

# 임베디드 소프트웨어 테스트 품질에 관한 연구

장선재, 김행곤  
대구가톨릭대학교 컴퓨터정보통신공학부  
e-mail : {jsjsid, hangkon}@cu.ac.kr

## A Study on Quality of Embedded Software Testing

Seon Jae Jang, Haeng Kon Kim  
Dept. of Computer Information & Communication Engineering

### 요 약

임베디드 시스템들은 빠르게 다양한 분야에 적용되고 있다. 이들 시스템들의 품질을 좌우하는 것은 소프트웨어이며, 제품 출시 시기와 제품의 품질 등에 매우 중요한 요소가 된다. 임베디드 소프트웨어 테스트는 이러한 임베디드 소프트웨어에서 오류가 없음을 증명하는 것으로 전체 소프트웨어 개발에서 많은 부분을 차지한다. 임베디드 소프트웨어 테스트는 소프트웨어의 기능적 부분을 주로 평가하며, 사용자가 필요로 하는 품질에 대한 평가 부분은 미흡하다.

본 논문에서는 임베디드 소프트웨어 테스트 도구를 이용하여 임베디드 소프트웨어의 품질을 평가할 수 있는 방법에 대해 연구한다.

### 1. 서론

임베디드 시스템들에 대한 사용자들의 요구 및 빠른 시장변화에 대응하기 위해, 다양한 분야에서 임베디드 시스템들이 빠르게 개발되고 있다.

이는 임베디드 시스템에 필수적인 부품들의 저가화 및 고성능화로 임베디드 기술의 적용이 보다 많은 분야에 적용 가능하게 되었기 때문이다. 임베디드 시스템의 성능은 고성능화 되었으며, 동일 분야에서 활동하는 여러 업체들 사이에서 비슷한 성능을 가지는 시스템들이 개발되고 출시되고 있다.

이러한 발전은 시스템들에 탑재 되는 임베디드 소프트웨어의 중요성을 더욱 증가시키고 있다. 하지만 하드웨어의 빠른 발전에 비해 소프트웨어의 발전은 느리며 또한, 기존의 기능 및 비기능에 비해 발전되는 하드웨어의 독선에 따라 복잡성이 매우 증가되었다. 복잡해진 임베디드 소프트웨어를 테스트하기 위해서는 테스트하는 기술 또한 높은 수준이 요구된다.

최근의 임베디드 소프트웨어를 테스트 하는 도구의 성능은 날로 향상되고 있으나, 일반적으로 테스

팅 도구들은 임베디드 소프트웨어를 기능적인 부분에서 분석하고 나타내는데 중점을 두고 있다. 따라서 관심이 증가되고 있는 임베디드 소프트웨어의 품질을 효과적으로 평가하는 기능은 많이 미흡하며, 다양한 분야에 적용되기 위한 적용방법이 요구된다.

본 논문에서는 강조되고 있는 임베디드 소프트웨어의 품질을 효과적으로 평가하기 위해 요구되는 사항들을 분석하고 이를 기반으로 효과적으로 임베디드 소프트웨어를 평가하기 위한 효율적 방법에 대해 논한다.

### 2. 관련연구

#### 2.1 임베디드 소프트웨어 테스트

임베디드 소프트웨어 테스트이란 임베디드 소프트웨어를 사전에 검사하여 오류를 발견하는 작업으로, 오류를 사전에 발견하고 해결하여 제품의 질을 높이는 작업이다. 표 1과 같이 임베디드 소프트웨어 테스트를 지원하는 도구들은 다양하게 출시되고 있다. 하지만 기존의 임베디드 소프트웨어 테스트 도구

<표 1> 임베디드 소프트웨어 테스트 도구들

| 종류            | 특징  |
|---------------|---|
| Bashan ERTT   | 제작된 모바일 폰과 같은 배움의 질을 향상시키기 위한 실시간 임베디드 테스트 툴  |
| CodeScroll    | 자동화된 화이트 박스 테스트를 사용하는 임베디드 테스트로 비정상 상황에 대한 대처를 시험하는 테스트 케이스 추가 적용   |
| CodeTEST      | 복잡한 S/W 시스템들에 가시성을 제공 S/W 질 향상, 개발 시간을 위한 임베디드 테스트 제품으로 JAVA와 C 용으로 나누어져 있음                                 |
| RESORT        | S/W Metrics 기반으로 프로그램의 기능, 구조, 행위, 등의 분석을 위한 임베디드 테스트 제품으로 JAVA와 C 용으로 나누어져 있음                              |
| TESSY         | 임베디드 S/W Metrics 기반으로 프로그램의 기능, 구조, 행위, 등의 분석을 위한 임베디드 테스트 제품으로 JAVA와 C 용으로 나누어져 있음                         |
| VectorCAST    | Ada, C/C++와 임베디드C++(EC++) 소스코드를 검색하고 호스트와 임베디드 환경을 위한 executable test harnesses를 제작하는데 필요한 테스트 코드를 자동적으로 생성 |
| TestComplete3 | 가벼운 모든 환경에서 사용이 가능하며 낮은 가격에 전기계 테스트와 다변경 가능   |
| Qengine 4     | 자비용으로 특화된 테스트 툴, 다양한 플랫폼 지원   |

