

모바일 어플리케이션의 원격 시험 서비스에 대한 표준화 연구

한영근* · 전준호** · 이현노**

*명지대학교 산업경영공학과 · **어니컴(주)

Standardization of Remote Testing Service for Mobile Applications

Young-Geun Han* · Jun-Ho Chun** · Hyun-Noh Lee**

*Dept. of Industrial & Management Engineering, Myongji University

**Onycom

Abstract

The objective of this study is to provide mobile application developers with a standard testing system and service method. By using this testing service, the developers can test the functional performance and quality of their applications from remote sites through Internet. This service standard helps to test applications quickly, easily, and cheaply through the intuitional scenario graphical environment, and becomes a base for constructing a testing system which can be shared by developers, users, and the third parties.

Keywords : Moblie Application Testing, Remote Testing

1. 서론

스마트폰과 같은 모바일 기기의 활용이 보편화됨에 따라, 모바일 어플리케이션의 복잡도와 다양성이 증가하고 있다. 이들 어플리케이션들은 상이한 운영체계를 위해 개발되고 있고, 다양한 해상도와 같은 여러 하드웨어 사양을 가진 모바일 디바이스와 캐리어(통신망)를 지원하고 있어서, 다양한 플랫폼 상에서의 품질과 신뢰성의 확보 문제가 중요시 되고 있다. 이에 따라 구현될 모바일 어플리케이션의 개발 과정에서 결함이 내재될 가능성도 높아지고 있다.

현재 모바일 어플리케이션은 단순한 오락 서비스로부터, 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[1].

이러한 상황에서 소프트웨어 개발자들은 이들 모바일 단말기에 포함된 하드웨어 리소스를 활용하기 위한 수많은 어플리케이션을 개발할 수 있는 자유도가 높아진 동시에, 다양한 하드웨어 및 플랫폼에서 무리 없이 실행되는 어플리케이션을 개발하고 테스트해야 하는 부담을 안게 되었다. 따라서 이론적으로는, 고객들이 사용할 다양한 단말기를 구입 또는 임대하여 개발하는 어플리케이션의 시험이 진행되어야 한다.

현재 영세한 수준의 어플리케이션 개발자들이 대다수인 상황에서 그들이 모바일 단말기를 전부 구입하는데에는 금전적인 부담이 따르기 때문에, 다양한 모바일 단말기를 갖춘 웹기반 시험 서비스의 제공자가 있다면 개발자들이 시간과 공간의 제약 없이 온라인 상에서 구현할 어플리케이션을 쉽고 빠르게 테스트할 수 있을 것이다.

† 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.

따라서, 본 연구에서는 어플리케이션 개발자들이 동시에 다양한 하드웨어 및 플랫폼의 환경에서 어플리케이션을 시험할 수 있는 웹기반 서비스 환경의 표준을 제시하며, 다수의 개발자들이 네트워크망을 통해 다수의 모바일단말기 테스트 자원을 공유할 수 있는 표준적인 서비스 방법과 그 시스템에 대해서 제안하고자 한다.

2. 모바일 어플리케이션의 개요

2.1 모바일 어플리케이션의 특성

사용자가 모바일 어플리케이션에 대해서 요구하는 품질 수준에 맞추기 위해서는, 방대한 성능 시험의 문제를 거쳐야 한다. 이 문제는 단순한 데스크탑 환경에서의 시험을 모방하는 것 이상으로 모바일 장치 고유의 다음과 같은 특성들이 존재한다[2].

- 장치의 다양성 : 스마트폰, 태블릿PC 및 이와 유사한 다양한 장치로부터 접속이 이루어질 수 있다. 또한 Android, iOS, Windows Mobile 등의 OS 플랫폼과 이들의 각 버전 또한 다양하다. 따라서 어플리케이션 개발자는 다양한 유형의 장치에서의 시험을 지원하는지 확인해야 한다.

- 업그레이드의 신속성 : 모바일 어플리케이션의 업데이트된 버전이 매우 빠른 속도로 나오고 있고, 급격한 개발 속도로 인해 품질시험 업무에 과부하에 걸릴 수 있고, 결국 어플리케이션 품질을 위협할 수 있다.

