

디지털제품 실행환경에서 소프트웨어 테스트 자동화 적용 기술

목 차

1. 서 론
2. 디지털제품 내 소프트웨어 테스트 현황
3. 제품 테스트 자동화 도구 개발 및 적용 사례
4. 테스트 자동화 발전 방향
5. 결 론

김용희 · 김진옥
(삼성SDS · 대구한의대학교)

1. 서 론

정보통신기에서 소프트웨어는 핵심적인 역할을 수행하며 중요성을 차지하고 있다. 그러나 국내 IT산업 환경에서 필요한 인적자원과 기술은 그에 미치지 못하고 있다. SoC(System On Chip) 및 핵심 시스템 소프트웨어의 해외기업에 대한 의존성은 갈수록 심화되고 있으며 시장 요구와 사용자 기호에 맞추기 위해 빠른 제품 개발에 대한 압력도 가속화되고 있다.

전통적인 업무용 소프트웨어와 임베디드 소프트웨어 개발 및 테스트 과정에서 <표 1>과 같은 상대적 차이점과 공통점이 발견된다. 표에 따르면 임베디드 소프트웨어 개발에도 기존의 업무용 소프트웨어와 비슷한 어려움이 상존함을 알 수 있다. 또한 하드웨어와 통합과정에서 추가되는 많은 문제들에 의해 전반적인 환경과 개발 및 테스트 영역에서의 개선이 필요하며, 이를 통해 품질 완성도와 개발 생산성 향상이 시급한 상황이다. 그러나 소프트웨어의 비 가시성, 재활용의 어려움과 인적 요소라는 자체적인 한계로 인해 여전히 기대할 만한 성과를 거두지 못하고 있

<표 1> 업무용 소프트웨어와 임베디드 소프트웨어에서의 개발과 품질 이슈

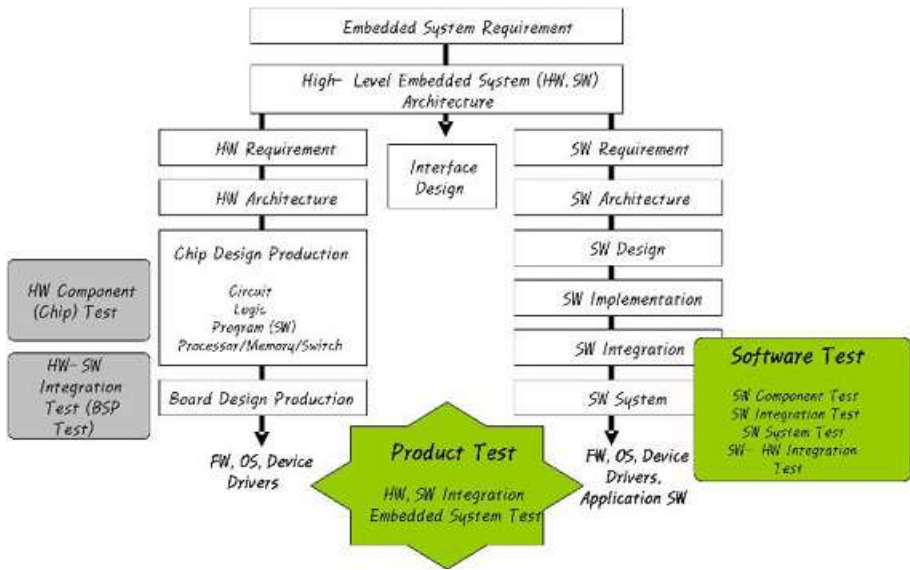
구분	업무용 소프트웨어	임베디드 소프트웨어	
차이점	개발 lifecycle	RFP/RFI 존재	명확하지 않은 요구사항
		소프트웨어가 주도	하드웨어 주도의 개발, 핵심칩 의존
		정해진 납기와 개발일정	빠른 출시요구에 따른 촉박한 납기
	품질보증	개발방법론 정착	양산(mass production)
		품질업무 중요성 인식	경영 및 개발리더의 인식 미흡
		관련 연구/적용경험 보유	하드웨어와 통합시험
유지보수	정해진 사용자	불특정 다수 사용자	
	유지보수 또는 SLA 계약	기능과 품질의 브랜드 영향도 높음	
공통점	개발 lifecycle	설계/개발전문인력 및 핵심기술 부족	
	품질보증	테스트관련 인력/기술 부족	
	유지보수	소프트웨어 재사용성 미흡	

는 실정이다.

