으로 작성 가능
- 빌트인된 체크포인트로 간편하게 PASS/FAIL 판단 요소 등록 가능
- 웹의 경우, 다양한 모바일 디바이스를 가상화(에이전

트 및 해상도 지정)하여 테스트 가능

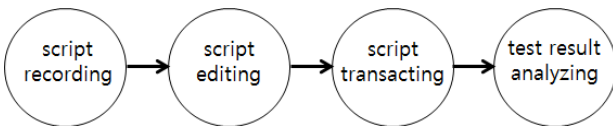
- 두 대 이상의 모바일 디바이스에 대해 연동테스트 가능
- 모바일 디바이스와 웹 또는 PC용 소프트웨어에 대해 연동 테스트 가능
- 원격 에이전트를 통하여 동시에 여러 곳에 있는 모바일 디바이스에 대해 테스트 가능

### 5. 자동화 테스트 기술 요소의 정의

어플리케이션 자동화 테스트 기술은 활용의 목적과 범위에 따라 차이가 존재한다. 본 연구에서는 분석 대상 어플리케이션 자동화 테스트 제품의 구성을 다음의 6가지로 정리하고 이를 기준으로 기술 요소를 분석하였다.

#### 5.1 어플리케이션 자동화 테스트 공통 기술

하나의 테스트 케이스를 기준으로 어플리케이션 자동화 테스트의 적용 절차는 자동화 방식 또는 자동화 제품의 구분 없이 <Figure 12>와 같은 절차로 실행된다.

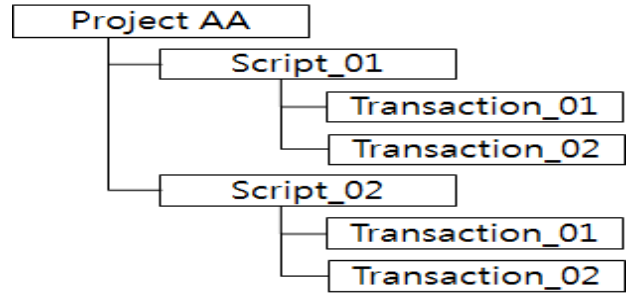


<Figure 12> Procedure of automatic testing

테스트 도구에 따라 다소간의 차이가 존재하나 각 단계별로 요구되는 핵심 기능은 다음과 같다.

##### (1) 스크립트 레코딩

- 프로젝트 생성  
자동화 테스트는 다수의 테스트 스크립트로 구성되며, 이를 통합하여 관리하기 위해 프로젝트로 그룹화하는 것이 필요하며, 일부 도구의 경우 프로젝트 상위에 프로젝트 그룹(또는 프로그램)이라는 상위의 구성 단위를 생성하기도 한다.
- 스크립트 생성  
테스트 레코딩의 결과가 기록될 스크립트 파일을 레코딩 전 생성한다. <Figure 13>은 스크립트의 체계를 보여준다.
- 레코드 방식 설정  
대부분의 어플리케이션 테스트 자동화 도구들은 다수의 레코딩 방식(이벤트, 이미지, 객체)을 지원하며, 레코딩 전 레코딩 방식을 설정하고, 레코딩 중 어플리케이션과 테스트 도구의 지원 기능에 따라 변경이 가능하다.



<Figure 13> Script hierarchy of automatic test

- 트랜잭션 구분  
레코딩 시 업무적/의미적으로 구별 되는 단위를 트랜잭션으로 구분 하며, 이는 실행 시 성능(응답속도)측정의 단위로 사용 된다.
- 테스트 대상 선택  
자동화 테스트 대상 어플리케이션을 사전에 구동시키는 방식과 레코딩 시작 후 구동 시키는 방식이 있으며, 사전에 구동 시키는 방식은 CLI 명령을 통해 자동 실행을 지원하며, 이는 테스트의 초기조건을 일관 되게 가져가는데 유리한 점이 있다.
- 체크 포인트 삽입  
체크 포인트는 테스트 결과를 검증하기 위해 사용되며, 스크립트의 실행 중단, 시스템 오류에 의한 응답시간 지연과 같은 기본적인(default) 체크 포인트 외에 문자열, 이미지, 파일, 데이터베이스, 객체 상태 등 도구에 따라 다양한 체크 포인트가 필요하다. 체크 포인트의 다양성이 어플리케이션 테스트 자동화 도구의 유용성의 수준을 결정한다.
- (2) 스크립트 편집
  - Statements  
대부분의 테스트 자동화 도구들은 지원 스크립트 언어에 따르는 작업명령을 지원한다.
  - Input Data 파라미터화  
데이터-드리븐 테스트의 전제가 되는 기능이며, 다수의 스크립트를 이용하여 실행 시나리오 구성 시 이전 단계의 결과값을 다음 단계의 입력값으로 사용하기 위해 필요하다.
  - 체크 포인트 파라미터화  
체크 포인트의 value가 동적으로 변화하는 경우 파라미터와 로직을 사용하여 가변적인 체크 포인트를 구성할 수 있다.
  - 외부 함수 호출  
스크립트에 다양한 기능을 부여하기 위해 사전 정의된 함수 또는 사용자 정의 함수를 사용 할 수 있도록 한다.
  - 외부 스크립트 호출



