

# 임베디드 소프트웨어 테스트 성숙도 모델(Emb-TMM) 설계절차 및 사례연구

## (Design Procedure and Case Study for the Test Maturity Model of an Embedded Software(Emb-TMM))

백상훈(Sanghoon beak)\*, 윤희병(Heebyung Yoon)\*\*

### 초 록

최근 임베디드 소프트웨어는 유·무선 통신의 발전 및 디지털 정보기기의 확장, 그리고 이러한 기기들의 컨버전스 등을 통해 그 활용 분야가 급격히 확장되고 있으며 그 기능이나 품질에서도 더욱 완전한 소프트웨어를 개발하려는 노력이 강조되고 있다. 이에 따라 소프트웨어 결함을 조기에 발견하여 임베디드 소프트웨어의 품질을 향상시키려는 소프트웨어 테스트 프로세스에 대한 중요성이 더욱 부각되었으나 고도의 정확성과 실시간 처리가 요구되는 임베디드 소프트웨어에 적용할 수 있는 테스트 프로세스 모델은 없는 실정이다. 따라서 본 논문에서는 임베디드 소프트웨어의 특징과 테스트 프로세스의 특성을 반영한 임베디드 소프트웨어 테스트 성숙도 모델(Emb-TMM) 설계를 위한 절차와 사례연구 결과를 제시한다. 이를 위해 Emb-TMM 설계절차를 참조모델 선정 및 영역 도출, 수준별 영역 분류 그리고 모델 설계라는 세 가지의 큰 부분으로 분류하여 제안한다. 그런 다음 제안한 Emb-TMM 설계절차를 통해 실제로 임베디드 소프트웨어 개발시 어떻게 적용할 수 있는지를 사례연구를 통해 제시한다.

### ABSTRACT

Recently, the application area of an embedded software become larger and larger rapidly due to the development of the wire and wireless communication, the expansion of the digital information device and the convergence of the digital devices and emphasize the effort of the development of more complete software. As a consequence, the importance of the software test process was raised to discover the defects of the software early and improve the quality of an embedded software. However there was no test process model for applying the embedded software which is required the highly precision and the real-time process. In this paper, therefore, we propose the design procedure and case study for the test maturity model of an embedded software(Emb-TMM) which reflects the characteristics of the embedded software and test process. For this, we suggest the three category of the proposed procedure which consists of the selection of the reference model and the derivation of the area, the categorization of the area level, and design model. Then we suggest the case study how the proposed procedure can be applied to the development of an embedded software actually.

Keywords : Emb-TMM, Embedded Software Test Maturity Model, Embedded Software, Test Process

## 1. 서론

임베디드 시스템이란 어떤 제품이나 솔루션 등에 탑재되어 특정 기능을 수행하는 시스템으로 일반적으로 사람의 간섭 없이 독자적인 기능을 수행하는 하드웨어와 소프트웨어의 결합으로 구성되며, 이러한 임베디드 시스템에 내장된 소프트웨어를 임베디드 소프트웨어라 한다. 최근 임베디드 소프트웨어는 유·무선 통신의 발전 및 디지털 정보기기의 확장, 이러한 기기들의 컨버전스 등을 기반으로 하여 그 활용 분야가 급격히 확장되고 있는 중이다.

임베디드 소프트웨어가 많은 분야로 확장되었고 임베디드 시스템에서 차지하는 그 비중도 적용된 대부분의 분야에서 반 이상을 차지할만큼 더욱 더 중요하게 되었다. 또한 임베디드 소프트웨어가 복잡해지고 대규모화됨에 따라 그 기능이나 품질에서도 더욱 완전한 소프트웨어를 개발하려는 노력이 강조되었으며 이러한 노력은 소프트웨어 결합을 조기에 발견하여 임베디드 소프트웨어의 품질을 향상시키려는 소프트웨어 테스트의 중요성을 더욱 부각시켰다.

소프트웨어 테스트 프로세스는 소프트웨어 개발과정에서 매우 중요한 요소이다. 완벽한 테스트를 통해 소프트웨어 시스템의 높은 품질을 보장받을 수는 있지만 현실적인 여건, 즉 시간, 비용, 인력 등을 고려시 이는 불가능한 사실이라 할 수 있다. 따라서 많은 소프트웨어 개발 조직들이 보다 높은 소프트웨어 품질을 획득하기 위해 그들이 갖고 있는 테스트 프로세스에 대한 강·약점을 파악하여 개선사항을 도출하며 이를 통해 조직의 소프트웨어 품질을 향상시키려는 노력을 지속적으로 하고 있다.

일반적인 소프트웨어 테스트 프로세스 개선모델은 TMM(Test Maturity Model)[1][2], TPI(Test Process Improvement)[3][4], TCMM(Test Capability Maturity Model)[5] 등이 있으나 임베

디드 소프트웨어에 대한 테스트 프로세스를 평가하고 개선하기 위한 고유의 모델은 현재 없는 실정이다. 또한 일반적인 성숙도 모델을 구조가 복잡하고 고도의 정확성과 실시간 처리가 요구되는 임베디드 소프트웨어 개발 프로세스에 적용하기에는 어려움이 있다.

따라서 본 논문에서는 임베디드 소프트웨어의 특징과 테스트 프로세스의 특성을 반영한 임베디드 소프트웨어 테스트 성숙도 모델(Emb-TMM) 설계를 위한 절차와 사례연구 결과를 제시하고자 한다. 이를 위해 Emb-TMM 설계절차를 참조모델 선정 및 영역 도출, 수준별 영역 분류 그리고 모델 설계라는 세 가지의 큰 부분으로 분류하여 제안한다. 그런 다음 제안한 Emb-TMM 설계절차를 통해 실제로 임베디드 소프트웨어 개발시 어떻게 적용할 수 있는지를 사례연구를 통해 또한 제시한다. 이를 통해 임베디드 소프트웨어 개발조직에서는 각 조직의 목표에 맞는 테스트 프로세스 성숙도 평가 모델을 설계하고, 문제점을 인식하여 지속적인 테스트 프로세스의 개선에 기여할 수 있을 것이다.

