

“ 자동화 기술을 활용한 임베디드 시스템 소프트웨어의 테스트 사례 연구 ”

한상섭, 양인승, 김은주
삼성전자, CS 경영센터, 전문기술 그룹
{ rams.han, inseung.yang, Drez.kim } @samsung.com

Abstract

디지털 제품의 기능 다양화 및 복잡화에 따라 제품 개발 주기 내 소프트웨어 개발 및 테스트 조직의 업무 중 30~40% 이상이 소프트웨어 테스트 업무에 집중하고 있다. 급변하는 소비자 요구사항에 대응하기 위해서는 신제품 개발 주기가 짧아지면서 개발 모델 수와 테스트 항목이 지속적으로 증가하고 있기 때문에 소프트웨어 기본 기능 및 UI (User Interface) 검증에 대한 자동화의 필요성이 대두 되었다. 본 논문에서는 휴대폰, DVDP, PDA에 대한 System Integration 작업을 통해 입출력을 정의하고 Black-Box testing을 구현하여 평균 64%이상의 자동화 율과 기본 Test Spec. 대비 80% 이상의 Test coverage를 확보했으며 많은 장단점을 도출할 수 있었다.

0. Introduction

오늘날 Embedded System의 품질은 소프트웨어 품질로 대변될 정도로 소프트웨어 품질에 대한 중요성이 대두되고 있다. 실제로 개발 단계의 많은 시간을 소프트웨어 테스트에 투자하고 있다. 디지털 제품의 소프트웨어는 제품의 모든 기능을 제어하는 중요한 역할을 담당하기 때문에 소프트웨어의 결함은 곧 시스템 결함으로 소비자에게 많은 불편함을 주고 회사 브랜드에 많은 손상을 입힐 수 있게 된다. 실례로, 잠재적인 소프트웨어 결함을 갖고 시장에 출시된 제품이 많은 손실을 입힌 경우는 산업계에서 어렵지 않게 찾아 볼 수 있다.

하드웨어와 다른 점은 하드웨어의 경우, 많은 결함이 개발 단계에서 테스트를 통해 결함을 재현하거나 하드웨어 자체에 테스트 벡터를 내장하여 자체 진단하는 기능까지 구현되어 있다. 하지만, Embedded System Software는 한정된 메모리 크기에 최적화된 코드로 그 기능을 구현해야 하기 때문에 하드웨어와 같은 테스트는 자체적으로 해결하기는 어렵다. 또한, 소프트웨어의 비가시적인 특성 때문에 잠재적인 결함을 발견하기 어려우며 시장에서 발견된 결함을 재현하기 또한 쉽지는 않다. 이러한 디지털 제품의 소프트웨어 품질을 어떻게 하면 향상시킬 수 있을까? 본 논문에서는 소프트웨어 자동화 테스트 기법을 디지털 제품에 적용하기 위해 다음과 같은 몇 가지 관점에서 접근했다.

- 1) 개발기간 내의 소프트웨어 시험에 드는 노력 감소
- 2) 소프트웨어 인력에 의한 단순 반복 시험 자동화
- 3) 납기 전 소프트웨어의 잦은 사양 변경에 따른 결함 과급 억제
- 4) 소프트웨어 테스트 자동화 율 향상, 테스트 커버리지 확보

단순히 테스트를 반복하거나 많은 시간과 인력을 투자한다고 해서 디지털 제품의 소프트웨어 품질 향상이 완성되는 것은 아니다. 본 논문에서는 소프트웨어 자동화 테스트 기법 적용을 위해 수행되었던 여러 가지 개선 방법들과 함께 설명하고자 한다.

1. 소프트웨어 자동화 테스트 프로세스

Embedded System Software의 소프트웨어 자동화 테스트 방법은 해당 디지털 제품의 입출력 정의에 따라 Black-Box test로써 구현될 수 있으며 보다 높은 자동화율 및 Test coverage를 구현하기 위해서는 Intrusive System Integration 작업이 필요함을 알 수 있었다. 실제로 휴대폰의 경우 제품의 입출력 부분인 LCD Driver와 Key Pad Driver에 대한 연구 후 그림 2.와 같은 시험환경을 구성하여 초기 시험 때 보다 30% 이상의 자동화율을 높일 수 있었다. 그러나, 시험 환경 구성만으로 소프트웨어 자동화 테스트의 효율을 높일 수는 없다. 표 1.은 디지털 제품의 소프트웨어 자동화 테스트 작업 절차를 나타낸 것이다. 소프트웨어 자동화 테스트를 위해 해당 제품의 입출력 관계 및 인터페이스 관계를 분석하고 개발 제품 및 모델의 특성을 파악하여 테스트 중점 사항을 도출하며 시스템의 제약 조건 및 소프트웨어의 기능 요구사항 그리고 구조를 분석해야 한다. 소프트웨어 자동화 테스트에서 가장 중요한 소프트웨어 테스트 Spec.은 각 제품 및 모델별로 테스트 의뢰자가 소프트웨어 요구사항에 맞게 제출해야 하며 이 Test Spec.은 시험 Pass/fail의 기준이 된다.