본고는 일반 소비자용 전자 제품(디지털 TV, 카메라, 휴대전화 등)에서의 소프트웨어 테스트 자동화 사례 제시를 통해 그 동안 소프트웨어 측면에서 관련 업계에서 겪어왔던 한계를 극복하는 계기를 제공하고자 한다.

2. 디지털제품 내 소프트웨어 테스트 현황

소프트웨어가 사용목적에 맞게 제대로 만들어졌는지를 판단하는 품질관리(Quality Manage-



(그림 1) 디지털제품 개발 및 테스트 절차

ment)는 크게 결함 예방 활동을 의미하는 소프트웨어 품질보증(Quality Assurance)과 결함을 발견하는 활동인 품질통제(Quality Control)로 이루어진다. 테스트(시험)은 품질통제 활동의 하나로서 결함을 발견하고 평가하는 활동 또는 예상결과를 만족하는지 검증하거나 차이를 식별하는 작업[1]이며 결함을 없애는 과정인 디버깅과는 차이가 있다. 또한 테스트는 하나 또는 여러 개의 테스트케이스 집합을 의미하기도 하는데 이때 테스트케이스는 테스트 기준과 방법을 문서화한 것으로서 입력조건, 테스트절차, 테스트 결과를 기술해놓은 문서이다[2].

임베디드 소프트웨어 개발 및 제품단계 품질 관리와 테스트 성과를 높이려는 많은 노력이 진행 중이지만 제품 및 프로세스관점에서 소프트웨어와 테스트 수준을 측정하기 위한 ISO 9000-3, SPICE, CMMI, TMM/TPI 등의 성숙도 모델이 실제 디지털 제품 개발과 정합되지 않는 문제가 존재한다. 또한 전통적인 소프트웨어 개발 라이프사이클 모델[3]이 임베디드 시스템에서는 적용이 어려운데 그 원인은 여러 가지가 있

겠지만 (그림 1)에서 보듯 하드웨어와의 통합에도 상당부분 기인한다. 다양한 디지털제품 개발 환경에서 정도의 차이는 있겠으나 소프트웨어 품질과 테스트에서의 공통적인 현황과 이에 따른 시사점은 다음과 같다.

2.1 임베디드 소프트웨어 개발 현황

디지털제품의 특성 상 빠른 시장 출시가 필요하므로 소프트웨어 또한 어느 정도의 잠재 불량을 포함한다. 때에 따라서는 발생한 결함 수정 비용을 제품 판매에 따른 이익으로 상쇄하기도 한다. 이는 하드웨어와 달리 모듈 표준화가 어려운 것이 원인이기도 하다.

또한 핵심칩(chip), 디바이스드라이버, 미들웨어와 응용프로그램 등 다양한 대상에 대해 일관된 개발표준이나 소프트웨어 개발 순환 주기(Software Development Life Cycle)를 적용하기가 어렵다. 전통적인 Waterfall 및 Iterative 방식 위주에서 최근 Agile 방식을 적용하는 등 개발방법을 다양화하기 위한 노력이 진행되고 있으나 요구분석과 설계 단계부터도 여전히 짧은 개발

기간과 전문인력 부족 등의 어려움을 겪고 있다.

그 외에 소프트웨어의 서비스화 추세에 따라 기존 PC 및 Web 기반 응용 프로그램과의 호환성과 통합관점에서 완성도가 매우 필요하며 이로 인한 소프트웨어 품질보증의 중요성은 더욱 증가하고 있다.

2.2 소프트웨어 품질 보증과 테스트 현황

임베디드 시스템의 경우, 테스트를 위한 V 모델을 적용할 때 통합 및 시스템 테스트 단계가 가장 중요하며 이를 극복하기 위해 다중 V 모델 및 테스트 ROI 등이 고려되고 있다.

소프트웨어 품질 보증과 테스트 기반 기술의 경우, 하드웨어와 사용자 인터페이스 관점 위주로 발전해온 국내 환경상 신뢰성, 사용성, 호환성 분야에서 하드웨어 품질보증과 전통적인 업무용 소프트웨어와 ISO 9126 등의 품질 특성[4] 검증 방법이 부분적으로 도입되어 왔다.

개발 단계 code 검증, 통합 및 제품단계 품질 보증 전반에 걸친 테스트 방법론, 도구 및 기법, 성과 측정을 위해 아직 체계적인 대안이 미흡한 측면이 있다. 테스트 방법론 측면에서는, 테스트의 목적 자체가 대상에서 결함을 찾는 것이지만 완벽한 테스트를 하거나 무결점을 증명하는 과정이 아닌, 주어진 자원(시간, 인력, 비용 등) 하에서 위험성이 높은 결함을 발견할 확률을 극대화하는 것이라는 Risk기반 테스트방법론[5]이나 TDD(Test Driven Development) 등이 주목을 받고 있다. 테스트 도구 및 기법 측면에서는 경험/구조/명세를 기반으로 한 테스트, Sanity/Smoke 테스트, 정적(Static)/동적(Dynamic) 테스트, 화이트박스/블랙박스 등이 테스트 자동화와 더불어 적용 중에 있다.

