

국방 소프트웨어 시험성숙도모델 개발을 위한 테스트 프로세스 모델 비교분석

송경희* · 이병걸** · 류동국*** · 김진수****

1. 서 론

최근 IT 업계에서는 소프트웨어 품질에 대한 중요성이 부각되고 있다. 특히 최종적인 제품이나 서비스의 품질 개선도 중요하지만 제품을 만들기까지의 과정과 프로세스의 개선도 못지않게 중요하다는 것을 인식하고 조직의 개발 프로세스 성숙도를 평가·개선하기 위한 모델 개발에 많은 노력을 기울이고 있다. 하지만 기존의 성숙도 모델(CMM, CMMI, SPICE 등)들은 소프트웨어 개발 전반에 걸친 프로세스의 개선을 포괄적으로 다루고 있어서 소프트웨어의 품질활동과 밀접한 관련이 있는 테스트 프로세스의 개선에 대해서는 상세하게 다루고 있지 못하다. 예로, CMM이나 SPICE에서는 주요 프로세스 영역(Key Process Area)에 독립적인 테스트 활동을 포함하지 않는다[1]. 따라서 테스트 프로세스 개선을 위해서는 별도의 모델이 필요했으며 TMM (Testing Maturity

Model), TPI (Test Process Improvement) 등의 모델들이 개발·적용되어 왔다.

하지만 이들 테스트 프로세스 모델들은 국방 도메인과 같은 특수한 상황에 그대로 적용하기에는 많은 어려움이 따른다. 국방 도메인의 특성상, 다른 제품에 비해 엄격한 수준의 테스트가 요구되며, 요구문서나 설계서와 같은 중간 산출물에 대한 검토뿐만 아니라, 테스트 도구나 환경요건 등에 대한 요구도 훨씬 복잡하고 상세하다. 국방 소프트웨어의 품질 개선을 이루기 위해서는 국방 정보체계나 무기체계 소프트웨어들의 품질 특성과 품질 요구사항이 충분히 고려되어야 한다. 따라서 국방 도메인에서의 소프트웨어 개발 환경 및 품질 특성에 맞는 테스트 프로세스 모델이 절실히 필요하다.

본 연구에서는 현재 진행 중인 국방 소프트웨어 시험 성숙도 모델인 MND-TMM (Ministry of Defense - Testing Maturity Model)의 개발을 위해, 국내외의 주요 테스트 프로세스 개선 모델을 비교한다. 기존 모델들의 특징점을 적극 활용하여 테스트 업무 현장의 best practice들을 조사하며 특히 국방 분야에서의 소프트웨어의 품질 특성과 테스트 요구사항을 분석하여 우리나라 국방 소프트웨어 개발 환경에 적합한 테스트 프로세스 모델을 개발하는 데 활용하고자 한다.

※ 교신저자(Corresponding Author) : 송경희, 주소 : 서울특별시 노원구 공릉2동 126번지(139-774), 전화 : 02)970-7745, FAX : 02)970-5981, E-mail : rodem@swu.ac.kr

* 서울여자대학교 대학원 석사과정

** 서울여자대학교 컴퓨터학과 교수
(E-mail : byong1@swu.ac.kr)

*** 국방과학연구소(Agency for Defense Development)
(E-mail : dkryu@lycos.co.kr)

**** 국방과학연구소(Agency for Defense Development)
(E-mail : jskim421@add.re.kr)

※ 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.

2. 국방 도메인 관련 테스트 현황과 개선점

소프트웨어의 품질 개선을 위해서는 고객 요구 사항 및 품질 특성의 정확한 이해와 함께 효율적이고 효과적인 품질 측정과 평가 활동이 필요하다. 이를 위해 최근 산업계에서는 소프트웨어의 품질 특성 정의 및 이들의 정량적 평가 방법을 기술하고 있는 국제표준 ISO 9126 및 ISO 14598를 도입하여 테스트 활동에 사용하고 있다. ISO 9126은 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성이라는 여섯 가지의 주 품질 특성과 완전성, 추적성, 자기기술성 등 40개의 부 특성을 항목을 정의하고 있다[2]. 국내에서도 이들 표준을 기반으로 소프트웨어 제품에 대한 품질 요구사항을 정의하고, 개발된 최종 소프트웨어의 품질을 측정, 평가하는 등 품질활동이 활발하게 일어나고 있다[3].