- 기본 어플리케이션과 브라우저 기반 애플리케이션의 혼재 : 기본 어플리케이션은 사용자의 장치에 상주하는 어플리케이션, 브라우저 기반 접근방식은 수정된 버전의 브라우저를 사용하여 어플리케이션에 온라인으로 액세스한다. 따라서 시험을 위해서는 이 두 유형의 모바일 어플리케이션을 모니터링할 수 있는 적절한 솔루션을 갖추어야 한다.

- 무선 흡의 다양성 : 최종 사용자는 3G, 4G, LTE 연결과 같은 다양한 대역폭으로 구성된 여러 종류의 네트워크를 사용할 수 있으므로, 이들을 효율적으로 에뮬레이트할 수 있어야 한다. 네트워크 지연, 패킷 손실, 연결 누락 등의 네트워크 상태도 예측할 수 없기 때문에 공통적인 네트워크 상태를 혼합하여 시험을 수행할 수 있어야 한다. 따라서 테스트 시 다양한 무선 흡을 고려하고 이러한 요소가 시스템에 어떤 영향을 미칠 수 있는지 확인해야 한다.

2.2 모바일 어플리케이션의 시험 요구사항

전술한 모바일 어플리케이션의 특징을 고려하여 이들을 효과적으로 시험하려면, 다음 기능을 적절히 수행해야 한다.

- 여러 장치를 지원할 수 있는 기능
- 원격지의 다양한 장소에서 네트워크를 통해 시험을 지원하는 기능
- 브라우저 기반 어플리케이션 및 기본 어플리케이션을 지원할 수 있는 기능
- 사용 편의성 및 빠른 턴어라운드 시간

현재 소프트웨어 테스트 시스템으로서 다양한 도구들이 제시되어 있는데, 대부분은 테스트 스크립트를 작성하고 실행하여 그 결과를 도출하는 형식을 취하고 있다. 그러나 이러한 방식은 사용자가 직접 문법에 맞는 테스트 스크립트를 작성하여야 하는 번거로움이 있고, 테스트 스크립트를 해석하고 분석하여야 하므로 유지보수에 어려움이 있을 뿐 아니라, 테스트 스크립트가 직관적이지 않아 누구나 용이하게 전체 테스트 시나리오를 파악하고 활용하지 못하는 단점이 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[3]. 그럼에도 UI 테스트는 모바일 어플리케이션 기술의 발전과 더불어 중요성이 점차 높아지고 있다.

3. 시험 자원 공유 시스템의 표준 제안

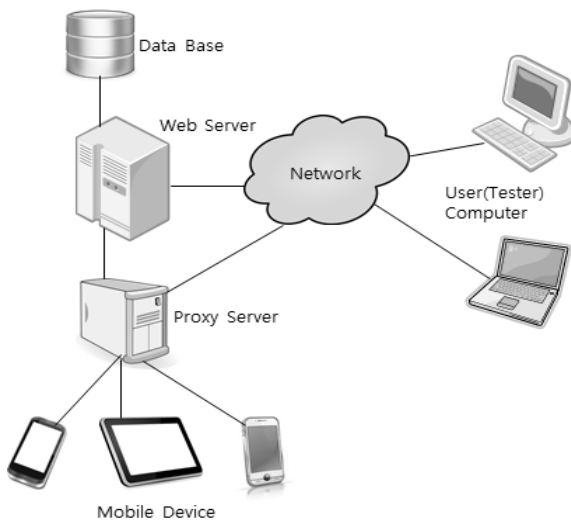
3.1 개요

본 서비스 시스템은 모바일 어플리케이션의 개발자들이 다양한 디바이스를 구입하지 않고도 개발된 어플리케이션이 적절한 기능과 성능을 보유하고 있는지 네트워크망을 통해 원격지에서 품질과 기능을 평가하기 위한 것이다. 즉, 개발자들이 동시에 다양한 하드웨어 및 플랫폼의 환경에서 어플리케이션을 시험할 수 있는 웹기반 서비스 표준을 제공하며, 다수의 개발자들이 네

트위크망을 통해 다수의 모바일단말기 시험 자원을 공유할 수 있는 방법과 그 시스템을 제안한다.

3.2 시험 자원 공유 시스템의 구성

<Figure 1>과 같이 네트워크를 통해 개발자(시험자)가 사용하는 적어도 하나 이상의 사용자단말기와 이들이 접속하는 서버 측 시스템으로 구성되어 있으며, 서버측 시스템은 웹서버, 정보를 저장하는 데이터베이스, 웹서버 및 사용자단말기와 통신하는 프록시서버, 및 프록시서버에 통신 연결된 적어도 하나 이상의 모바일단말기를 포함한다.