인력 양성 측면에서는 수 년 전부터 소프트웨어 품질 보증 및 테스트 전문가를 양성하기 위해 기업, 관련기관 등의 노력이 진행되고 있으나 여전히 단순 제품테스트 위주에 머물고 있으며 프

로세스 및 기법, 기술에 대한 이론과 경험을 갖춘 인력이 부족하다.

그 외 테스트업무 측면에서 아웃소싱이 도입되고 있으나 비용 및 효과 측면에서 여전히 성장에 더딘 추세이다.

2.3 테스트 자동화 현황

테스트 자동화는 넓은 의미에서는 테스트 관리, 테스트 실행, 테스트 결과 확인과 같은 테스트 활동을 수행하거나 이를 지원하기 위해 소프트웨어 도구를 사용하는 것이며, 좁은 의미로 테스트 케이스 별로 테스트 실행 자체를 소프트웨어로써 자동화하는 것을 의미한다.

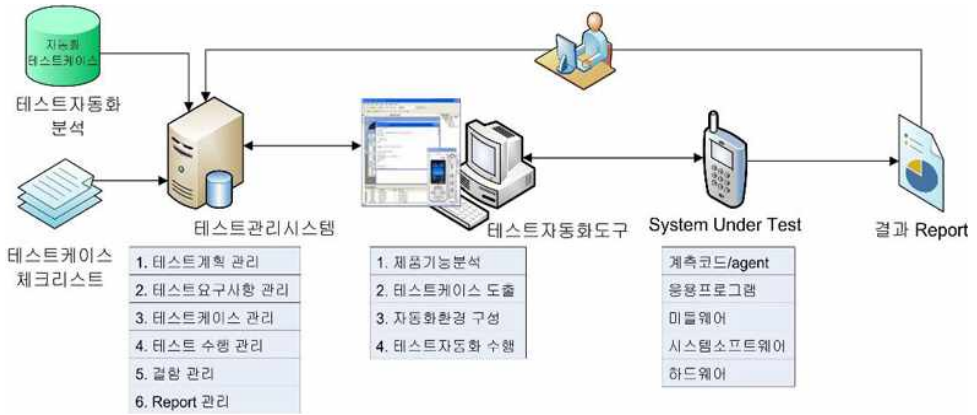
개발단계의 경우, Source Code의 결함을 줄이기 위한 Code Inspection, 정적/화이트박스 테스트영역에서 Open Source와 다양한 상용도구들이 적용되고 있으나 제품 실행 측면의 경우, 전문 인력 부족과 빠른 납기 등의 사유로 인해 활성화가 어렵다. 제품 테스트의 경우 주로 Record-Playback방식과 사용자 메뉴 또는 State Transition Diagram 기반으로 자동화가 수행된다.

개발 및 제품단계 공통적으로 테스트 플랫폼 또는 프레임워크와 기업 테스트 관리 프로세스 관점에서의 접근이 필요하다.

3. 제품 테스트 자동화 도구 개발 및 적용 사례

본 장에서는 디지털 제품의 실행 환경(Dynamic) 측면에서 블랙박스 테스트기법이 적용된 사례를 분석해본다.

기업별로 다양한 제품을 생산하거나 특정 제품에 역량을 집중하는 경우가 있는데 그 효과 측면에서 먼저 테스트를 하기 위한 기술적/업무적 기반구조 또는 프레임워크[6]를 구축하는 것이 가장 중요하다. 이외에 소프트웨어 품질보증의 중요성 공감과 명확한 목표, 테스트 환경에 대한 물리적/논리적 인터페이스 정의, 개발과 테스트



(그림 2) 테스트 자동화 업무 흐름

조직간 협업 문화, 필요 자원 확보 등이 핵심 요소이다. (그림 2)는 테스트 자동화 관련 업무흐름을 제시하고 있다.

최근 들어 전사적 소프트웨어 엔지니어링 업무의 중요성이 증가함에 따라, 점차적으로 테스트 업무와 요구 공학, 결함 관리 간의 효과적인 연계라는 발전을 예측할 수 있다.