<표 3> ISO/IEC 9126 기반 품질평가 사항

| 주특성   | 부특성   | 적용                       |
|-------|-------|--------------------------|
| 가능성   | 적용성   | 다양한 타겟 보드에 적용 가능 여부      |
|       | 정확성   | 테스트 결과의 정확성              |
|       | 상호운용성 | 타겟보드의 상호운용성              |
| 신뢰성   | 보안성   | 테스트 결과의 보호               |
|       | 일관성   | 타겟 보드의 오류, 사용미숙 등        |
|       | 유연성   | 오류 발생시 작업 복구 또는 결과 보호    |
| 효율성   | 시간반응성 | 실시간 정보 처리                |
|       | 자원효율성 | 타겟 보드의 CPU, 메모리, 전력 사용량도 |
|       | 이해성   | 테스팅 도구의 이해 여부            |
| 사용성   | 응답성   | 테스팅 도구의 사용자 인터페이스의 용이성   |
|       | 유용성   | 도구 활용의 용이성               |
|       | 직관성   | 사용자가 도구의 사용 방식에 잘 적응하는가? |
| 역시성   | 적용성   | 다양한 기기에서 사용 가능한가?        |
|       | 실시성   | 실지가 용이한가?                |
|       | 공존성   | 타겟 내의 어플리케이션과의 충돌은 없는가?  |
| 유지보수성 | 내재성   | 기존의 제품과의 차이점은 무엇인가?      |
|       | 해석성   | 도구 자체의 오류 진단의 가능 여부      |
|       | 변경성   | 도구의 변경이 용이한가?            |
|       | 안정성   | 변경 후의 도구에 발생하는 문제 최소화    |

들은 주 기능이 기능적인 부분을 평가하는데 있어 최근 관심이 증가하고 있는 품질에 관한 평가는 미흡한 수준이다.

2.2 품질평가

임베디드 시스템들에 대한 사용자들의 요구가 증가함에 따라, 자연히 탑재되는 임베디드 소프트웨어에 수준이 높아지면서 소프트웨어에 대한 품질이 중요한 부분으로 대두 된다. 임베디드 소프트웨어는 일반 패키지 소프트웨어와는 다른 여러 특징들을 고려하기 위해서 다양한 특성을 고려하고 반영하여야 한다.

국제표준 ISO/IEC 9126은 표 2와 같이 품질평가 사항들을 나타내고 있다.[1]

소프트웨어 품질을 평가함으로써 프로젝트에 사용된 자원과 성능에 대한 품질을 보증하여 최종 제품에 대한 성공을 확신하고, 사용자에게 제품의 품질을 보증하는 구체적인 자료를 제시할 수 있다.[2]

3. 요구사항 분석

3.1 개요

소프트웨어 품질을 평가하기 위한 기본적인 사항을 ISO/IEC 9126 기반으로 정리하면 표 3과 같다.

표 3에서 나타나는 사항들은 기본적인 임베디드 소프트웨어를 위한 품질을 나타내며, 임베디드 시스템들에 따라 제시되는 특성 중에서 일부를 선택적으로 사용할 수 있다.