<Figure 1> Configuration of testing systems

3.2.1. 웹서버

웹서버는 통상의 정보처리시스템으로서 기기정보관리부와 클라이언트통신부를 포함한다. 이 두 부분은 각각 물리적으로 구분된 시스템으로서 서로 데이터통신을 하도록 구현되어 웹서버가 일군의 시스템으로서 구현될 수도 있고, 하나의 물리적 시스템 내에서 실행되는 각각 개념적인 모듈 또는 프로세스일 수도 있다. 웹서버는 적어도 하나 이상의 사용자단말기로부터 웹접속을 수신하므로 통신부하를 분산하기 위한 로드밸런싱 서버를 포함할 수 있다.

기기정보관리부는 프록시서버의 제어부와 통신하여 프록시서버에 연결되어 있는 각 모바일단말기의 기기 정보 및 기기의 상태정보를 수신하는데, 이 때 HTTP 프로토콜을 이용하여 데이터를 송수신할 수 있다.

클라이언트통신부는 사용자단말기로부터 웹 접속을 받아들이고 웹 서비스를 제공하는 구성요소이므로 통상의 웹데몬 등으로 구현될 수 있다.

3.2.2. 프록시서버

프록시서버는 통상의 정보처리시스템으로서 제어부, 표시관리부, 설치관리부, 이벤트관리부로 구성된다. 이들 구성요소는 물리적으로 구분된 시스템으로서 서로 데이터통신을 하도록 구현되어 프록시서버가 일군의 시스템으로서 구현될 수도 있고, 하나의 물리적 시스템 내에서 실행되는 각각 개념적인 모듈 또는 프로세스일 수도 있다. 또한, 프록시서버는 사용자단말기로부터 네트워크 통신을 통해 접속되므로 통신부하를 분산하기 위한 로드밸런싱 서버를 포함하는 일군의 시스템으로 구현될 수 있다.

프록시서버에는 각각의 특징을 갖는 모바일 단말기들이 통신으로 연결되어 있다. 통신연결이라 함은 범용직렬 버스(Universal Serial Bus) 인터페이스에 의한 연결일 수도 있고, 통상의 네트워크망을 통한 연결일 수도 있다. 각 모바일단말기들은 각기 다른 운영체제를 사용하는 다양한 플랫폼의 디바이스일 수 있다.

또 다른 특성의 모바일단말기를 프록시서버의 통신연결 인터페이스에 연결하면, 프록시서버에서 이 추가된 모바일단말기의 운영체제에 접근하여 하드웨어 모델 및 운영체제의 정보를 기기정보로서 추출하거나, 시스템의 관리자로부터 상기와 같은 기기정보를 입력받을 수 있다.

3.2.3. 사용자 단말기

사용자단말기는 <Figure 2>와 같은 사용자 브라우저를 통해 웹데몬으로 구현된 클라이언트통신부에 접속하는데, 클라이언트통신부를 통해 사용자 단말기에 송신되는 웹페이지는 클라이언트통신부와 데이터를 송수신하는 웹서버통신부 및 프록시서버로 접속하여 모바일단말기의 시험을 진행하는 프록시서버통신부로 구성된다.



<Figure 2> Screen example of user browser

3.3 시스템 구성 요소의 기능

3.3.1. 프록시서버의 기능

모바일단말기의 기기정보는 프록시서버의 메모리 또는 기타 저장장소에 기록되어 제어부에 의해 웹서버의 기기정보관리부로 푸시된다. 또한 상기 제어부는 상기 연결된 각 모바일단말기의 상태로서 <Table 1>과 같은 상태정보를 추출하여 기기정보와 함께 기기정보관리부로 푸시한다.