3.1 테스트 자동화 도구 개발 및 적용 사례

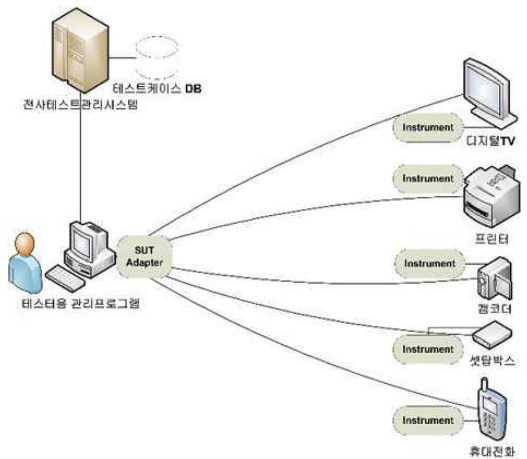
3.1.1 테스트 자동화 도구 개발

자동화 도구의 도입 또는 개발을 위해서는 테스트 자동화 프레임워크(플랫폼)가 명확히 정의되고 합리적인 목표가 수립되어야 한다. 다양한 디지털제품 테스트를 위한 범용적인 자동화 도구는 찾기 어려우므로 실제 제품의 동작에 따른 구조 수정이 필요하다.

또한 제품에 대한 테스트케이스 분석을 거쳐 적용 범위를 선정해야 한다. 그리고 테스트 자동화 도구 자체의 안정적인 소프트웨어의 품질이 전제되어야 하므로 code inspection과 충분한 테스트를 거쳐야 한다.

테스트 자동화 도구는 크게 테스터용 관리 프로그램(host), 테스트 대상(System Under Test 또는 target)내 응용프로그램(instrument 또는

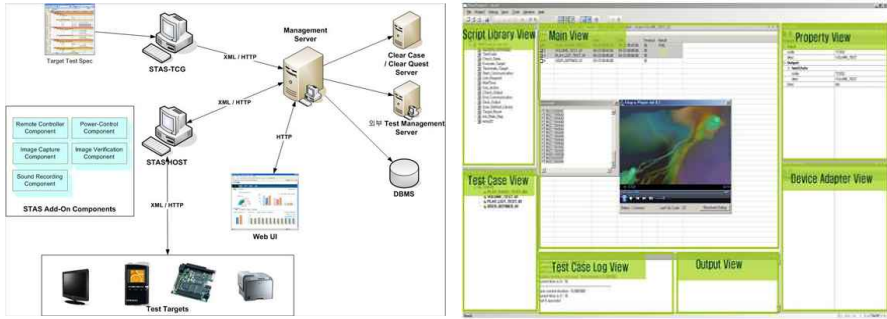
agent), 그리고 관련 인터페이스(host-target, host-기타 응용프로그램 등)로서 정의할 수 있다. (그림 3)에서 이와 같은 구조를 볼 수 있다.



(그림 3) 제품 테스트 자동화 환경

(그림 4)에서는 테스트용 관리프로그램이 테스트 과제 정의, 테스트 케이스 관리, 스크립트 언어 엔진, 디지털제품 인터페이스, 테스트 결과 보고로서 구성되어 있음을 알 수 있다.

이처럼 테스트 자동화 도구는 프레임워크 관점에서 호스트, 인터페이스, 에이전트의 세 가지로 구성할 수 있으며 소프트웨어 표준 환경을 정의한 후, 실제 테스트 환경에 따라 수정되어야



(그림 4) 테스트 자동화 도구 구현 사례

하며 유지보수와 이식성 등 소프트웨어 품질 특성[4]을 가져야 한다.

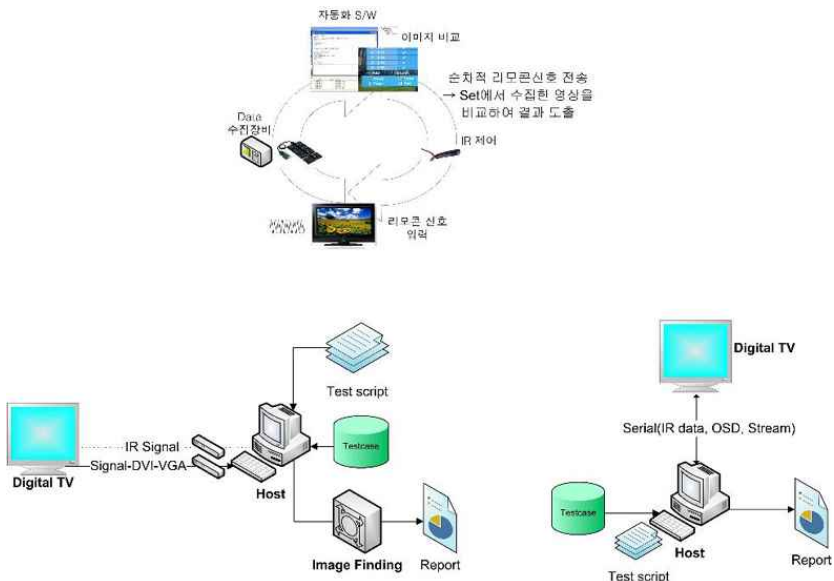
<표 2>는 테스트 자동화를 위해 사전에 고려할 사항이다.