스크립트 내에서 다른 스크립트를 호출하여 실행하는 기능으로 주로, 개별적으로 작성된 스크립트를 시나리오로 구성하여 통합테스트에 사용 시 활용 된다.

· 디버그 기능 지원

테스트 스크립트 디버깅을 위해 사용되는 기능으로 일반적인 컴파일러 수준의 기능을 제공한다.

(3) 스크립트 실행

· 테스트 시나리오 설정

자동화 테스트 실행 대상 스크립트, 실행 횟수, 중단 조건 등을 선택한다.

· 임계치 설정

응답 속도, 오류 발생 횟수 등의 한계값을 사전에 설정한다.

· 로깅 설정

동일 스크립트를 여러 차례 실행하게 되며, 이미지, 텍스트 등이 테스트 결과를 기록하게 된다. 필요에 따라 테스트 로그를 선택하여 가독성을 높이고 디스크를 효율적으로 사용한다.

· 리커버리 스크립트 설정

스크립트 실행 시 오류가 발생하여 더 이상의 실행이 불가할 경우 테스트 초기상태로 복원하는 스크립트를 설정 한다. 단순 실행의 경우 설정하지 않아도 무방하나, 다수 스크립트/반복 실행의 경우에는 반드시 필요하게 된다.

(4) 테스트 결과 분석

· 리포트 생성

로깅 설정에 따른 데이터로 리포트를 제공한다. 전체 테스트 결과 요약, 스크립트/트랜잭션 단위 테스트 결과, 개별 체크포인트 결과를 트리 형태로 구성하는 것이 일반적이다.

## 5.2 어플리케이션 자동화 테스트 확장 기술

(1) 모바일 어플리케이션 테스트 지원 기술

모바일 어플리케이션은 <Table 4>와 같이 일반적인 소프트웨어의 테스트 환경과 다르게 디바이스와 구성의 조합이 매우 다양하고, 테스트 대상 기능과 네트워크의 고려사항이 많다. 이와 같은 테스트 환경을 지원하기 위한 다음과 같은 추가기능을 필요로 한다.

· 디바이스 조회 및 선택 기능

클라이언트 기반 도구에서는 PC에 연결된 디바이스를 자동으로 선택하게 되지만, 클라우드 기반으로 다수의

디바이스를 제공하는 도구의 경우 조건(모바일 디바이스, OS Version, 로케이션, 통신유형 등)에 따라 디바이스를 검색하고 사용 예약 및 선택하는 기능이 필수적으로 필요하다.

<Table 4> Mobile testing environment

Environment	Characteristics
Device	<ul style="list-style-type: none"> <li>Various mobile OS &amp; versions : iOS, Android, Blackberry, Windows Phone, Symbian, etc.</li> <li>Various spec. : screen resolutions, memories, etc.</li> <li>Various devices : hundreds of manufacturers, thousands of models</li> </ul>
Function	<ul style="list-style-type: none"> <li>Mobile device native functions : camera, GPS, voice, sensors, etc.</li> </ul>
Network	<ul style="list-style-type: none"> <li>Several carriers</li> <li>Various performance, network speed</li> <li>Network offline mode</li> <li>Network switching</li> </ul>

· 모바일 OS(안드로이드, iOS) 기본 기능 제어

Gesture, 하드웨어 버튼, 위젯, 스와이프, 스크롤등의 기능 제어가 필요하고, View, Activity, Dialog, Menu, Context Menu 등을 제공하고, Native 기능 제어가 요구된다.

· 모바일 테스트 추가 기능

패키지를 설치하고 제거할 수 있어야 하며, Email, SMS, Call 수발신 기능 제공으로 다양한 테스트 상태 구성(예: 모바일 어플리케이션 실행 중 SMS 수신)한다.

· 테스트 스크립트 재사용 기능

이미지 인식 기반 자동화 방식에서는 화면 해상도에 차이가 있는 디바이스를 하나의 스크립트로 테스트하기 위해 auto-scaling기능이 필요하다. 하나의 스크립트로 모든 디바이스에서 동일한 테스트 결과를 얻을 수 있도록, 디바이스간 대상 아이콘, 이미지들을 동일한 타겟으로 그룹화하는 image collection 기능이 필요하다.