## 2. 관련 연구

### 2.1 임베디드 소프트웨어 특성

임베디드 소프트웨어에 대한 특성은 산업계, 학계 그리고 소프트웨어 관련 기관마다 차이가 있다. LG 전자기술원에서는 임베디드 소프트웨어의 특성을 실시간 처리지원, 고 신뢰성, 최적화 기술 지원, 특정 시스템 전용, 다양한 솔루션과 개발 도구 필요 등으로 구분하였다[6]. 삼성전자 SW센터에서는 발생하는 이벤트에 대한 빠른 응답, 테스트 환경 개발의 어려움, 전력 소모의 최소화, 한정된 메모리, 시스템 설치를 위한 도구 필요 등으로 구분하였다[7]. 이와 관련한 국외 논문에서는 실

시간성, 반응성, 신뢰성, 목적의 한정성, 하드웨어와 통합성, 내구성 등으로 구분하고 있으며[8][9], 한국소프트웨어진흥원에서는 실시간 처리 지원, 고도의 신뢰성 요구, 하드웨어에 최적화, 특정 시스템 실행목적, 고도의 개발 도구 필요 등으로 구분하고 있다[10].

각 분야에서 분류한 임베디드 소프트웨어 특성의 공통점을 분석해 보면 실시간 처리성, 고신뢰성, 하드웨어 통합성, 목적 한정성, 고도의 개발 도구 필요성 등으로 분류될 수 있을 것이다.

## 2.2 임베디드 소프트웨어 테스트 프로세스

IEEE STD 1012-1998에서는 소프트웨어 테스트를 “수동이나 자동으로 시스템을 시험 작동시키고 평가하는 작업으로서 명시된 요구를 만족하고 예상된 결과와 실제 결과와의 차이를 구별하기 위한 것”이라고 정의하고 있다[11].

일반적인 소프트웨어 테스트 프로세스는 요구사항 분석부터 코딩까지 테스트 계획과 설계를 준비하고 코딩 이후 단위 테스트, 통합 테스트, 시스템 테스트, 수락 테스트의 순서로 진행된다. 그러나 임베디드 소프트웨어 테스트 프로세스는 이러한 일반적인 소프트웨어 테스트와는 그 절차가 다르다. 이러한 차이점을 식별하기 위해 국내에서 개발한 임베디드 소프트웨어 개발방법론인 마르미-EM과 유럽의 실시간 임베디드 시스템을 위한 객체지향의 컴포넌트 기반 개발방법론인 DESS를 선정하여 분석하였다.

### 2.2.1 마르미-EM

마르미-EM[12]은 한국전자통신연구원(ETRI)이 2006년에 개발한 제품계열 기반의 임베디드 시스템 개발방법론이다. 이 개발방법론의 특징은 프로세스 경로를 유형별로 분류하여 수행되는 프로젝트의 특성을 반영하였으며 수행 프로젝트의

효율적 개발 활동을 지원하기 위해 각 활동이나 그 특성에 맞는 공정으로 구성되어 있다. 여기서 마르미-EM의 전체 구조는 6개의 개발 경로, 즉 응용개발, 미들웨어 개발, 터미널 개발, 제품 개발, 자산 개발, 제품생성으로 구성되었으며 4개의 작업지침으로 구성되어 있다. 테스트 영역 활동은 테스트 계획, HW 테스트 수행, SW 컴포넌트 테스트 수행, SW/HW 통합 테스트 수행, SW 통합 테스트 수행, 시스템 테스트 수행, 인수/필드/인증 테스트 수행 등 총 9개 작업으로 구성되어 있다.

### 2.2.2 DESS

DESS(Software Development Process for Real-Time Embedded Software Systems)[13]는 벨기에, 체코, 프랑스, 독일, 이탈리아, 네덜란드가 2001년에 공동으로 개발한 실시간 임베디드 시스템을 위한 객체지향의 컴포넌트 기반 개발방법론이다. DESS는 소프트웨어 개발방법론 중 하나인 V-모델을 기본 모델로 하며 UP(Unified Process)의 반복 점진적인 개발 방법을 이용한다.

DESS 개발방법론은 평행적으로 작용하는 실현 워크플로우-V, V&V 워크플로우-V, 요구 관리 워크플로우-V 등 3개의 V로 구성되어 있다. 이중 테스트 관련 워크플로우-V는 V&V 워크플로우-V로 검토, 모델 확인, 컴포넌트 테스트, 통합테스트, 시스템 테스트, 인수 테스트를 수행한다.

## 2.3 테스트 성숙도 평가모델

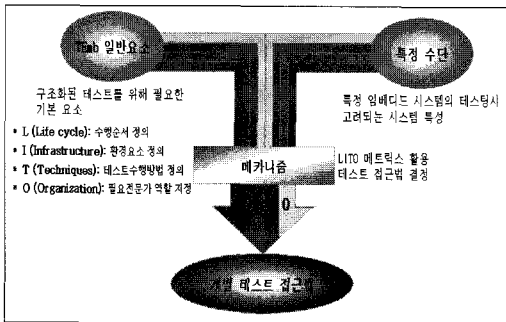
현재 이용되고 있는 테스트 성숙도 평가모델의 구조는 크게 두 가지 형태이다. 첫째는 하나의 영역이 하나의 수준에 적용되는 단계별 형태로 TMM, CMM, TCMM 등이 있다. 각 모델은 각기 5개의 성숙도 수준으로 나뉘어져 있으며 각 단계별 성숙도 목표와 하부목표 등을 정의하고 있다. 둘째는 하나의 영역이 전체의 수준에 적용되는 지

속형 모델로 TPI 모델이 여기에 해당된다. TPI 모델은 3개의 성숙도 수준과 14개의 등급으로 구성되며 각 등급은 각 수준에서 어느 핵심 영역이 개선되어야 할지를 파악하는데 유용한 구조이다.