3.1.2 적용 사례

디지털제품별로 차이는 있으나 제품 실행 시에 사용자 환경 및 기능 측면에서 공통적인 테스트 대상이 존재한다. 즉 메뉴 등 사용자 인터페이스, 이미지, 소리, 동영상 등 멀티미디어의 정상적인 동작 여부가 테스트 자동화 업무의 우선적인 대상이 된다. 그 외 스트레스, 부하테스트 등의 성능과 정밀 측정에 자동화를 적용할 수 있으며 장시간의 단순 반복 테스트에 대해서는 특히 시간절감 효과가 높다.

<표 2> 제품별 사전 고려사항

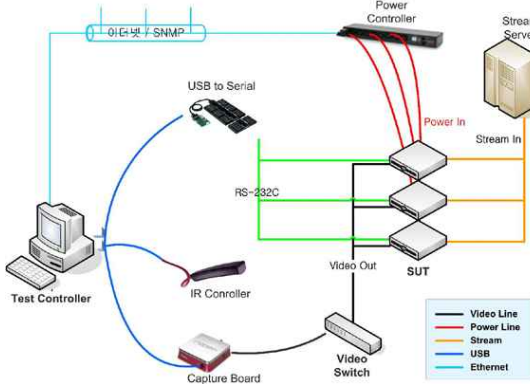
구분	고려사항	제품 종류
설치형(home) 제품	- 시스템자원이 충분 - 긴 개발기간 - 개발표준 또는 개발플랫폼과 연결	디지털 TV 디지털 셋탑박스 프린터 복합기(MFP)
이동형(mobile) 제품	- 시스템자원이 부족하고 개발기간이 짧음 - 다양한 개발환경 - 제품 순환주기가 짧음	휴대전화 MP3 플레이어 디지털 카메라 디지털 캠코더



(그림 5) 디지털TV 제품 테스트자동화 사례

제품 별로 구체적인 사례를 살펴보면, 디지털 TV의 테스트 대상은

- 제품 사용지역에 따른 디지털 TV규격 표준
- 메뉴 및 OSD(On Screen Display) 언어와 메시지의 정확성
- 디지털영상의 정확성 등이며 다음과 같이 테



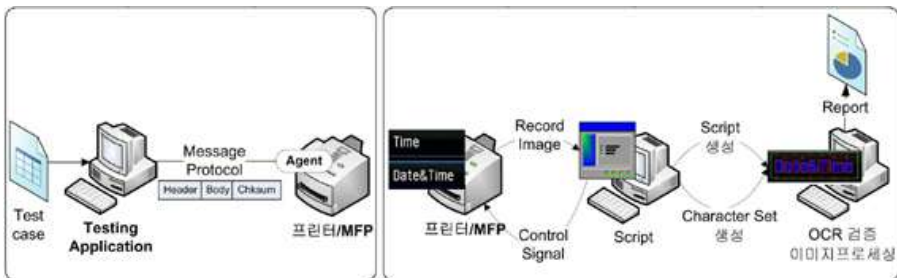
(그림 6) 셋탑박스 테스트 자동화 시스템 구조

스트 자동화 환경을 구성할 수 있다.

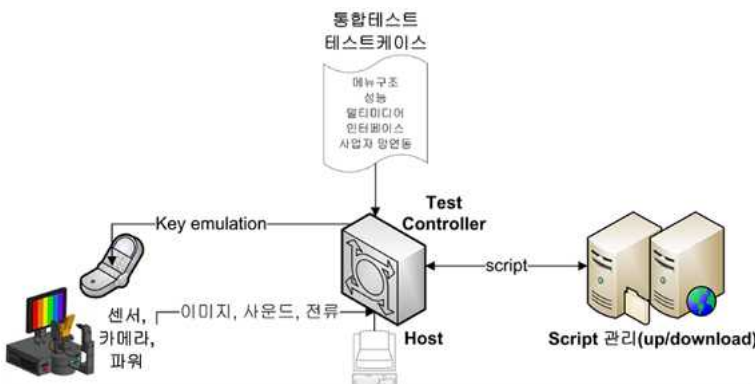
디지털 셋탑박스는 디지털 TV와 유사한 기술 구조 및 테스트대상을 갖는다.