<표 2> 품질평가 요구사항

| 특성       | 내용  |
|----------|---|
| 신뢰성      | 어디웨어와 연동하여 동작하기 때문에 오류발생 시 적용환경에 따라 큰 문제점을 발생시키며 수정하기 어려움           |
| 역시성      | 어디웨어 플랫폼의 적용 변동에 의한 다양한 플랫폼 지원과 소프트웨어 변경이 쉽고 간단하게 이루어져야 함           |
| 재사용성     | 신제품 개발 주기가 빨라지는 문제점에 대한 실시간으로 반응, 동작을 수행해야 함                        |
| 실시간성     | 내부 또는 외부에서 발생하는 이벤트에 대해 실시간으로 반응, 동작을 수행해야 함                        |
| 호환성/확장성  | 호환성 사용됨으로 기존 임베디드 소프트웨어에 대한 호환과 기능확장이 용이해야 함                        |
| 어디웨어 최적화 | 소프트웨어와 어디웨어가 동시에 개발되므로 서로간에 최적화가 이루어져야 함                            |
| 강한 내구성   | 외부 요인에 의해 일부 기능에 이상이 생겨도 기본적인 중요 기능은 동작할 수 있어야 함                    |
| 최소형/소경량  | 임베디드 시스템은 주로 System-on-chip 기술이 적용되고 있으며, 소프트웨어는 이에 최적화되어 개발되어야 함   |
| 사용성      | 임베디드 시스템은 주로 일반 사용자들을 대상으로 하는 경우가 대부분으로 정장 지식 없이도 사용할 수 있는 편리성이 요구됨 |
| 상호운용성    | 여타의 임베디드 시스템과 다른 임베디드 시스템과 연결되어 작동할 때 이상 없이 작동하여야 함                 |

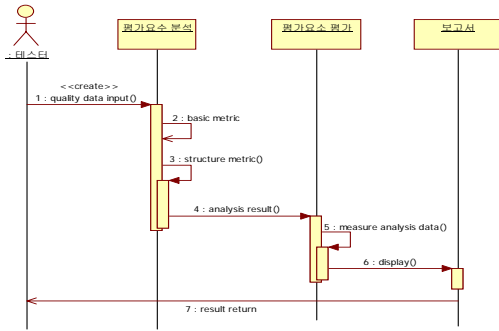
임베디드 소프트웨어의 품질평가를 위해서 품질 검사표가 사용되는데, 품질검사표는 품질평가 수행 시 필요사항을 추출하여 요약한 것이다. 표 4는 품질평가에 사용되는 품질평가표의 예를 나타낸다.

임베디드 소프트웨어 테스트 도구들은 지금도 다양하게 연구·개발되고 있으며 사용되는 분석방식 또한 다양하다. 테스트 도구들에서 균일한 평가 결과를 제시하기 위해서는, 평가에 사용될 테스트 도구를 분석하고 이를 품질평가에 반영할 수 있어야 한다. 이는 모든 도구에서 공통적으로 발견하고 수집할 수 있는 사항과, 테스트 도구 고유의 특징들로 분류할 수 있다.

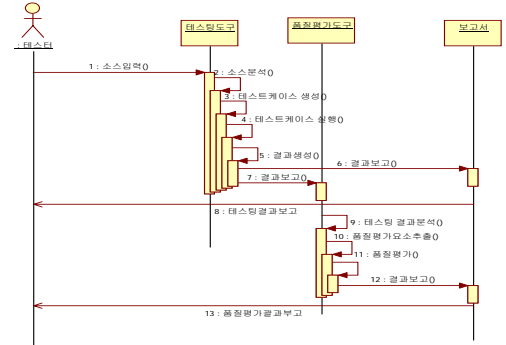
다양한 테스트 도구들에 적용되기 위한 방법 또한 중요하게 고려할 사항이다. 임베디드 테스트는 동일한 분야에서도 다양한 테스트 도구나 방식이 존재하기 때문에, 테스트 결과를 수집하여 분석하는 방법이 필요하다. 본 논문에서는 분산된 테스트 도구들에서 품질을 평가하는 자료를 수집하기 위한 방법으로 크게 4가지 예를 제시한다. 품질평가를 위한 방법으로는 품질평가 기반의 임베디드 소프트웨어 테스트 도구를 개발하여 사용하는 방법, 웹을 이용하여 사용자가 직접 정보를 입력하는 방법, 사용자의 컴퓨터에 직접 연결되어 테스트 결과를 바로 수렴하는 방법, 마지막으로 등록된 테스트 도구들에서 자동적으로 품질평가 시스템이 테스트 결과를 수집하여 평가하고 다시 사용자에게 전송하는 방법 등을 들 수 있다.[3,4]

<표 4> 품질평가표

| 예외사항    | 임베디드 시스템이 제안 받고 있는 경계값의 범위를 벗어난 입력에 대한 예외처리를 하고 있습니까?   |
|---------|---|
| 경계값 처리용 |   |
| 측정 항목   | A 경계값 확인 대상 항목 수<br>- 사용자 문서에 명시된 경계값에 대한 테스트케이스를 작성  |
|         | B 각 항목별 테스트케이스 성공률의 합 각 항목별 테스트케이스 성공률의 합<br>- 테스트케이스에 따라 테스트를 수행하여 발생과 일치하는 비율   |
| 계산식     | - 경계값 처리용 (BEC) = B/A<br>- $B = \sum_{i=1}^n \frac{Success_{TC_i}}{Total_{TC_i}}$<br>- Success_TC : i 번째 경계값 처리 기능 확인을 위해 수행한 테스트케이스 중 성공한 건 수<br>- Total_TC : i 번째 경계값 처리 기능 확인을 위해 수행한 테스트케이스 수 |
| 결과 영역   | 0 ≤ 경계값 처리용(BEC) ≤ 1  |
| 판계법     | 결과값   |