(2) 회귀 테스트 지원 기술

어플리케이션 자동화 테스트의 기본적인 기능으로 추가적 기술요소를 필요로 하지 않는다.

(3) 데이터 드리븐 테스트 지원 기술

다량의 데이터를 교체하며 반복적으로 테스트를 수행하여 데이터 및 어플리케이션의 신뢰성을 확인 하는 데이터 드리븐 테스트를 자동화하기 위한 추가 기능은 다음과 같다.

- Input Data 파라미터화

데이터 드리븐 테스트의 전제가 되는 기능이며, 다수의 스크립트를 이용하여 실행 시나리오 구성 시 이전 단계의 결과값을 다음 단계의 입력값으로 사용하기 위해 필요하다.

- 스프레드시트 지원

CVS 또는 Excel 파일로 데이터를 저장하고, input data 파라미터와 맵핑하여 실행하는 기능은 거의 모든 테스트 자동화 도구에서 지원하고 있다.

- ODBC 지원

스프레드시트 파일이 아닌 ODBC 지원을 통해 데이터 베이스의 데이터를 조회하여 테스트 입력값으로 사용하는 기능으로 일부 고급 테스트 자동화 도구에서 지원하고 있다.

- 데이터 생성 기능

데이터를 입력이 아닌, 데이터 타입에 따라 sequence/random하게 데이터를 생성하여 테스트 입력값으로 사용하는 기능이 필요하다.

#### (4) 24X365 모니터링 지원 기술

- 스케줄링

반복적인 실행을 위해 실행 대상 및 실행 주기를 설정하는 기능이다.

- 임계치 설정

응답 속도, 오류 발생 횟수 등의 한계값을 사전에 설정한다.

- 데시보드

테스트 현황 및 트렌드를 한눈에 보여주는 데시보드 기능으로 모니터링을 지원하는 자동화 테스트 도구는 모두 지원하고 있다.

- 결함 격리

발견된 결함 재연을 위해 결함이 발생한 트랜잭션을 레코딩하여 보여주는 기능(트랜잭션 실행 시작 시 레코딩을 시작하여 결함이 발생하면, 저장하고 결함이 발생하지 않으면 삭제함)으로 일부 고급 테스트 지원 도구에서 제공한다.

- 알람/노티스

설정된 조건을 충족 시 UMS(Unified Messaging Service)를 통해 알람 제공하고, 설정 시 결함 내역 및 테스트 결과 파일을 메일로 전송한다.

#### (5) 매뉴얼 테스트 지원 기술

- 매뉴얼 테스트 로그 리포팅 기능

수동으로 테스트 진행 시 이벤트가 발생한 시점의 키워드 및 Screen Image를 로그로 기록하고 리포팅하는 기능으로 사용자가 탐색적 테스트 진행 시 과정/결과 기록 및 결함 재연 시나리오로 사용된다.

## 6. 모바일 어플리케이션 테스트 자동화의 국내외 사례

### 6.1 안드로이드 제품의 스트레스 테스트

모바일 제품에 들어가는 반도체 제품을 생산하는 글로벌 메모리 반도체 회사에서, 자사의 메모리 제품을 사용한 모바일 단말기가 다양한 사용 환경에서 문제가 발생하지 않는지 테스트하기를 원하였다. 이에 따라 손쉽게 사용할 수 있고, 스크립트 재활용성이 높아야 하며, 동시에 여러 대의 제품을 테스트 할 수 있어야 하는 선정 기준에 의해 IBM Rational Test Workbench Mobile을 적용하였다.

그 결과로 다양한 어플리케이션이 반복적으로 실행하고 종료될 수 있도록 스크립트를 작성하고 여러 개의 모바일 디바이스를 연결하여 동일한 테스트 스크립트가 최소 하루 이상 지속될 수 있도록 자동화 테스트를 구현하여 문제가 생기면 자동으로 이메일로 통보할 수 있도록 하였다. 이러한 자동화를 통하여 테스트 리소스를 대폭 줄일 수 있었다.

### 6.2 인터넷뱅킹 어플리케이션 테스트

국내의 한 은행에서 모바일 디바이스 다양화에 따른 디바이스 관리 문제, 즉 단종된 디바이스에서 발생하는 인터넷뱅킹 어플리케이션 오류로 VOC 증가 문제와 선별된 최신 모바일 디바이스에서 만 테스트 후 릴리즈로 품질 저하 문제 등을 인식하였다.