프린터와 인쇄, 스캔, 복사, FAX 기능을 가진 복합기(Multi Function Printer)는

- 인쇄 품질
 - 사용자 환경에 설치되는 프린터 드라이버/지원 소프트웨어
 - LCD 등 사용자 인터페이스가 테스트 대상이다.
- 이동형 디지털제품의 대표적인 휴대전화는 기존 통화기능에 MP3 재생, 다국어 지원, 일정 관리, 멀티미디어 관리 등 많은 사용자 기능이 추가됨에 따라 소프트웨어 품질의 중요성이 심하게 대두되고 있다.
- 메뉴, 다국어 등 사용자 인터페이스
 - 멀티미디어 인터페이스 및 기타 응용프로그램 기능 동작
 - 전류 및 전압



(그림 7) 프린터 테스트 자동화 시스템 구조

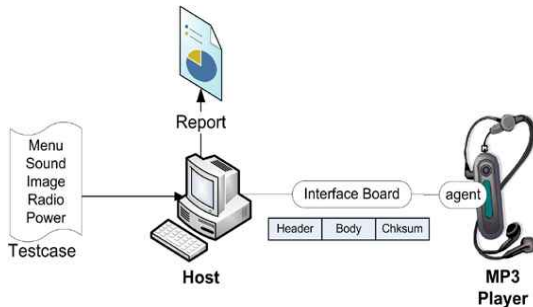


(그림 8) 휴대전화 테스트 자동화 시스템 구조

- 사업자 망연동 테스트 케이스에 대한 테스트 자동화가 필요하다.

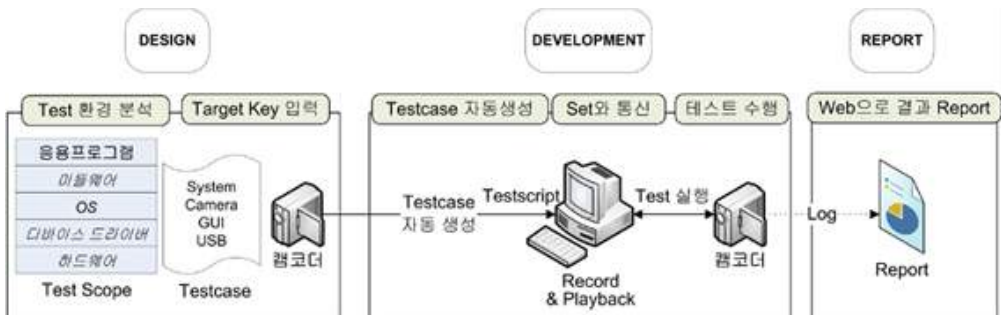
MP3 플레이어의 경우, CPU, 메모리, 통신 인터페이스의 자원이 제한되어 있으므로 테스트 환경 구성이 쉽지가 않다. 따라서 테스트를 위한 별도의 물리적/논리적인 인터페이스가 준비되어야 하고 음악 재생 및 외부와 통신을 위한 task 동작에 있어 고유의 성능에 영향을 일으키지 않도록 주의를 기울여야 한다. 또한 제품 성격상 개발과 시장 출시 기간이 촉박함을 고려해야 한다.

- 메뉴 동작 등 사용자 인터페이스
- 메모리, 사운드, 라디오 기능
- PC와의 인터페이스 등이 테스트 대상이며 테스트 자동화 환경 구성은 다음과 같다.



(그림 9) MP3 플레이어 테스트 자동화 시스템 구조

디지털 카메라와 캠코더에 대해 아래처럼 Record-Playback 방식으로 사용자 Key입력에 따른 시스템 동작 테스트를 자동화할 수 있다.



(그림 10) 디지털 캠코더 테스트 자동화 시스템 구조

3.2 테스트 자동화 프로세스

제품 테스트 자동화 절차는 개발 및 품질 보증 업무와 조화를 이루어야 하며 특히 개발자와 협업이 중요하다. 시장 요구 사항이나 멀티미디어 칩 등의 환경이 수시로 바뀔에 따라 유기적인 대응이 필요하다. 이를 위한 테스트 자동화 업무 프로세스는 다음과 같이 구성될 수 있다.