## 2.4 TEmb(Testing Embedded software) 방법

구조화된 테스트 방법인 TEmb 방법[14]은 4개의 기본요소, 즉 개발주기와 연관된 테스트 활동을 정의한 생명주기(L: Life cycle), 활동을 수행하기 위해 사용 가능한 기술(T: Technique), 적합한 기반구조 및 도구(I: Infrastructure), 바람직한 조직(O: Organization)으로 구성되어 있다. 구조적 테스트 접근법을 형성하는 테스트의 일반적 요소는 테스트 프로젝트 계획, 표준 기법 적용, 테스트 환경 사용, 테스트 팀 조직, 공식 보고서 작성 등이 있다. 이러한 기본요소는 공통적이며 각 테스트 프로세스에서 고려되어야 한다.

테스트 대상에 따라 각각 고려해야 할 사항이 다르므로 기본요소만으로는 테스트를 세부적으로 구현할 수 없다. 따라서 특정 임베디드 시스템에 대해 고려되는 시스템의 특징인 특별 측정치가 결정되어야 한다. TEmb는 테스트 접근에 대한 ‘메카니즘’이라고 할 수 있다. <그림 1>은 TEmb 방식에 대한 개념적인 부분을 나타낸다.



<그림 1> TEmb 개념

TEmb의 생명주기는 어떤 행동들이 수행되어야

하고 어떤 순서를 따르는지에 대한 정의와 테스트 그리고 관리자에게 프로세스에 대한 적절한 이해력을 제공한다. 기법은 특정 활동들을 수행하는 표준방식을 정의하여 수행방식을 지원하며, 기반구조는 계획된 활동들의 수행을 가능하게 만들기 위해 테스트 환경에 필요한 것을 정의한다. 조직은 계획된 활동을 수행해야 하는 사람들의 역할과 요구되는 전문지식, 타 분야와의 상호작용하는 방식 등을 정의한다.

## 3. Emb-TMM 설계절차 제안

### 3.1 Emb-TMM 설계절차

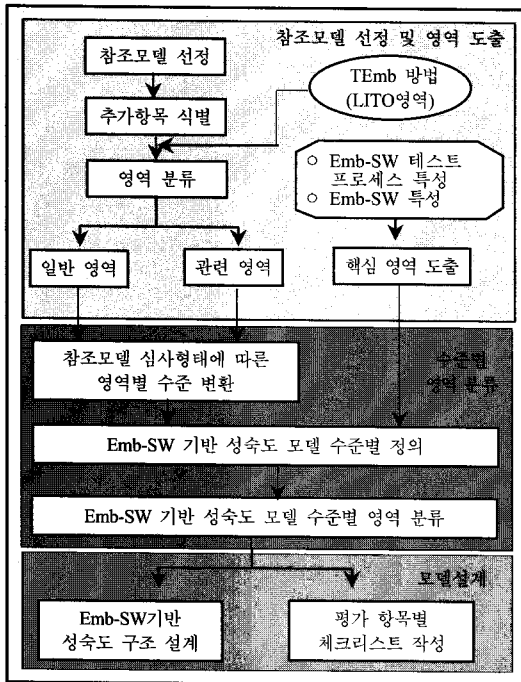
임베디드 소프트웨어 테스트 성숙도 모델(Emb-TMM)을 설계하기 위해서는 임베디드 소프트웨어의 고유 특성과 이를 고려한 최적의 참조모델 선정 및 영역 도출이 우선 고려되어야 한다. 그리고 도출된 영역으로부터 수준별로 분류한 후 이를 모델에 반영하여 최종적으로 Emb-TMM을 설계하는 3단계(참조모델 선정 및 영역 도출 → 수준별 영역 분류 → 모델 설계) 작업절차가 필요하다.

첫 번째 작업은 참조모델을 선정하고 영역을 도출하는 작업이다. 선정된 참조모델을 분석하고 TEmb 방법의 LITO 영역을 적용하여 Emb-TMM 설계시 반영할 추가항목을 식별하고 영역을 분류하여 일반 영역과 관련 영역을 결정한다. 이후 임베디드 소프트웨어 테스트 프로세스의 특성과 임베디드 소프트웨어의 특성에서 핵심영역을 도출한다.

두 번째 작업은 수준별로 영역을 분류하는 단계로 참조모델의 심사형태에 따라 일반 영역과 관련 영역을 영역별 수준으로 변환하고, 임베디드 소프트웨어 기반의 성숙도 모델의 수준별 정의를 내린 후 일반 영역, 관련 영역, 핵심 영역을 성숙도 모

델 수준별 영역으로 분류한다.

세 번째 작업은 모델을 설계하는 단계로 성숙도 모델의 구조를 설계하고 평가 항목별로 체크리스트를 작성한다. 위와 같은 Emb-TMM 설계절차를 도식화하면 <그림 2>와 같다.



<그림 2> Emb-TMM 설계절차

### 3.2 세부 설계절차

#### 3.2.1 참조모델 선정 및 영역 도출

Emb-TMM 설계를 위해서는 먼저 참조모델을 선정해야 한다. 현재 사용되고 있는 패키지 소프트웨어의 테스트 성숙도 모델은 TMM, TPI, TCMM 등이 있다. 테스트 참조모델을 선정하기 위해서는 테스트 모델의 기반구조와 각 모델별 장·단점을 분석하여 참조모델을 선정한다.

참조모델 선정의 기준은 성숙도 수준에 대한 명확한 기준, 조직의 테스트 프로세스에 대한 독립적인 평가 그리고 테스트 프로세스 성숙도 모델 중 높은 완성도를 인정받는 모델이다.

참조모델이 선정되면 참조모델의 영역을 중심으로 유사 모델 영역을 TEmb 방법으로 구분한 후 Emb-TMM 설계를 위한 추가 영역을 식별한다. 참조모델과 참고모델의 영역을 TEmb의 일반요소로 분류하여 생명주기(L), 기법(T), 기반구조 및 도구(I), 조직(O)으로 세분화 한다. 각 모델의 영역들은 일반요소 또는 특정수단에 해당한다. 즉 각 영역이 수행하는 고유의 활동에 따라 구분이 가능하다. 특정수단으로 분류되는 Emb-TMM 영역은 임베디드 소프트웨어의 특성과 관련이 있는 관련영역과 임베디드 소프트웨어의 특성과는 연관은 없으나 테스트 프로세스 평가시 반영해야 하는 일반영역으로 분류할 수 있다.