<Table 1> Status data of mobile device

| Status ID | Status Information | Description |
|-----------|--------------------|---|
| 1 | Free | Not using now (No connected user) |
| 2 | Device Error | Device connection error |
| 3 | NetError | Communication error with mobile devices |
| 4 | xx.xx.xx.xx | IP number of current user computers |

프록시서버의 제어부와 웹서버의 기기정보관리부의 통신에는 HTTP 프로토콜이 사용된다. 이들의 통신에 있어 통상의 TCP/IP를 이용한 세션을 연결하여 두고 활용한다면 통신안전성이 떨어질 수 있고, 이러한 처리 모듈의 개발비용이 들 수 있다. 본 표준은, 기기정보관리부를 보다 적은 개발비용으로 구현할 수 있고, 통신안전성을 극대화하며, 웹서버의 데이터베이스에 대한 직접적인 접근을 제한하기 위해서 HTTP 프로토콜에 의하여 데이터를 송수신하는 모듈로써 구현되거나, 바람직하게는 통상의 웹데몬 및 HTTP 요청신호를 처리하는 웹페이지만으로 간단하게 구현할 수 있다. 실제로 프록시서버에 device1, device2, device3의 세 모바일단말기가 연결되어 있는 경우에 이들 모바일단말기의 기기정보 및 상태정보를 PUT 메소드의 인자값으로 하여, 제어부가 기기정보관리부에 송신하는 HTTP 요청신호는 <Table 2>와 같이 구현될 수 있다.

<Table 2> Signal code to request status data

```

httpConnection =
Session.GetHttpConnection( "211.47.235.17" ,
8083);
httpFile =
httpConnect->OpenRequest( "PUT" ,
"/webTestForte/setDeviceStatus.jsp" );
httpFile->SendRequest(NULL,0,
"device1=111.11.11.11&device2=Free&device3=DeviceError" , 53);
    
```

위에서 보이는 바와 같이 기기정보관리부에 대한 접속은 공개되지 않는 것이 바람직하므로 상기 기기정보관리부의 접속포트번호는 널리 알려진 포트를 제외하고 "8083"과 같이 별도로 정의하여 운영하는 것이 바람직하다.

3.3.2. 웹서버의 기능

HTTP 요청신호를 수신한 기기정보관리부는 전달받은 인자로부터 각 모바일단말기의 상태정보를 추출하여 데이터베이스에 저장 또는 업데이트한다. 기기정보관리부는 저장 또는 업데이트의 결과로서 HTTP 응답신호를 프록시서버의 제어부에 송신하는데, 이 때, 제어부는 이러한 응답신호를 파싱하여 각 모바일단말기의 상태정보 업데이트 성공여부를 인식하고 로그로 기록한다.

보안성의 극대화를 위해 기기정보관리부는 HTTP 요청신호를 송신한 측의 발신자 아이피주소를 확인하고, 상기 발신자의 아이피주소가 기 등록된 프록시서버의 아이피주소인 경우에만 상기 푸시된 각 모바일단말기의 기기정보 및 기기의 상태정보를 업데이트하도록 구현하는 것이 바람직하다. 이와 같은 방법으로써 어플리케이션 시험에 되는 자원에 대한 정보가 웹서버를 통해 추상화되고, 기타 장치의 데이터베이스 접근이 제한되므로 보안성이 향상되고, 통신트래픽을 최소화할 수 있고, 또한 통신안전성을 극대화할 수 있게 된다.

데이터베이스에 저장된 각 모바일단말기의 기기정보 및 기기의 상태정보는 클라이언트통신부에 접속한 어플리케이션 시험 사용자단말기에 송신된다.