(그림 11) 테스트 자동화 프로세스

테스트 프로세스(공정) 관리 측면에서의 시스템화는 반드시 필요하다. 장기적인 소프트웨어 품질 보증 효과를 위해서는 개발 프로세스와 유기적인 연계, 테스트 업무의 시스템화라는 기반 구조를 확보해야 한다. 최근 전사적인 테스트 프로세스 관리는 디지털 제품 개발뿐만 아니라 대부분의 소프트웨어 개발에서 필요성이 커지고 있다. 테스트 프로세스 관리 시스템이 가져야 할 기능은 (그림 12)와 같다.



(그림 12) 테스트 프로세스 관리시스템의 기능

테스트 프로세스 관리시스템은 요구사항관리, 결함관리, 형상관리, 지표관리 등 소프트웨어 엔지니어링 관련 시스템들과 효과적으로 결합되어 제품 소프트웨어 품질 향상과 개발 생산성 향상이라는 높은 시너지 효과를 가져올 수 있다.

4. 테스트 자동화 발전 방향

4.1 테스트 자동화 적용 효과 및 시사점

디지털제품에 대한 테스트 자동화를 통해 개발 또는 테스트 납기를 단축하여 제품 개발 생산성을 높이는 성과는 정량화하기가 쉽지 않다. 이유는 제품의 시장 순환 주기가 길거나 디지털제품의 특성 상 명확한 불량원인 파악이 어렵기 때문이다. 그러나 최근 1~2년의 테스트 자동화 적용 사례를 분석해 봤을 때 디지털 제품 별로 차이는 있지만 전체 테스트케이스 중에서 가장 기본적인 제품 기능 검증에 대해 30% 이상을 자동화시켜 해결할 수 있다. 또한 일부 테스트 케이스는 자동화 기술을 도입함에 따라 10% 이하로 시간이 절감되는 경우가 있었으며 단순 반복되는 테스트이거나 화질, 음질, 문자 등 정밀 측정

과 비교가 필요한 경우 테스트 자동화 효과가 매우 높았다. 그 외 개발자나 테스터의 업무를 좀 더 효율적으로 변화시킬 수 있었다.

다만 환경 구성에 시간이 소요되는 테스트 자동화보다 숙련된 수작업 테스트가 더욱 효과적인 경우가 존재하였다. 따라서 디자인-개발-테스트 조직 간의 업무 프로세스 연결과 각 산출물 관리를 시스템화 시키는 것이 테스트 자동화 효과를 높이기 위한 선결요건이라고 판단된다.

테스트 자동화 도구를 도입할 때 비용과 성과 측면의 분석이 전제되어야 하며 꾸준한 지표 관리를 통한 성과 확보는 바람직하지만, 수작업이 필요한 테스트 케이스나 자동화 기술 및 인력 수준을 고려하지 않은 무리한 추진은 결코 효과가 높지 않다는 것을 유의해야 한다.

그 외 시사점을 정리해보면,

- 테스트 업무는 비용으로 고려됨. 따라서 품질-납기-수익 관점에서 조직 내 의사 결정자의 균형적인 시각이 필요함
- 테스트 프레임워크 또는 플랫폼 준비가 선결되어야 개발 플랫폼과 연결을 통한 적용 기간 단축 등 시너지 효과가 크며 테스트 자동화용

Code 추가가 용이함

- 자동화 기술 도입/개발 및 인력 확보/교육에 따른 비용을 고려해야 함[7]. 소프트웨어와 하드웨어 개발 경험과 공학이론, 제품 도메인에 대한 지식 확보가 필요함
- 테스트 자동화 도구 자체의 무결함이 선결되어야 Verification/Validation 타당성 확보[8]
- 디자인, 개발, 테스트 조직 간의 업무프로세스가 정합되는 경우, 테스트 자동화 효과가 높음
- 전사적 테스트 프로세스 관리 시스템 구축

4.2 발전 방향 및 고려 사항

인력측면에서 테스트 분야에 대한 적절한 성장 경로가 제시되어야 우수한 인력을 확보할 수 있다. 소프트웨어 및 하드웨어 개발 경험, 품질 관련 지식, 디지털 제품에 대한 깊이 있는 식견을 가진 전문가란 개발 및 품질 어디에서도 필요하나 그 공급은 많지 않으므로 교육/훈련을 통한 역량 확보가 수반되어야 한다.

기술측면에서 일단 테스트 프레임워크가 존재하고 경험이 축적되면 각 테스트 케이스 별로 추가적인 자동화 여부 분석과 적용이 가능할 것이다. 여전히 디지털 제품에 있어 다양한 인터페이스, 동영상 등 멀티미디어에 대한 품질 확보는 쉽지가 않은 분야이다. 또한 하드웨어와 소프트웨어를 통합적으로 테스트하기 위한 기술 요구가 증가할 것이다.