핵심 영역은 임베디드 소프트웨어 테스트 프로세스의 특성과 임베디드 소프트웨어의 특성에서 도출하는 심사항목이다. 임베디드 소프트웨어 테스트 프로세스의 특성은 관련연구에서 언급하였듯이 하드웨어와 소프트웨어의 통합테스트이고 임베디드 소프트웨어의 특성은 실시간 처리성, 고신뢰성, 하드웨어 통합성, 목적 한정성이다. 이러한 각각의 특성은 임베디드 소프트웨어가 만족해야 하는 품질 특성이 된다. 따라서 Emb-TMM 핵심 영역은 품질 특성과 이를 만족하기 위한 품질 목표에 대한 평가로 이루어진다.

#### 3.2.2 수준별 영역 분류

일반 영역과 관련 영역으로 분류된 심사항목을 참조모델의 심사형태에 따라 영역별 수준으로 변환해야 한다. 참조모델을 단계별 형태로 선정할 경우는 지속형에서 도출한 심사항목을 단계별 형태에 맞게 변환해야 하며 참조모델이 지속형인 경우에는 지속형에 맞게 변환해야 한다.

참조모델의 영역별 수준으로 심사항목을 변환 시 지속형 모델의 경우 분석된 영역의 세부 수준이 상위 수준을 만족하는 경우 하위 수준은 자동적으로 만족하는 특성이 있다. 단계형 모델에서 도출한 심사항목을 지속형에 맞게 변환시에도 이

러한 특성에 맞게 변환시켜야 한다.

영역별 수준 변환이 완료되면 Emb-TMM 수준을 참조모델에 근거하여 수준별로 정의한다. 단계형 모델인 TMM, TCMM 또는 CMM은 5단계이며 이를 근거로 단계별 수준을 정의하면 초기 단계, 정의 단계, 통합 단계, 관리 단계, 최적화 단계로 정의할 수 있다.

초기 단계는 테스트 프로세스가 정의되어 있지 않은 단계로 MIS 중심의 테스트 활동을 수행하는 단계이며, 정의 단계는 임베디드 소프트웨어 특성에 대한 정의가 수립되고 MIS 영역 중 특성과 밀접한 관련 영역 중심으로 수행되는 단계이다. 통합 단계는 임베디드 소프트웨어에 대한 특성과 테스트 목적이 반영되고 하드웨어와 소프트웨어 통합 중심의 활동이 수행되는 단계이며, 관리 단계는 품질 측정을 위한 테스트 활동이 수행되고 각 특성에 대한 측정과 평가의 객관성이 확보되는 단계이다. 마지막으로 최적화 단계는 임베디드 소프트웨어 특성에 대한 테스트 프로세스 개선 활동이 끊임없이 관리되고 제어되는 단계로 정의할 수 있다.

참조모델의 수준 정의가 완료되면 일반 영역, 관련 영역, 핵심 영역의 각 심사항목을 수준별 영역으로 분류한다. 일반 영역은 임베디드 소프트웨어의 특성이 반영되지 않은 일반적인 소프트웨어 테스트 프로세스 평가시 적용되는 영역으로 초기 수준에 해당한다. 관련 영역은 임베디드 소프트웨어 특성과 밀접한 관련이 있는 영역으로 임베디드 소프트웨어의 특성은 반영되지 않았으나 수준별 정의에 따라 2단계인 정의 수준으로 분류될 수 있다. 핵심 영역은 임베디드 소프트웨어 특성과 하드웨어와 소프트웨어 통합 테스트 및 품질 특성이 반영된 영역으로 심사항목의 내용과 특성에 따라 통합 수준부터 최적화 수준까지 분류할 수 있다.

핵심 영역 중 하드웨어 통합성 테스트, 목적 한정성에 따른 테스트 목표 설정, 하드웨어와 소프트웨어 테스트 제어, 하드웨어와 소프트웨어 테

스트 단계 명시 등은 통합 수준으로 분류할 수 있다. 임베디드 소프트웨어 특성에 대한 품질 특성을 측정하는 실시간 처리성 평가, 고 신뢰성 평가는 관리 수준으로 분류할 수 있으며, 임베디드 소프트웨어 테스트 프로세스가 지속적으로 개선될 수 있도록 품질에 대한 목표 설정과 결과에 따른 차이를 개선하기 위한 실시간 처리성 제어, 고 신뢰성 제어, 임베디드 소프트웨어 테스트 프로세스 최적화 등은 최적화 수준으로 분류할 수 있다.

### 3.2.3 모델 설계

테스트 성숙도 모델의 각 수준은 특정 테스트 성숙도를 내포하고 있으며 성숙도 목표를 지원하는 영역과 해당 영역을 수행하는 세부요소를 포함하고 있다. 테스트 성숙도 모델 구조는 성숙도 수준에서부터 수준을 구성하는 관점으로 구성되어 있으며 많은 성숙도 모델이 CMM 구조에 그 기반을 두고 있다.

Emb-TMM 구조를 설계할 때 고려해야 될 사항은 임베디드 소프트웨어 특성과 테스트 프로세스의 특성을 반영해야 한다는 것이다. 또한 임베디드 소프트웨어 테스트 프로세스의 지속적인 개선을 보장하기 위해 관련 데이터를 유지하고 관리하는 요소도 또한 반영해야 한다. Emb-TMM 구조의 구성요소에는 성숙도 수준, 테스트 능력, 성숙도 목표, 영역, 일반 영역, 핵심 영역, 프로세스 개선 데이터 영역, 구현 및 조직 적용, 주요 관점 등이 있다.