4. 원격 시험 절차의 표준 제안

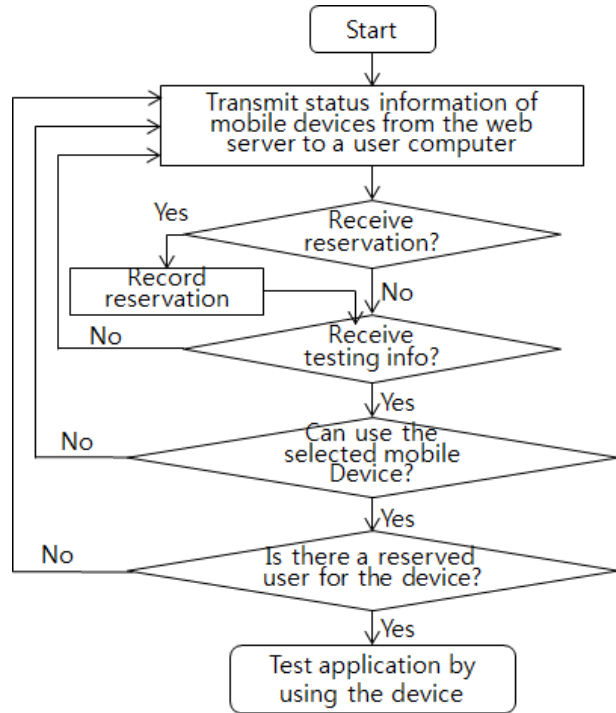
4.1 시험 예약 및 사용 허가

사용자단말기의 웹서버 통신부는 <Figure 3>에 나타낸 바와 같이 각 모바일단말기의 기기정보 및 기기의 상태정보를 수신하여 표시한다. 사용자는 표시된 각 모바일단말기의 기기정보 및 기기의 상태정보를 참조하여 어플리케이션 시험을 예약할 수 있다. 사용자로부터 선택된 모바일단말기의 식별번호 및 예약시간정보를 포함하는 예약요청이 사용자단말기의 웹서버통신부로부터 기기정보관리부로 송신되어 데이터베이스에 기록되는 방식으로 예약절차가 진행된다. 또 이외는 병렬적으로 사용자는 모바일단말기를 선택하여 사용을 요청할 수 있다.

제어부는 데이터베이스의 기기 상태정보를 참조하여 상기 사용요청된 모바일단말기가 현재 사용가능한 상태인지 판단하고, 예약정보를 참조하여 현재 다른 사용자에게 예약된 상태인지 판단하여 사용요청한 사용자 단말기의 웹서버 통신부에 사용허가를 내린다. 웹서버 통신부는 해당 모바일단말기의 기기정보로부터 단말기의 식별번호를 추출하여 프록시서버 통신부가 프록시 서버에 접속하고 상기 식별번호에 해당하는 모바일단말기에 대하여 시험을 진행하도록 한다. 이 때, 프록시 서버 통신부는 웹서버통신부를 통해 동시에 다수 개가 실행될 수 있으므로 사용자는 자신이 개발한 어플리케이션을 동시에 서로 다른 여러 플랫폼에서 실행하며 시험해볼 수 있다.

4.2 원격 시험의 진행

모바일단말기의 시험은 사용자브라우저에 포함된 프록시서버 통신부와 프록시 서버에 포함된 표시관리부, 설치관리부, 및 이벤트관리부가 서로 데이터를 송수신하는 방식으로 진행된다. 우선, 사용자는 프록시서버 통신부를 통해 자신이 개발한 어플리케이션의 소스파일 또는 오브젝트파일 등을 설치관리부로 송신하고, 설치관리부는 상기 수신한 소스파일 또는 오브젝트파일을 해당 모바일단말기에 전송하여 저장하는 방식으로 설치할 수도 있고, ADB(Android Debug Bridge)와 같이 해당 모바일단말기의 운영체제에서 제공하는 설치API(application programming interface)를 이용하여 설치할 수 있다.



<Figure 3> Flow chart of the procedure for testing reservation and permission

프록시서버의 표시관리부는 모바일단말기의 화면정보를 시험자의 프록시서버 통신부에 송신하고, 프록시 서버 통신부는 수신한 화면정보를 사용자단말기의 화면에 표시하여 마치 사용자가 해당 모바일단말기를 실제 보고 있는 것과 같은 환경에서 시험을 진행할 수 있도록 한다. 표시관리부는 해당 모바일단말기의 화면정보를 추출하기 위해 별도로 해당 단말기에 설치되고 실행되는 VNC (Virtual Network Computing) 모듈 또는 해당 모바일단말기의 운영체제에서 제공하는 API를 사용할 수 있다.

이벤트관리부는 프록시서버 통신부로부터 수신한 이벤트정보를 모바일단말기에 전달하여 시뮬레이션되도록 한다. 이벤트정보는 수치값 및 이벤트종류 정보를 포함하는데, 이벤트종류는 커서위치이벤트, 버튼입력이벤트, 모션센서의 센싱이벤트, 자이로스코프센서의 센싱이벤트, 위치정보수신이벤트 중에서 적어도 하나 이상을 포함한다. 예를 들어, 사용자가 사용자단말기의 입력장치를 통해 프록시서버 통신부에 표시된 모바일 단말기의 화면정보를 보면서 특정 위치에 커서를 위치한다면, 커서위치이벤트에 해당하는 이벤트종류 정보와 위치좌표를 이벤트정보로서 이벤트관리부에 송신하고, 이벤트관리부는 해당 모바일단말기의 운영체제에서 제공하는 API를 통해 모바일단말기의 커서를 해당 위치에 위치하도록 제어할 수 있다. 같은 방법으로 모션센

서의 센싱이벤트를 통해 해당 모바일단말기가 마치 움직이고 있는 것처럼 시뮬레이팅할 수 있고, 위치정보수신이벤트를 통해 해당 모바일단말기가 마치 이동하고 있는 것처럼 시뮬레이팅할 수도 있다.