프로세스 측면에서는 개발부터 테스트 공정 관리 및 테스트 자동화까지 표준 업무 구조에 기반한 시스템화가 핵심적인 사항이다. 제품에 따라 다양한 요구 사양, 기술적 구성 요소와 소프트웨어 품질 요소가 존재하므로 이를 표준화 할 수 있는 업무 프로세스와 기반 시스템이 반드시 필요하다. 특히 시장 순환 주기가 짧은 제품의 경우, 개발 및 사용자 인터페이스에 대한 디자인 프로세스와 테스트 자동화 업무를 서로 조율해야 할 필요가 있다. 개발 방법론에 테스트 자동

화 프로세스와 테스트 공정 관리 프로세스를 반영하여 이를 시스템화한다면 전사적인 소프트웨어 엔지니어링 측면에서의 관리가 가능할 것이다.

그 외 임베디드 소프트웨어 개발에 있어 분석과 설계분야 또한 함께 성숙해야 하며 합리적인 지표 관리를 통한 성과와 시장 품질 비용 측정, 경영자와 개발자들의 품질에 대한 적극적인 인식 전환 또한 테스트 자동화와 품질 확보를 위해 필수 불가결한 요소이다.

5. 결론

일반 사용자용 디지털제품과 각종 정보통신기기에서 소프트웨어의 증가하는 복잡성을 고려할 때, 품질 수준 또한 적절히 확보되어야 한다. 유사한 기능, 디자인, 브랜드 요소를 갖춘 제품들의 경우, 결국 품질수준이 시장에서 중장기적인 결정요소가 될 것이다. 또한 제품 이미지에 대한 품질 결함 및 시장 불량 과급효과 영향력은 더욱 커질 것이다.

본고에서는 디지털제품들에 대한 소프트웨어 테스트 자동화 기술 사례 분석을 통해 임베디드 시스템 개발 특성에 맞춘 소프트웨어 품질 확보 방법을 살펴보았다. 소프트웨어, 하드웨어, 시스템 등 여러 요소간의 통합적인 품질 관점 확보, 테스트 기술 연구와 개발, 전문기업에 의한 테스트 아웃소싱 등 소프트웨어 품질 측면의 지속적인 노력을 통해 임베디드 소프트웨어 분야에서도 다양한 테스트 자동화 기술이 개발되고 더욱 낮은 비용으로 활용이 가능할 것이다.

그 동안 하드웨어에 기반한 대량 생산과 디자인 요소를 통해 국내 디지털 제품은 세계적인 경쟁력을 확보할 수 있었다. 소프트웨어 테스트 자동화 등 디지털 산업분야에서 상대적으로 낙후한 분야까지 국내기술이 진출할 수 있는 계기가 되어 향후 큰 과급효과를 기대한다.

참고문헌

- [1] IEEE Std 829 1998 : Standard for Software Test Documentation
- [2] IEEE Std 610 1990 : Standard Glossary of Software Engineering Terminology
- [3] Shari Lawrence Pfleeger, Software Engineering Theory and Practice, Prentice Hall, 2001.
- [4] ISO/IEC 9126, Information Technology Software Quality Characteristics and Metrics
- [5] Rex Black, Pragmatic Software Testing, Wiley, 2007.
- [6] Bart Broekman and Edwin Notenboom, Testing Embedded Software, Addison Wesley, 2003.
- [7] Cem Kaner, Testing Computer Software(2nd Ed.), International Thomson Computer Press, 1993.
- [8] IEEE Std 1020 1998 : Standard for Software Verification and Validation

저자약력



김 용 익

1992년 연세대학교 응용통계학과(학사)
 1991년 2006 KT 삼성전자/SDS VOD 및 디지털TV
 시스템 개발, 국가 정보화지원사업 PM 및 컨설팅
 2007년~현재 삼성SDS EO사업부 SQA지원파트장(수석보),
 정보관리기술사
 관심분야 : 임베디드 소프트웨어 품질보증, 테스트 자동화,
 IT 컨설팅
 이 메 일 : yongheekim@samsung.com



김 진 옥

1989년 성균관대학교 졸업(학사)
 1998년 성균관대학교 정보통신공학과 졸업(석사)
 2002년 성균관대학교 대학원 전기전자 및 컴퓨터공학과
 졸업(박사)
 2004년~현재 대구한의대학교 정보경영대학
 모바일콘텐츠학부 조교수
 관심분야 : 멀티미디어공학, 패턴인식, 영상처리,
 유비쿼터스 컴퓨팅 등
 이 메 일 : bit@dhu.ac.kr