이러한 시험 사전 작업을 거친 후 해당 제품의 입출력을 정의하여 시험 환경을 구성한다. 이때 정의된 입출력 관계가 시험 환경에 적합 하지 못할 경우 tailoring 작업이 필요하게 된다.

자동화 스크립트 작성은 항상 처음과 끝이 동일해야 하며 해당 소프트웨어 구조를 완벽히 분석하여 모든 기능을 활성화 시키고 검증해야 한다. 이때 소프트웨어 전원 On/Off 와 같은 반복적인 작업은 소프트웨어 신뢰성 시험 항목으로 구성하여 소프트웨어 뿐 만 아니라 시스템 시험으로 발전시킬 수 있다. 소프트웨어 자동화 시험을 위한 모든 사전 작업을 완료한 후 테스트 스크립트를 실행시키고 결과 Log를 분석하면 시험을 완료하게 된다.

표 1.의 소프트웨어 자동화 테스트 프로세스는 휴대폰, DVDP, PDA 제품을 시험하면서 정착 되었으며 자동화율 및 Test Coverage 측정의 결과 기준은 제품별 특성에 따라 Test Spec.을 기준으로 측정 되었다.

Process	Output
1) System Analysis	Test Plan
2) Software Analysis	Test Requirement Software Menu Tree Analysis of Critical points in Software
3) Test Spec. Analysis	Design of Test Scenario A Standard of Test Results
4) Constructing Test Environment	Design of Test System System I/O Refined
5) Writing and Operating Scripts	Test Implementation
6) Software Reliability Testing	Software Stress, Repetitive, Aging test Power On/Off Test
7) Results Analysis	Fault Detection rate, Test Coverage Automation

표 1. 소프트웨어 자동화 테스트 프로세스

2. Previous condition for automated software test

Embedded system software 자동화 시험을 위한 선결 조건은 다음과 같은 몇 가지로 나누어 설명할 수 있다.

- 1) Test Spec. 및 Checklist는 필수적으로 준비되어 있어야 한다.
- 2) 시험대상 시스템의 입출력이 명확히 정의되어 있어야 한다.

첫째, 해당 제품의 소프트웨어 테스트 Spec.과 결과 기준이 명확해야 하며 시험에 필요한 시스템 Spec.과 software Spec.을 기본으로 한 정확한 Test Spec.이 필요하다. 둘째, 정확한 입출력 정의를 통해 시험환경을 구성할 수 있다. 이는 소프트웨어 자동화 테스트의 가장 중요한 부분으로서 해당 시스템의 입출력 정의는 Test Coverage 및 자동화에 많은 영향을 준다. 실제로 휴대폰의 경우 전체 테스트 프로세스의 45% 이상을 시험환경 구성에 투자하였고 83%의 가장 좋은 자동화율과 90%이상의 Test Coverage를 확보할 수 있었다.

3. 소프트웨어 자동화 테스트 기법

3.1.1 System Analysis

보다 높은 Test coverage와 자동화율의 결과를 얻기 위해서는, 시험을 하기 전에 해당 제품의 소프트웨어 기능 특성은 무엇이며 어떤 특징을 갖는지, 하드웨어와의 인터페이스는 어떠한 지 등의 제품에 대한 연구 및 분석이 필요하다.

소프트웨어 기능 구조 및 시스템 전체 기능에 영향을 미치는 소프트웨어의 모듈

제품	입출력	테스트 종류
휴대폰	Keypad/LCD	Intrusive
DVD Player	Remote Control/S-Video	Non-Intrusive
PDA	Touch Screen, USB etc.,	Non-Intrusive

표 2. Defined I/O of Products

대한 연구가 선결되어야 이를 바탕으로 시험의 깊이가 결정된다. 휴대폰, DVDP, PDA는 표2.와 같은 특성 및 기능들을 가지고 있다.