따라서 클라우드 서비스를 통해 다양한 디바이스들과 다양한 국내/국외 통신사 및 네트워크 환경을 지원하고, 모바일 보안모듈 설치 시에도 테스트 자동화가 가능하며, 한번 작성된 테스트 스크립트로 모든 디바이스가 테스트가 가능한 조건에 의해 HP Unified Functional Testing Mobile을 도입하였다.

1주간 1명의 테스트 엔지니어가 모바일 뱅킹 어플리케이션에 대한 테스트 자동화 스크립트를 작성하게 되었다. 보안 모듈 적용 영역은 이미지 기반 인식방식을 사용했고, 보안카드 입력을 자동화하였으며, 보안키보드 입력용 API를 개발하였다. HP ALM을 사용하여 테스트 스크립트와 결과를 통합 관리하게 되었다.

### 6.3 뉴스제공 어플리케이션 테스트

일일 평균 전세계 220만명의 독자층을 보유한 영국

의 일간신문사에서 디지털 구독이 15%이상 지속적으로 증가하고 새로운 모바일 어플리케이션의 출시로 end-to-end 테스트 시간이 증가하게 되었는데, 5명의 테스터가 종일 전담 테스트하고 있었으나 한계에 부딪혔다. 이에 따라 functional and regression 테스트에 도구를 도입하여 담당자들의 업무 과부하를 해소하고 결함으로 인한 뉴스지연을 방지하고자 했다.

iOS에서 HTML5 콘텐츠와 네이티브 웹 어플리케이션의 테스트가 가능하고, 테스트 도구의 사용이 용이하며, 실제 디바이스 상에서 데이터 드리븐 테스트가 수행될 수 있는 도구로서 eggPlant Mobile/eggOn을 선정하게 되었다.

사용후 3개월 안에 1명의 테스트 엔지니어가 모바일 어플리케이션에 대한 테스트를 90% 완료하여 목표를 달성하였다. 또한 iOS와 안드로이드 기기들의 모든 테스트 스크립트에 대해 3개월 만에 작성을 완료하였고, eggPlant 플러그인을 사용하여 Jenkins와의 통합도 완료하였다.

적용 효과로는, 코어 테스트 케이스에 대해 매뉴얼 테스트 시간이 6~8시간이었으나 테스트 도구를 사용하여 10분으로 단축되었다. iPad 스크립트를 iPad2와 iPad3에서 재사용이 가능했고, 24시간 반복 실행하여 모바일 웹 어플리케이션의 정상 가동 여부를 모니터링할 수 있었다. 또한, 매뉴얼 테스트에서 할 수 없었던 다양한 기기들에 대한 예외케이스도 테스트를 시작할 수 있었다.

## 6.4 안드로이드 기반 산업용 PDA 테스트

국내와 해외 시장에서 다양한 제품으로 시장 영향력이 높은 산업용 모바일 제품을 생산하는 기업이 있다. 이 기업의 생산구조가 고객 요청에 맞추는 다품종 소량 생산이어서 그 만큼 테스트해야 할 대상이 많아서, 적은 인원으로 반복적인 테스트를 진행함에 따라 업무 피로도가 증가하여 자동화를 통하여 업무 효율을 향상시키는 방안을 고민하게 되었다. 이 기업의 요구사항은 손쉽게 사용할 수 있어야 하고 스크립트의 재활용성이 우수한 도구였는데, 검토 결과 TestComplete Mobile를 채택하였다.

시간이 많이 걸리고 매뉴얼 테스트가 반복적으로 이루어지는 테스트에 대해 먼저 자동화 테스트를 도입하였는데, 그중의 하나는 안드로이드 기반 PDA 제품의 바코드 리더 인식 테스트를 자동화하여 인식률을 측정하는 것이었다. 또한 제품의 배터리 사용 시간 측정을 위한 테스트에도 자동화를 적용하여, 매뉴얼 테스트 시

테스터가 수 시간 동안 다양한 테스트 시나리오를 반복하여야 했으나, 자동으로 테스트를 진행하고 다수의 기기에 대해 평균 사용시간을 측정 가능하게 됨에 따라 더 정확한 사용시간을 얻을 수 있었다.

## 6.5 전자지갑 어플리케이션 테스트

1984년 설립된 국내 통신서비스 제공회사에서 전자지갑 수요 증가와 신용카드 등록 및 서명 서비스에 따라 30여개 모바일 디바이스에 대해 1개월 이내에 테스트를 완료해야 했다. 10명의 외주 테스터가 전일 테스트하고 있으나 납기 한계에 부딪히게 되었다. 따라서 모바일 디바이스 제공 클라우드 서비스를 통한 테스트 대상 디바이스를 확대하고, 해외 제공 모바일 어플리케이션의 원격 테스트가 필요하게 되었다.