(그림 1) 품질평가 흐름



(그림 3) 품질기반 임베디드 소프트웨어 테스트 도구

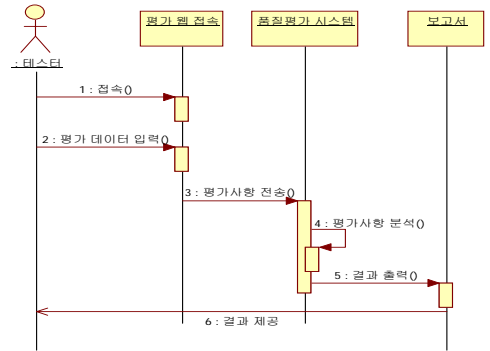
3.2 요구사항 정의

임베디드 소프트웨어의 품질을 분석하기 위해서는 요구되는 품질평가 요소들을 기반으로 테스트 실행결과 및 기타 정보를 기반으로 수행하게 된다. 본 연구에서는 ISO/IEC 9126를 기반으로 기존에 사용되는 임베디드 소프트웨어 테스트 도구들의 테스트 수행 결과를 이용한다.

그림 1은 품질평가의 흐름을 나타낸다. 평가 도구는 임베디드 소프트웨어 테스트 도구에서 자료를 받아들인다. 입력되는 자료는 요구사항에서 작성된 품질평가에 적용되어 최종적으로 임베디드 소프트웨어에 대한 품질을 평가한다.

임베디드 소프트웨어 품질 검사를 위해서는 요구사항 분석에서 제시된 것처럼 4가지 방법을 예로 들 수 있다. 그림 2는 제시된 품질 방법들의 예를 나타낸다.

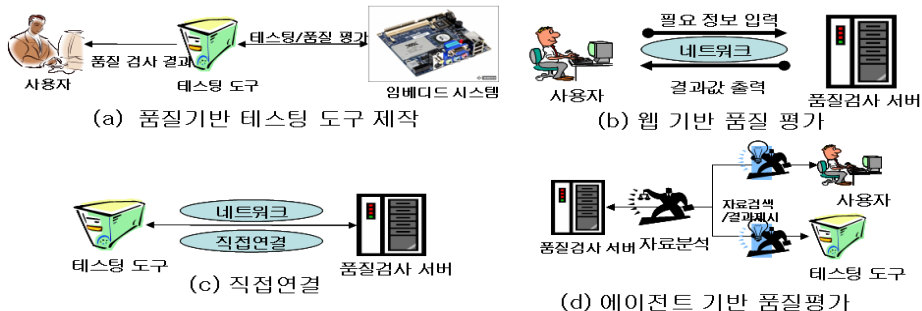
- 품질기반의 임베디드 소프트웨어 테스트 도구를 개발하는 경우, 검증 방식이 통일되어 품질을 정확하게 판단할 수 있지만, 지원되지 않는 시스템들이나 언어에 대해서는 평가 할 수 없다. 그림 3은 제시된 방법의 흐름을 나타낸다.
- 웹을 이용하는 방법의 경우 사용자가 직접 평가 시스템에 접속하여 평가에 필요한 요소들을 체크하고 선택된 사항들에 대한 값을 입력하여 결과를 확인하는 방법이다.



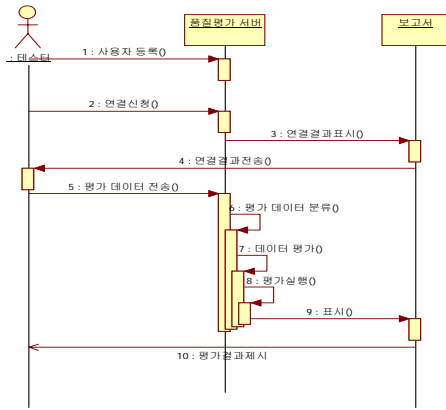
(그림 4) 웹 기반 품질평가

그림 4는 웹 기반 방법의 흐름을 나타낸다. 이 방법은 다양한 테스트 도구에서 사용된 결과를 손쉽게 적용시킬 수 있으나, 테스트 도구들의 테스트 케이스 생성 방식에 대한 분석이 요구된다. 또한 직접 요구값을 직접 입력해야 한다.