3.1.2. Software Analysis

소프트웨어 자동화 테스트는 Black-Box Test 기법을 사용하므로 소프트웨어 내부 프로그램 자체 내용은 중요하지 않다. 단, 시험 대상 시스템의 소프트웨어가 어떤 구조로 구성되어 있으며 어떤 상태 천이를 갖고 어떤 제어 논리를 구현하는지에 대한 정보 분석을 통해 해당 제품의 menu tree diagram이나 그림1.과 같은 state diagram이 직접 test script 구성에 유용하게 사용된다.

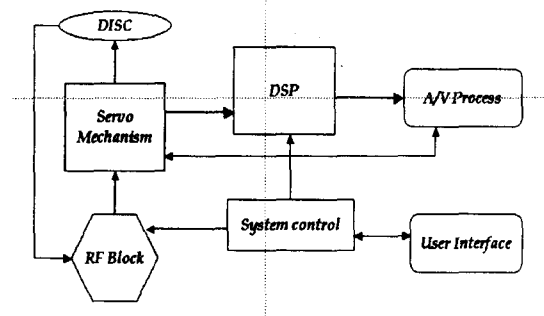


그림 1. State Diagram

3.1.3. Analysis of Software Test Specification

Test Spec.은 제품별로 그 형태나 종류가 매우 다양하다. DVDP의 경우, NF002의 NTSC 및 PAL 방식에 따라 표준 Test Navigation Disc.를 표준 reference로 시험을 하였으며 휴대폰 및 PDA는 소프트웨어 Menu Tree에 따라 그 영역을 한정지어 Test Spec.을 구성하였다.

3.1.4. Constructing Test Environment

휴대폰 LCD의 경우, 2 gray, 4 gray로 발전하면서 현재 256 색상, 65000 색상, 그리고 TFT-LCD에 이르기까지 급변하였다. 이런 LCD는 소프트웨어 자동화 테스트 환경을 구축하는데 많은 걸림돌이 되었다. 그림2.는 이러한 LCD의 system Integration 모습을 보여주고 있다.

Integration 작업은 크게 입력 부분의 Keypad와 출력 부분의 LCD 부분으로 나누어 keypad driver, LCD driver에 직접적 연결을 통해 RS-232C 통신으로 신호를 호스트 PC에서 제어할 수 있도록 구성하였다.

이러한 Intrusive 방법은 보다 높은 자동화율과 Test coverage를 확보하는데 큰 도움이 되었다.



그림2. 휴대폰 테스트

3.1.5. Writing and operating automated software test scripts

그림3.은 휴대폰의 자동화 테스트 scripts의 실례이다. 자동화 스크립트의 구성 원칙 및 동작 요구사항은 몇 가지를 들 수 있다.

- 1) 자동화 Test scripts의 처음과 끝이 일정해야 한다.
- 2) Test 시나리오에 대한 Pass/Fail Log에 따라 결함 지점을 검색하는데 어려움이 없어야 한다.
- 3) 사용자 정의 함수 및 단일 시험 기능 함수들이 random하게 동작할 수 있어야 한다.
- 4) 자동화 Test scripts 동작은 시스템

타이밍에 민감하므로 정확한 실시간 제약 조건을 수행해야 한다.

위와 같은 사항들은 자동화 테스트 툴을 실제 제품에 적용하여 얻어낸 정보들이며 Embedded system에서 자동화 테스트를 구현할 경우 꼭 고려해야 할 사항들이다.

