

Embedded System의 효율적 TEST 방법에 대한 연구

*김성근, 정기현
아주대학교 전자공학과
e-mail : stradivarius@dreamwiz.com

A Study on Efficient Test Method of Embedded System

*Seong-Gun Kim, Ki-Hyun Chung
School of Electronics Engineering
Ajou University

Abstract

As an Embedded system get more complicated, the importance of the software test increases. Especially it is mandatory to test the embeded software for a charge degrade of the development company which is happened due to the error, a danger element clearance to the customer, a performance enhance of the system. In this paper, we propose efficient test method which discovers the bug an embedded system using suitable test case selection and automation test.

I. 서론

임베디드 시스템의 복잡화가 가속화 되면서 소프트웨어의 테스트의 중요성은 점점 강조되어 왔다.

소프트웨어 테스트의 목적은 버그 발견, 사용자와 회사가 입게 될 위험의 감소, 개발 유지비용 감소, 시스템의 성능향상 등이 있다.

이 중에서도 개발 유지비용 측면에서 프로그램의 버그를 빨리 발견 할수록 수정하는 데 드는 비용은 줄어든다. 1990년 HP 에서는 전체 R&D 예산의 1/3 을 제품의 시장 출시 이후 버그를 수정하는 재개발 비용과

시장 Service 비용으로 지출했다.

이상적인 소프트웨어 테스트는 프로그램의 모든 가능한 동작을 테스트 하는 것으로 목표는 바람직 하지만 무한개 조합의 테스트를 해야 하므로 제한된 기간과 인력으로 테스트를 해야 하는 실제와는 거리가 멀다.

본 논문에서는 한정된 자원과 비용으로 효과적인 소프트웨어 테스트를 위한 테스트 케이스 선정방법과 자동화 테스트 방법에 초점을 맞추어서 기술한다.

II. 본론

2.1 적합한 Test Case 의 선정방법 제안

테스트 케이스를 만드는 다양한 방법이 있지만 크게 2가지 기본적인 방식이 있다.

기능 테스트(블랙박스 테스트)는 구현된 것이 요구하는 사양에 얼마나 잘 맞는지를 평가하는 테스트 방식이고, 적용범위 테스트(화이트박스 테스트)는 코드의 특정 부분이 수행하도록 하는 경우를 선택한다.

기능 테스트는 기능 명세서, 동작 사양서등으로 작성한다. 최초의 테스트 케이스의 문제점은 Total 테스트 케이스 수가 너무 많고, 내용이 방대하여 테스트시 복잡하고 시간이 많이 소요되었다. 이에 중요도와 시간을 고려하여 A, B, C 등급으로 테스트 케이스에 등급

을 부여하고, 기능 항목을 구분 통일하여 정리하고, Description 항목의 내용축소 및 간략화를 실시하고, 동작군 별로 나누어 중복 기능을 통합하였다. 그 결과 테스트 케이스의 수가 56% 이상 감소되어 테스트에 걸리는 시간을 반으로 단축하였으며, A등급 점유율이 23% 로 주요항목에 대한 테스트를 단시간에 효율적으로 실시할 수 있게 되었다.

제품군		Total T/C 항목 수(EA)			A등급 점유율(%)	
		변경전	변경후	감소율	변경전	변경후
Video	MODEL 1	6,427	2,789	56.6%	54.0	20.7
	MODEL 2	6,591	3,176	51.8%	53.5	22.0
	MODEL 3	8,769	3,737	57.4%	57.0	23.0
	MODEL 4	1,636	469	71.3%	48.5	24.7
	소계	23,423	10,171	56.6%	53.3	22.6

그림 1. 기능 테스트 최적화

적용범위 테스트는 IBM Rational Test Real Time Tool 을 사용하여 Static Analysis, Component Test를 수행하고, 기준에 미달되는 Module에 대하여 Code 수정작업을 실시하였으며 기능 테스트에서 발견하지 못한 소프트웨어 버그를 수정할 수 있었다. 소프트웨어 개발자들이 직접 Tool을 사용하게 함으로써

Static Analysis는 기본 Coding Rule 준수여부와 함수의 복잡도를 확인하는 테스트로 Review로 Error 11 개 항목 수정 및 Warning 항목을 최소화로 변경하였으며, Test Metrics 로 복잡도가 20이상인 함수의 분리하는 Code 수정 작업을 실시하였다.

Component Test는 함수가 올바르게 구현되었는지 확인하는 테스트로 Code Coverage를 80% 이상으로 선정하여 만족하지 못하는 부분을 수정하였다.

적용범위 테스트 적용한 것과 적용하지 않은 동일 모델을 동일 조건으로 테스트 결과 문제점이 35% 감소한 것으로 확인 되었다.

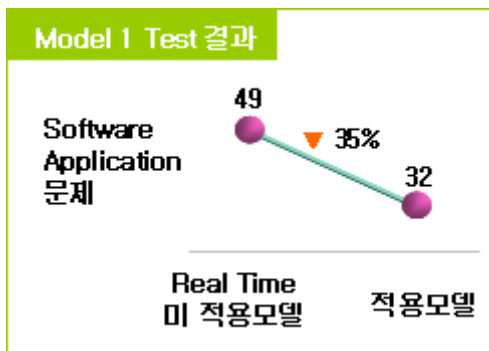


그림2. 적용범위 테스트 적용전후 문제점 수

Program (Auto Sequence Test System)등을 이용하여 테스트 자동화를 구현하였다. ASTS Program은 앞에서 작성된 테스트 케이스를 기초로 하여 테스트 스크립트를 작성하여 정해진 시퀀스에 맞추어 PC에 연결된 리모컨이나 프로세서에 직접적으로 키 값을 입력하는 방식으로 자동화 테스트를 실행하게 된다. 테스트 자동화율은 75% 정도에 달한다. 실행된 테스트 결과는 카메라나 Capture Card로 PC에 저장된다. 시험원은 자동화 테스트 완료 후, PC에 저장된 Test Log 화일과 저장된 화면을 보면서 결과를 비교한다. 저장된 파일을 2X, 4X, 6X 배속으로 빨리 결과를 확인 할 수 있다. 이것은 테스트 인원의 반복적인 테스트 시간 자체를 제거하는 효과와 더불어 확인하는 시간을 감소시킬 수 있으며, Test Log 파일을 통해 에러 발생 시점과 정확한 Mode를 알 수 있게 된다.

III. 결론 및 향후 연구 방향

적합한 테스트 케이스 최적화 선정방법과 만들어진 테스트 케이스의 75% 정도를 자동화 테스트로 진행함으로써 약 50% 이상의 테스트 비용과 시간을 감소시킬 수 있었다.

향후에는 본 논문에서 제시한 효율적인 테스트 방법을 도구로 하여 테스트 케이스 작성시, 시장 주요 문제점의 케이스를 추가하고 자동화 테스트 비율을 높여 차후 테스트 모델에 적용할 것이다. 또한, 자동화 테스트에서 결과로 녹화된 화면과 기대된 화면과의 비교 프로그램을 개발하여 테스트 결과 리포트를 Output으로 받아 확인 작업의 Loss를 줄여 보다 진보된 System으로 발전시킬 예정이다.

참고문헌

- [1] Ron Patton : Software Testing 2005
- [2] Paul C. Jorgensen, Software Testing - A Crafts-Man's Approach, CRC Press 1995
- [3] Arnold S. Berger : Embedded System Design
- [4] MISRA-C:2004 standard.
- [5] IBM Test RealTime Code Review Manual_V1.1

2.2 자동화 Test 방법 제안

PC, 카메라, TV Capture card, UIRT 리모컨, ASTS