테스트 성숙도 모델로 테스트 조직을 평가하는 방법에는 모델마다 차이가 있다. 테스트 성숙도 모델로 조직을 평가하는 방법을 심사방법이라 한다. 심사방법을 크게 분류하면 체크리스트와 CMM 심사에 근거한 설문서와 인터뷰 형식의 방법이 있다. 테스트 성숙도 모델의 평가에 대한 해당 영역의 구체적인 수행 내용을 평가하는 데는 체크리스트가 장점을 가지고 있다. 특히 체크리스트는 각 영역에 대한 세부항목을 식별하거나 평가

하는 것이 가능하여 많은 모델이 체크리스트 중심의 평가방법을 참조하고 있다.

체크리스트는 각 수준에 포함된 영역 요소를 세부적으로 분류하고 평가 기준에 따라 그 결과를 산출한다. 체크리스트 작성에는 SEI에서 제공하는 성숙도 설문서(MQ)와 ‘테스팅 임베디드 소프트웨어’에서 제시한 체크리스트를 참조한다[14]. 성숙도 설문서는 SEI에서 제공하는 설문서 템플릿으로 핵심 영역별로 그룹화되어 있으며 각 핵심 영역에 대한 질문으로 구성되어 있다[15].

‘테스팅 임베디드 소프트웨어’에서는 전체 체크리스트를 5가지 항목으로 세분화하고 각 항목별 세부요소에 대한 체크리스트를 제공한다. 따라서 Emb-TMM의 체크리스트는 성숙도 설문서와 제시된 체크리스트를 참조하여 각 영역에 대해 영역별 수준 평가요인을 선정하고 각 평가 요인에 대해 세부평가 요인별로 평가하며 구성요소는 성숙도 수준, 영역, 수준평가 요인, 세부평가 요인으로 구성한다. 체크리스트의 기본 표기는 구성요소의 상관관계를 고려하여 <표 1>과 같이 나타낸다.

<표 1> Emb-TMM 체크리스트 표기

표 기	내 용
Level N	성숙도 수준 N
N.1	해당 성숙도 수준 N의 첫 번째 영역
N.1.a	첫 번째 영역의 첫 번째 수준평가 요인
N.1.a.1	첫 번째 수준평가 요인의 첫 번째 세부 평가 요인
O / X	수준평가 결과 표기

## 4. Emb-TMM 설계절차 적용(사례연구)

### 4.1 참조모델 선정 및 영역 도출

임베디드 소프트웨어에 대한 테스트 프로세스 성숙도 모델 설계를 위해 TMM, TPI, TCMM을 참조모델의 대상으로 선정하여 분석한다.

TMM은 단계형 모델로 5단계의 수준, 13개의 영역으로 구성되어 있으며 심사형태는 설문서와 인터뷰 방법을 채택하고 있다. TPI는 지속형 모델

로 20개의 영역을 가지고 있으며 각 수준은 서로 다른 수준을 가질 수 있다. 단계는 A에서 D까지이며 14개의 수준으로 구성되어 있고 심사형태는 체크포인트 방식을 취한다. TCMM은 단계형 모델로 5단계의 수준, 17개의 영역으로 구성되어 심사형태는 설문서와 인터뷰 방법을 채택하고 있다.

이들 대상 참조모델을 분석한 결과 TMM이 기존의 모델 중 가장 일관성이 있고 성숙한 모델구조를 갖고 있으며 심사 모델, 절차 심사 도구, 질문서, 팀 교육 등에 관한 기준을 잘 제시하고 있는 것으로 판단되어 참조모델로 선정한다.

Emb-TMM 영역은 참조모델인 TMM 영역을 중심으로 TPI, TCMM의 영역을 TEmb의 일반요소로 분류하여 LITO로 세분화하여 18개의 항목을 도출하였다. 이 중 12가지 항목은 TMM과 TCMM의 영역에서 참조하였으며 ‘테스트 환경’, ‘테스트 도구’, ‘의사소통’, ‘보고’, ‘테스트웨어 관리’등 5가지 항목은 TPI에서 도출하였다. 이상의 18개의 항목을 임베디드 소프트웨어 및 테스트 프로세스의 특징과 관련 있는 관련 영역과 관련이 없는 일반항목을 일반 영역으로 분류한다. <표 2>는 참조모델 및 대상 참조모델에서 도출한 관련 영역 및 일반 영역이다.

<표 2> 관련 영역 및 일반 영역 도출 결과

구분	관련 영역(9개)	일반 영역(9개)
영역	<ul style="list-style-type: none"> <li>• 테스트 계획 프로세스 수행</li> <li>• 테스트 환경*</li> <li>• 테스트 도구*</li> <li>• 조직적 검토 프로그램 수립</li> <li>• 소프트웨어 품질평가</li> <li>• 결함예방</li> <li>• 품질제어</li> <li>• 의사소통*</li> <li>• 테스트 프로세스 최적화</li> </ul>	<ul style="list-style-type: none"> <li>• 테스트와 소프트웨어 생명주기 통합</li> <li>• 테스트 조직 설립</li> <li>• 기존 테스트기술 및 방법 규정</li> <li>• 테스트 측정 프로그램 수립</li> <li>• 테스트와 디버깅 목표</li> <li>• 교육훈련</li> <li>• 보고*</li> <li>• 테스트 프로세스 제어</li> <li>• 테스트웨어 관리*</li> </ul>

\* TPI 모델에서의 추가 식별 항목

핵심 영역은 임베디드 소프트웨어 테스트 프로세스의 특성과 임베디드 소프트웨어의 특성 그 자

체로부터 도출한다. 임베디드 소프트웨어 테스트 프로세스 특성에는 하드웨어와 소프트웨어의 통합 테스트가 있으며, 임베디드 소프트웨어의 특성에는 실시간 처리성, 고 신뢰성, 하드웨어 통합성, 목적 한정성 등이 있다. 이러한 특성으로부터 도출한 핵심 영역은 총 8개이며 관련 영역에 포함된 ‘테스트 프로세스 최적화’ 영역은 성숙도 모델의 최상위 수준에 해당하며 테스트 프로세스의 지속적인 개선을 의미한다. 따라서 ‘테스트 프로세스 최적화’는 관련 영역이 아닌 핵심 영역에 포함시키며 항목의 명칭 또한 ‘임베디드 소프트웨어 테스트 프로세스 최적화’ 영역으로 변경한다. <표 3>은 이와 같이 하여 도출된 핵심 영역을 나타낸다. 결과적으로 Emb-TMM 영역은 핵심 영역 9개, 관련 영역 8개 그리고 일반 영역 9개로 총 26개의 영역을 도출하였다.