- 테스트 도구가 품질평가 시스템이 연결되어 있는 경우 테스트 결과를 바로 평가 시스템에 전송하여 결과를 얻어낼 수 있다. 또한 평가 사항을 미리 입력할 수 있기 때문에 웹에 접속하여 사용하는 방법보다 편리하다. 하지만 하나의 테스트 도구와 연결되며, 사용자의 컴퓨터가 항상 평가 시스템과 연결되어 있어야 한다. 그림 5는 제시된 방법의 흐름을 나타낸다.



(그림 2) 품질평가 방식의 예



(그림 5) 직접연결된 품질평가



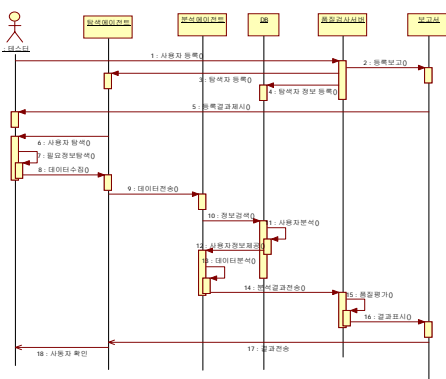
(그림 8) 웹 기반 품질검사 과정의 예

그림 8은 그림 7을 기반으로 작성된 품질평가 시스템의 예이다. 사용자는 우선 시스템에 등록 하고 품질검사에 사용될 목록을 설정한다. 설정 후 각각의 설정항목에 값을 입력시키면 시스템에서 이를 분석하고 품질 검사 결과를 사용자에게 제시한다.

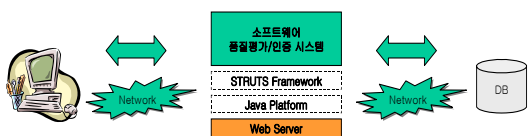
- 등록된 테스트 도구들에서 자동적으로 품질평가 시스템이 테스트 결과를 수집하는 방법 중 대표적인 방법은 에이전트를 사용하는 방법이다.

그림 6은 에이전트를 이용한 경우의 흐름을 나타낸 것이다. 품질평가 시스템이 등록된 사용자의 컴퓨터나 서버에 접속하여 필요한 정보를 입수하여 분석하고 결과를 다시 전송한다. 제안된 방법의 경우, 테스트 도구에 대한 이해 및 자료를 수집하고 전송하는 에이전트의 작동 방식에 대한 이해, 수령한 자료의 분류 및 품질검사 방식을 정의하여야 한다.

그림 7은 평가방법 중 하나인 웹 기반 평가 시스템의 구조를 나타낸 것이다. 시스템은 웹서버를 기반으로 소프트웨어 품질평가/인증 시스템이 작동하는 구조로 되어 있다.



(그림 6) 에이전트 기반 품질평가



(그림 7) 웹 기반 품질검사 구조

#### 4. 결론 및 향후연구

본 논문은 임베디드 소프트웨어의 품질평가를 위해 테스트 도구를 이용한 품질평가 방법을 제시하였다. 보다 정확한 품질을 평가하기 위해서는 다양한 테스트 도구에 대한 검사 방식에 대한 평가 및 각 도구들의 특징적 사항들의 수렴, 각 테스트 도구의 결과 생성 방식과 필요정보 수집능력이 요구된다.

향후, 다양한 검사 방식들에 대한 연구를 및 이를 바탕으로 품질기반의 임베디드 소프트웨어 테스트 도구의 개발이 연구될 것이다.

#### 참고문헌

[1] 배현섭 "텔레매틱스 분야의 임베디드 소프트웨어 테스트" 월간 임베디드월드, 슈어소프트테크, 2005  
 [2] B. Littlewood, L. Strigini, "Software reliability and dependability: A roadmap," The Future of Software Engineering. ACM Press, New York, 2000.  
 [3] J. Costa. s. Devadas, and J. Monteiro, "Observability analysis of embedded software for Coverage-Directed validation." In Proceedings of the International Conference on Computer Aided Design, pp. 27-32, 2000  
 [4] Ken Deeter, Kapil Singh, Steve Wilson, Luca Filipozzi, and Son Vuong, "APHIDS: A Mobile Agent-Based Programmable Hybrid Intrusion Detection System," Proceedings of MATA 2004, Florian polis, Brazil, October 2004