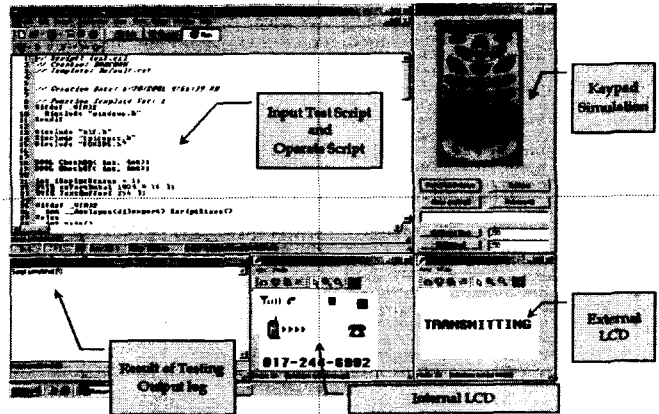


그림3. 자동화 테스트 스크립트

3.1.6. The example of Software reliability testing

소프트웨어 신뢰성 시험은 대부분 단순 반복적인 시험 기능을 무한 반복 수행하여, 소프트웨어 결함보다는 시스템 결함을 찾아내는데 그 목적이 있다. 우리는 휴대폰, DVDP, PDA의 Power on/off 및 기타 단순 반복 기능에 대해 무한 반복 테스트를 구현하여 시스템이 다운되거나 소프트웨어의 특정 기능에 결함이 발생하는 현상을 관찰할 수 있었다. 그 이유는 Embedded System의 경우 하드웨어와 소프트웨어의 기능이 논리적으로 완전히 분리된 것이 아니라 상호 연동 (Interlocking)되어 최종 기능을 발휘하기 때문에 긴밀한 관계 Interface 상태에 있다는 것을 간과하지 말아야 한다. 다음은 간단한 Power on/off 예제이다.

```

int Power_On_Off_Testing_DVDP()
{
    for(;;)
    {
        ir_SendSignal(Power);
        verify;
        delay(time);
        ir_SendSignal(Power);
        verify;
        delay(time);
    }
    return 0;
}

```

3.1.7 Results Analysis

자동화 테스트 결과에 대한 분석은 다음과 같은 지표로 완성된다. /우리는 휴대폰, DVDP, PDA에 대한 자동화율 및 Test coverage 그리고 결함 검출율에 대한 지표를 분석하여 자동화 테스트 효율을 도출하였다. 식1), 2), 3)은 제품별 혹은 모델별로 그 지표가 바뀔 수 있으며 지표가 세분화될수록 보다 정확한 통계 지표로 활용될 수 있다.

식 1) 자동화율

$$= (\text{자동화 가능 건수} \div \text{Test 건수}) \times 100$$

제품	Test 건수	자동화 건수	자동화율
휴대폰	1580	1311	82.97%
DVDP	168	92	54.76%

표 3. 자동화율

식 2) Test Coverage

$$= (\text{Test 건수} \div \text{Test Specification}) \times 100$$

제품	Test Spec. 건수	Test 건수	Test Coverage
휴대폰	1755	1580	90%
DVDP	184	168	91%

표 4. Test Coverage

식 3) 결함 검출율

$$= (\text{결함건수} \div \text{Test 건수}) \times 100$$

제품	Test 건수	결함 검출 수	결함 검출율
휴대폰	1580	47	2.97%
DVDP	168	8	4.76%

표 5. 결함 검출율

4. Conclusion

Embedded system software의 품질 향상을 위한 소프트웨어 자동화 테스트 기법의 적용은 표6.과 같은 시간 단축율의 정량적 결과 뿐 만 아니라, 개발 단계 중 소프트웨어 버전 변경에 따른 에러 파급 영향의 잠금 (Locking) 장치로써 많은 효과를 볼 수 있었다.

Division	비 자동화	자동화	시간 단축율
Computer	2700hr	1260hr	53%
DVDP	1120hr	422hr	62%
Visual Media	320hr	210hr	34%

표 6. 테스트 효율 측정

또한 이러한 자동화 테스트 기법의 적용은 테스트 인력의 효율적 활용과 테스트 기술의 고급화라는 비가시적 효과를 가져온다. 소프트웨어 자동화 테스트 기법을 적용하면서 만일 제품에 정의된 입출력을 상용 틀이 대처하지 못하면 테스트 효율 및 시간에 대한 노력이 많이 소요될 것이다. 또한 Test reference의 데이터베이스(DB)화하는 시간이 많이 소요되기 때문에 개선이 필요했다. 현재 우리는 보다 많은 디지털 제품에 소프트웨어 자동화 테스트 기법을 적용하기 위해 많은 tailoring 작업을 시도하고 있으며 보다 많은 기능 결함들을 검출하기 위해 abnormal test case들을 수집, 분석하고 있다.

참고문헌

- [1] 한상섭(2001), Report of Automated Software Test for Mobile Phone in Samsung Electronics
- [2] 한상섭(2000), Report of Automated software test for DVDP in Samsung Electronics
- [3] 한상섭(2002), Report of Automated software test for PDA in Samsung Electronics
- [4] Elefriede Dustin, Jeff Rashka, John Paul(1999), Automated SoftwareTesting, Addison-Wesley
- [5] Cem Kaner, Jack Falk, Hung Quoc Nguyen(1999), Testing Computer Software second edition, Wiley
- [6] Ron Patton(2001), Software Testing, SAMS