<표 3> 임베디드 SW 각 특성에서 도출한 핵심영역

구분	특성	핵심영역
임베디드 SW 특성	실시간 처리성	<ul style="list-style-type: none"> <li>실시간 처리성 평가</li> <li>실시간 처리성 제어</li> </ul>
	고 신뢰성	<ul style="list-style-type: none"> <li>고 신뢰성 평가</li> <li>고 신뢰성 제어</li> </ul>
	하드웨어 통합성	<ul style="list-style-type: none"> <li>하드웨어 통합성 테스트</li> </ul>
	목적 한정성	<ul style="list-style-type: none"> <li>목적 한정성에 따른 테스트 목표 설정</li> </ul>
임베디드 SW 테스트 프로세스 특성	HW/SW 통합 테스트	<ul style="list-style-type: none"> <li>HW/SW 통합 테스트 단계 명시</li> <li>HW/SW 통합 테스트 제어</li> <li>임베디드 소프트웨어 테스트 프로세스 최적화</li> </ul>

## 4.2 수준별 영역 분류

Emb-TMM 설계의 대상 참조모델인 TMM과 TCMM에서 도출한 영역 항목은 단계형으로 Emb-TMM과 일치하지만 TPI에서 선정한 영역 항목은 지속형으로 TMM과 동일한 단계형 모델로 변환시켜야 한다. 이를 위해 능력 성숙도 모델인 CMM을 이용할 수 있다. TPI 모델은 테스트 조직의 테스트 프로세스 개선을 단계별로 평가하

는 가이드라인을 제공하며 테스트 프로세스 향상의 궁극적 목적은 CMM 3단계에 도달하는 것이다[3]. 따라서 TPI 모델에서의 최적화 수준을 달성한다는 것은 CMM 3단계를 만족한다는 의미이다. 이러한 관계를 적용하여 TPI에서 추가적으로 도출한 영역의 수준별 변환은 CMM을 이용하여 단계형으로 변환이 가능하며, 단계형으로 변환된 영역 요소는 참조모델인 TMM의 단계형 영역으로 전환이 가능하다. <표 4>는 TMM과 CMM의 수준별 상관관계를 나타낸다.

<표 4> TMM과 CMM 수준별 상관관계

수준	TMM	CMM
Level 5	최적화	최적화
Level 4	관리	관리
Level 3	통합	정의
Level 2	정의	반복
Level 1	초기	초기

지속형 모델에서 추가 식별된 영역요소(<표 2> 참조)를 CMM을 이용하여 단계형 모델인 TMM으로 변환하기 위해 각 영역별 세부 수준을 분석한다. 영역도출 단계에서 추가적으로 식별한 일반 영역의 ‘보고’와 ‘테스트웨어 관리’는 최적화 수준에서 CMM의 Level 3에 해당하며 제어수준 및 효율적 수준은 Level 2에 해당한다. 또한 관련 영역의 ‘테스트 도구’는 제어 수준 및 효율적으로만 구성되어 Level 2에 해당하며, ‘테스트 환경’과 ‘의사소통’은 최적화 수준이 Level 3, 제어 수준 및 효율적 수준은 Level 2에 해당한다.

이와 같이 매핑하여 지속형 모델 요소를 CMM을 이용하여 단계형으로 변환할 수 있으며 <표 4>를 이용하여 다시 TMM의 영역별 수준으로 변환이 가능하다. <표 5>는 지속형 모델의 영역을 CMM 모델을 이용하여 TMM의 영역별 수준으로 변환한 결과이다. 이와 같은 변환 작업을 수행하게 되면 지속형 모델인 TPI에서 추가 도출된 영역



이 TMM에 적용 가능한 단계형 모델의 영역으로 전환된다.

<표 5> 영역 수준 변환

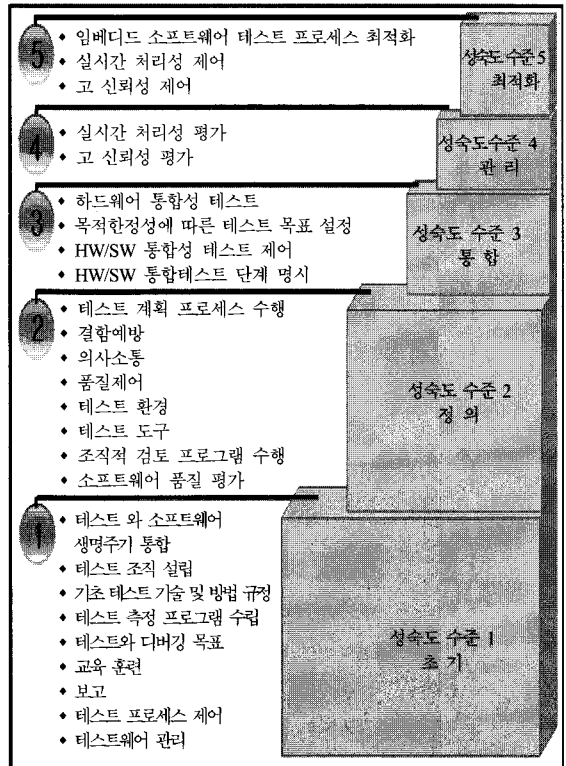
지속형 영역	해당 수준	CMM 수준	TMM 수준	
테스트 환경	Level A, B 테스트환경관리	제어 수준 효율적 수준	Level 2	Level 2
	Level C 테스트환경최적화	최적화 수준	Level 3	Level 3
테스트 도구	Level A, B, C 테스트 도구 관리	제어 수준 효율적 수준	Level 2	Level 2
테스트 웨어 관리	Level A, B, C 테스트 도구 관리	제어 수준 효율적 수준	Level 2	Level 2
	Level D 테스트웨어관리 최적화	최적화 수준	Level 3	Level 3
의사 소통	Level A, B 내부 의사소통	제어 수준 효율적 수준	Level 2	Level 2
	Level C 프로젝트조직 의사소통	최적화 수준	Level 3	Level 3
보 고	Level A, B, C 결함 및 위험보고	제어 수준 효율적 수준	Level 2	Level 2
	Level D 프로세스개선보고	최적화 수준	Level 3	Level 3

다음은 Emb-TMM의 수준을 참조모델인 TMM에 근거하여 수준별로 정의한다. Emb-TMM은 TMM과 같이 5개의 수준을 가진 단계형 모델로 정의되며, 각 수준의 정의를 TMM과 같이 초기 단계, 정의 단계, 통합 단계, 관리 단계, 최적화 단계로 정의할 수 있고 그 내용은 <표 6>에 나타나 있다.