시험 과정을 통해 발생한 모바일단말기의 시스템로그는 사용자가 디버깅에 활용할 수 있도록 기록되고, 사용자단말기에 송신될 수도 있다. 또한, 시험 과정에서 사용자가 해당 모바일단말기를 통해 유료의 부가서비스를 사용하는 경우가 발생할 수 있다. 예를 들어, 시험 도중 사용자가 전화통화를 하였거나, SMS를 이용하였거나, 결제서비스를 사용하였거나 하는 등의 부가서비스를 사용한 경우에 제어부는 이러한 부가서비스 사용내역을 기록하고, 기록된 사용내역에 따라 사용허가된 사용자단말기의 사용자에게 대하여 과금할 수 있다.

제어부는 시험이 완료되거나 프록시서버통신부와의 연결이 끊기고 나면, 해당 시험에 사용되었던 모바일단말기에서 설치된 파일을 삭제하고 환경설정을 초기화한다. 기기정보관리부는 예약정보를 참조하여 임의의 모바일단말기에 대하여 현재 시험 중인 사용자의 예약시간이 종료한 경우에는 프록시서버 통신부에 시험종료 신호를 송신하여 시험이 종료되게 한다.

5. 결론

본 연구에서는 모바일 어플리케이션의 개발자들이 다양한 디바이스를 구입하지 않고도 네트워크망을 통해 원격지에 있는 모바일 단말기들을 대상으로 어플리케이션을 시험해 볼 수 있는 시스템의 구성과 시험 프로세스에 대한 표준을 제시하였다.

본 서비스를 통해 개발자는 미보유 단말기를 확보할 부담이 해소되어 개발비를 절감할 수 있고, 다양한 모바일기기에 개발 어플리케이션을 적용할 수 있어서 판매수익을 증대시킬 수 있을 것이다. 또한 어플리케이션 출시 후 장애나 오류 발생 시 신속한 확인과 대응으로 고객 만족도를 향상시킬 수 있을 것이다.

모바일 어플리케이션 원격시험 서비스는 모바일 어플리케이션을 생산하는 산업 분야에 적용할 수 있다. 표준화된 시험 시스템 설계와 절차를 통하여 빠른 어플리케이션의 개발과 어플리케이션의 완성도를 높인다. 본 서비스가 표준화되면 어플리케이션이 가져야 하는 기능 및 품질특성들에 대해 개발자, 사용자 및 제3자가 각자의 목적에 맞게 활용할 수 있다. 개발자는 개발 시 사용자가 요구하는 기능 및 품질특성을 모바일 어플리케이션이 충족하는지를 점검할 수 있고, 사용자는 검수

시 요구되는 기능 및 품질특성을 충족하는지를 파악할 수 있으며, 제3자 공인기관은 인증기관의 역할을 수행할 때 본 표준을 활용할 수 있다.

6. References

- [1] STA Testing Consulting Technical Report, "The Technology Status of Automatic Testing for Mobile Applications," 2014. 10.
- [2] HP Business Reports, "Mobile Testing," <http://hp.com/go/mobiletesting>
- [3] Cohn, M., "Succeeding With Agile: Software Development Using Scrum," Addison-Wesley Professional, 2009. 11.
- [4] Onycom Website, <http://www.onycom.com>
- [5] Amalfitano, D, Fasolino, A.R., Tramontana, P., and Robbins, B., "Testing Android Mobile Applications: Challenges, Strategies, and Approaches," *Advances in Computers*. Vol.89, 2013: 1-52.
- [6] Gao, J, Bai, X.Y., Tsai, W.T., and Uehara, T., "Mobile Application Testing: A Tutorial," *COMPUTER*, Vol.47, No.2, 2014: 46-54.

저자 소개

한 영 근



현 명지대학교 산업경영공학과
교수

주소 : 경기도 용인시 처인구 명지로 116 명지대학교
공과대학 산업경영공학과

이 현 노



현 어니컴(주) 수석연구원

주소 : 경기도 용인시 기흥구 영덕동 1029

전 준 호



현 어니컴(주) 상무이사

주소 : 경기도 용인시 기흥구 영덕동 1029