3G 폰의 어플리케이션 테스트가 가능하고, Test Tool의 사용이 용이하고, 실제 디바이스 상에서 데이터 드라이브가 수행될 수 있으며, 클라우드 서비스를 통해 다양한 모바일 디바이스를 제공하는 요구사항에 따라 DeviceAnywhere를 사용하게 되었다.

이것을 통해 전자지갑의 추가 기능인 신용카드 등록 및 전자서명 등록을 테스트 하였으며, 전자지갑 기본 기능에 대해 회귀 테스트를 수행하였다. 모바일 어플리케이션의 90%를 1개월간 자동화하여 테스트 수행했고, 클라우드 서비스를 통해 홍콩 캐리어(통신망)를 이용한 원격 테스트도 진행하였다.

적용 결과로 코어 테스트 케이스에 대해 매뉴얼 테스트 시간이 7~8시간이었으나 테스트 도구를 사용하여 1시간으로 단축되었다. 동일한 테스트 스크립트를 사용하여 30개의 모바일 디바이스 테스트가 가능했다. 24시간 반복 실행하여 모바일 웹 어플리케이션의 정상 가동 여부를 모니터링하며 결함을 등록하여 자동으로 리포팅을 할 수 있었다. 또한 매뉴얼 테스트에서 할 수 없었던 다양한 기기들에 대한 예외케이스도 테스트를 시작할 수 있었다.

## 7. 결론

본 논문에서는 테스트 자동화의 필요성을 달성하고 어려움을 극복하기 위한 어플리케이션 자동화 방식과 최신 기술이 적용된 어플리케이션 자동화 테스트 제품(모바일 어플리케이션 테스트 자동화 제품 중심)을 조사하고, 활용 사례를 검토하였다.

모바일 어플리케이션의 수가 증가하고 그 중요성도 증대하면서, 그리고 모바일 디바이스 환경의 다양성에

기인하여 모바일 어플리케이션 자동화 테스트는 선택이 아닌 필수로 인식되고 있으나, 기술적·환경적 한계로 인해 실제 적용에는 제약성을 가지고 있다. 이를 해결하기 위해 어플리케이션 자동화 테스트는 최근에 다음과 같은 추세를 보이고 있다.

이벤트시뮬레이션 기반, 키워드 기반, 이미지 인식 기반, 객체인식 기반 방식을 혼용하여 테스트 자동화 방식의 한계에 따른 제약사항을 극복하는 방향으로 진행되고 있다.

어플리케이션 자동화 테스트 기술 요소 측면에서는 공통 기능의 안정성을 기반으로 다수의 지원 기술을 추가하여 자동화의 범위와 활용성을 높이고 있다.

## 8. References

- [1] Cohn, M., "Succeeding With Agile: Software Development Using Scrum," Addison-Wesley Professional, 2009. 11.
- [2] Yu, M.S., "Case of Using Open Source for Testing Automation," Testers Insight, Vol. 2, 2009.
- [3] Kwon, W.I., "Software Testing Is a Nation's Capacity," Testers Insight, Vol. 4, 2009.
- [4] Hwang, Y.S., et al., "A Keyword-based UI Test Framework for Web Services. " Journal of KISS : Software and Applications. Vol. 38, No. 12, 2011: 657-662.
- [5] Chun, B.Y., "A Design and Implementation of Unit Testing Tool for Web Application Programs, " Master's Thesis, Seoul National University of Science and Technology, 2010.
- [6] Kwon, W.I., et al., "Software Testing Practices for Developers 3ed," STA, 2012.
- [7] Johnson, W.D., "Software Quality Tip: The Benefits of Keyword-based Software Test Automation," SearchSoftwareQuality.com.
- [8] HP Products Document and Manual, 2013
- [9] eggPlant Mobile/eggOn Products Document and Manual, 2014.
- [10] TestComplete Mobile Products Document and Manual, 2014.
- [11] <http://www-03.ibm.com>
- [12] <http://www.testplant.com>
- [13] <http://www.smartbear.com>

## 저자 소개

### 한영근



현 명지대학교 산업경영공학과 교수.

주소: 경기도 용인시 처인구 명지로 116 명지대학교 공과대학 산업경영공학과

### 이현노



현 어니컴(주) 수석연구원

주소: 경기도 용인시 기흥구 영덕동 1029

### 박창환



현 (주)STA테스팅컨설팅 수석 컨설턴트

주소: 서울시 강남구 선릉로 100길 삼성빌딩