<표 6> Emb-TMM 수준 정의

구분	정 의
Level 5 최적화	• 임베디드 소프트웨어 특성에 대한 테스트 프로세스 개선활동이 끊임없이 관리되고 제어되는 단계
Level 4 관리	• 임베디드 소프트웨어 품질측정을 위한 테스트 활동이 수행되고 각 특성에 대한 측정과 평가의 객관성이 확보되는 단계
Level 3 통합	• 임베디드 소프트웨어에 대한 특성과 테스트 목적이 반영되고 하드웨어와 소프트웨어 통합 중심의 활동이 수행되는 단계
Level 2 정의	• 임베디드 소프트웨어 특성에 대한 정의가 수립되고 MIS 영역 중 특성과 밀접한 관련 영역 중심으로 수행되는 단계
Level 1 초기	• 임베디드 소프트웨어 테스트 프로세스가 정의되어 있지 않은 단계로 MIS 중심의 테스트 활동을 수행하는 단계

Emb-TMM 영역을 <표 6>과 같이 각 수준별 정의에 따라 분류하면 크게 3가지 영역으로 나눌 수 있다. 일반 영역은 임베디드 소프트웨어의 특성이 반영되지 않은 기본적 영역으로 초기 수준으로 분류가 가능하다. 관련 영역은 임베디드 소프트웨어 특성과 밀접한 관련이 있는 영역으로 정의 영역으로 분류할 수 있다. 마지막으로 핵심 영역은 임베디드 소프트웨어 특성이 반영된 영역으로 통합 수준에서 최적화 수준까지 분류한다. 통합 수준은 테스트에 해당하는 항목 중 통합과 관련된 영역으로 분류하며, 최적화 수준은 테스트 프로세스의 최적화 및 제어 수준에 해당하는 항목으로 분류하고, 그 외의 항목은 관리 수준으로 분류한다. 이러한 절차를 따라 Emb-TMM 영역을 각 수준별로 분류한 결과가 <그림 3>에 나타나 있다.

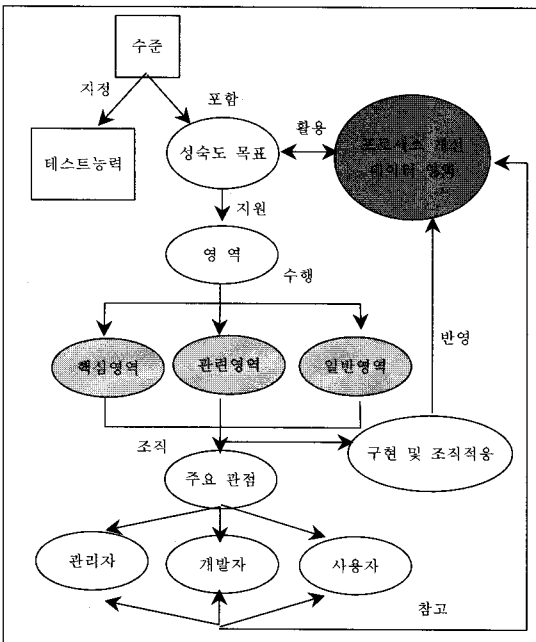


<그림 3> Emb-TMM 영역 분류

### 4.3 모델 설계

Emb-TMM 구조 설계시 고려해야 될 사항은 임베디드 소프트웨어 특성과 테스트 프로세스의 특성을 반영해야 한다는 것이다. 또한 테스트 프로세스의 지속적인 개선을 보장하기 위해 관련 데이터를 유지하고 이를 관리하는 요소를 반영해야 한다. 구성요소는 성숙도 수준, 테스트 능력, 성숙도 목표, 영역, 일반 영역, 핵심 영역, 프로세스 개선 데이터 영역, 구현 및 조직 적용, 주요 관점 등으로 구성한다.

구조 내 구성요소의 관계는 성숙도 수준을 달성하기 위한 상호 연관관계를 지니고 있다. 성숙도 수준은 테스트의 능력을 지정하고 성숙도 목표는 수준에 포함된다. 영역은 각각 핵심 영역, 관련 영역, 일반 영역, 일반 영역으로 수행되고 개발자, 관리자, 사용자의 관점에 의해 조직된다. 수행 영역은 해당 조직에 구현되어 적용되고 그 결과는 프로세스 개선 데이터 영역에 반영된다. 반영된 데이터 영역은 차후 성숙도 목표에 활용된다. <그림 4>는 Emb-TMM 구조의 예이다.



<그림 4> Emb-TMM 구조

마지막으로 Emb-TMM의 체크리스트를 작성한다. 본 논문에서는 테스트 성숙도 수준 2단계 영역인 테스트 계획 프로세스에 대해 간단한 체크리스트 작성 사례를 제시한다.

해당 영역, 즉 '2.1 테스트 계획 프로세스 수행'은 4개의 수준 평가 요인(2.1.a - 2.1.d)을 포함하고 있으며, 각 수준 평가 요인은 1개 또는 2개의 세부 평가 요인(예: 2.1.a.1 - 2.1.a.2)을 포함하고 있다. <표 7>는 수준 2단계 영역인 테스트 계획 프로세스에 대한 체크리스트 작성 예를 보여주고 있다.

<표 7> 체크리스트 작성 사례

Level 2		평가
2.1 테스트 계획 프로세스 수행		
2.1.a	템플릿 사용 테스트 계획 수립	
2.1.a.1	테스트 계획 수립을 위한 표준 절차를 정의한 문서가 있다	
2.1.a.2	테스트 레벨별 테스트를 계획하는 문서가 있다	
2.1.b	사용자 요구사항을 테스트 계획에 입력하는 절차 수립	
2.1.b.1	사용자 요구사항을 반영한 테스트 범위와 기준에 대해 기술되어 있다	
2.1.b.2	프로젝트와 관련된 관리자의 참여 절차가 있다	
2.1.c	임베디드 소프트웨어 테스트 계획 템플릿 개발	
2.1.c.1	임베디드 소프트웨어 테스트를 위한 표준 절차가 있다	
2.1.d	임베디드 소프트웨어 특성에 따른 관리자 요구사항을 테스트 계획에 포함	
2.1.d.1	개발 임베디드 소프트웨어 특성분석을 위한 절차가 있다	
2.1.d.2	개발 제품의 특성을 반영하기 위한 관리자의 참여절차가 테스트 계획에 포함되어 있다	

### 5. 결론

본 논문에서는 임베디드 소프트웨어의 특징과 테스트 프로세스의 특성을 반영한 임베디드 소프트웨어 테스트 성숙도 모델(Emb-TMM) 설계를

위한 절차와 사례연구 결과를 제시하였다.

먼저 Emb-TMM 설계를 위해 참조모델 선정 및 영역 도출, 수준별 영역 분류, 모델 설계의 3단계 설계절차를 제시하였다. 특히 참조모델 선정을 위해 TMM, TPI, TCMM을 분석하였으며 최종적으로 TMM을 본 논문의 참조모델로 선정하였다. 참조모델 선정 후 TEmb 방법의 LITO 영역 적용, 추가항목 식별, 임베디드 소프트웨어 특성 및 테스트 프로세스 특성 등을 통해 8개의 일반 영역, 9개의 관련 영역 그리고 9개의 핵심 영역을 도출하였다. 그런 다음 도출된 각 영역을 성숙도 모델 수준별로 분류하였고 최종적으로 Emb-TMM 설계절차를 제안하였다. 마지막으로 제안한 Emb-TMM 설계절차를 어떻게 적용할 수 있는지를 임베디드 소프트웨어에 적용하였으며 그 결과를 5단계 성숙도 모델을 제시하였다. 이를 통해 임베디드 소프트웨어 개발조직에서는 각 조직의 목표에 맞는 테스트 프로세스 성숙도 평가 모델을 설계하고, 문제점을 인식하여 지속적인 테스트 프로세스의 개선에 기여할 수 있을 것이다.

향후 연구로는 본 논문에서 제시한 Emb-TMM 설계절차를 실제 임베디드 소프트웨어 개발에 종사하는 관련자들의 피드백을 통해 검증 및 보완 작업을 수행하고, 이를 통해 실제 개발 현장에서 적용 가능한 Emb-TMM을 설계해야 한다.

## 참고문헌

- [1] Ilene Burnstein, Taratip Suwanassart and Robert Carlson, "Developing a Testing Maturity Model: Part I," CrossTalk, Journal of Defense Software Engineering, No. 8, pp.21-24, August 1996.
- [2] Ilene Burnstein, Taratip Suwanassart and Robert Carlson, "Developing a Testing Maturity Model: Part II," CrossTalk, Journal of Defense Software Engineering, No. 9, pp.19-26, September 1996.
- [3] Tim Kooman and Martin Pol, Test Process Improvement, Addison-Wesley, ISBN 0-201-59624-5.
- [4] Jari Andersin, "TPI-a model for Test Process Improvement," Seminar on Quality Models for Software Engineering, 5 October 2004.
- [5] Durant, J., "Software Testing Practices Survey Report," Software Practices Research Center, 1993.
- [6] 이희연, "LG전자의 Embedded S/W 활용방안," LG전자기술원 정보기술연구소, 2004.12.
- [7] CTO전략실, "임베디드 시스템의 시험과 시험환경," 삼성전자 SW센터, 2003.6.
- [8] Ivica Crmkovic, "Component-based Software Engineering for Embedded Systems", ICSE, 2005.05
- [9] Martin Toerngren, "Component-based vs. Model based Development A Comparison in the Context of Vehicular Embedded Systems", IEEE, 2005.
- [10] 한국소프트웨어진흥원, 「국내 임베디드 소프트웨어 산업 실태조사에 관한 연구」, 2004.04.
- [11] IEEE Std 1012-1998, IEEE Standard for Software Verification and Validation, IEEE Computer Society, 20 July 1998.
- [12] 한국전자통신연구원, 「제품계열 기반 시스템 개발방법론 마르미-EM」, Ver 1.0, 2006.
- [13] Stefan Van Bealen, Joris Gorinsek and Andrew Wills, The DESS Methodology, ITEA, 2001.12.
- [14] Bart Broekman and Edwin, Testing Embedded Software, Addison-Wesley, 2003.
- [15] David Zubrow, William Hayes, Jane Siegel, Dennis Goldenson, "Special Report-Maturity Questionnaire", Software Engineering Institute, 1994.

---

|| 저자 소개 ||

---

**백 상 훈 (E-mail: navy51th@hanmail.net)**

1997      해군사관학교(문학사)  
2006~현재    국방대학교 전산정보 석사과정  
관심분야    소프트웨어 테스트, 소프트웨어 규모추정

**윤 희 병 (E-mail: hbyoon37@hanmail.net)**

1983      해군사관학교(이학사)  
1986      연세대학교(공학사)  
1991      미국 해군대학원 전산공학(석사)  
1998      미국 조지아공대 전산공학(박사)  
2002~현재    국방대 전산정보학과 부교수  
2004~현재    아주대/서강대 정보통신대학원 외래교수  
관심분야    임베디드 소프트웨어, 소프트웨어 테스트, 소프트웨어 아키텍처