그러나 이러한 품질 개선 모델을 국방 소프트웨어 분야에 그대로 적용하기는 매우 어렵다. 국방 소프트웨어는 품질 요건을 갖추지 못할 때 초래하는 결과가 중대하므로 매우 높은 신뢰성을 요구한다. 또한 소프트웨어와 특정 하드웨어가 결합된 내장형(Embedded) 시스템이 많으므로 통합적인 테스트가 필요하다. 특히 안정적인 사무실 환경이 아닌 야전 상황까지 고려한 테스트가 필요하므로 치밀한 전략과 실제적인 테스트 환경 등이 뒷받침되어야 한다.

이렇듯 테스트 요건은 까다롭고 복잡한 반면 무기체계의 종류가 매우 다양하고 인터페이스가 적은 내장형 시스템들이 많으므로 테스트 과정은 더욱 어렵다고 볼 수 있다. 그러므로 정확한 품질 수준이 정의되어야 하며, 각 수준마다 적절하고 실제적인 테스트 프로세스가 확립되어 있어야 한다.

2.1 국방 도메인의 분류

국방 도메인에서의 소프트웨어 품질 특성을 파악하기 위해서는 먼저 체계적인 국방 도메인의 분류가 필요하다. 현재 국내외적으로 공통적으로 수용되는 국방 소프트웨어의 분류는 다음과 같다 [4].

- 지휘통제·통신 무기체계
- 감시·정찰 무기체계
- 기동, 함정, 항공, 화력, 방호, 기타 무기체계
- 비무기체계, 무기체계 외의 국방정보체계

무기체계라 함은 유도무기, 항공기, 함정 등 전장에서 전투력을 발휘하기 위한 무기와 이를 운영하는 데 필요한 장비, 부품, 시설, 소프트웨어 등 제반요소를 통합한 것을 말한다. 무기체계 외의 소프트웨어 등 제반요소는 비무기체제로 분류된다.

이 중에서도 ‘지휘통제·통신 무기체계’ 및 비무기체계 내의 소프트웨어는 대규모 집약형 체계로 구성되어 있어서 매우 엄격하고 높은 수준의 품질을 요구한다. 따라서 시험 평가에 소요되는 비용이 전체 개발 기간의 절반 이상을 차지하는 경우가 대부분이다. 국방 도메인에서의 소프트웨어의 특징은 다음과 같다[4].

- Mission-critical 한 체계이다.
- 단독 체계(standalone system)만으로 구성되지 않고, 다수의 체계들이 전체 체계를 구성하는 체계들의 체계이다.
- 소프트웨어 집약형 체계(software intensive system)이다.
- 하드웨어와 소프트웨어의 복합 체계로서 소프트웨어에 대한 시험뿐만 아니라, 하드웨어를 포함하는 전체 체계에 대한 시험이 필요하다.

- 체계들 사이의 데이터 호환이 매우 중요하며 다양한 체계들 간의 상호운용을 보장하기 위해 표준에 의한 구현이 중요하고, 시험 과정에서 상호운용성이 검증되어야 한다.

2.2 국방 도메인 관련 표준들

본 절에서는 국방 도메인과 관련하여 해외의 테스트 프로세스 관련 규격 사례들을 기술한다. 미 국방성에서는 1988년 DOD-STD-2167A (Defense System Software Development)를 시작으로 하여 군사관련 시스템 규격들이 만들어졌다. 이 규격들은 통상 MIL-STD, 또는 MIL-SPEC 이라고 지칭되며, 소프트웨어 개발과 관련해서는 MIL-STD-498이 제정되어 있다. MIL-STD-498이 군용이라면 이를 일반 기업 버전으로 변경(de-militarize)한 것이 J-STD-016이다[5]. MIL-STD-498은 소프트웨어 라이프사이클 프로세스를 정의하는 IS 12207에 의해 대체되었지만, 군 환경에서의 소프트웨어 개발과 테스트 절차를 정의한 최초의 표준이었다[6]. 전체 개발 주기 중 테스트와 관련한 프로세스는 다음과 같다 [7].

- Software implementation and unit testing
- Unit integration and testing
- CSCI (computer software configuration item) qualification testing
- CSCI/HWCI (hardware configuration item) integration and testing
- System qualification testing

MIL-STD-498과 J-STD-016은 ISO 12207에서 정의하는 acquirers, suppliers, developers, operators, maintainers 중 주로 개발자 관점에서만 기술되었다[8]. 그러므로 특정 조직의 테스트

성숙도 측정이나 개선 지침, 개발 과정상의 테스트뿐만 아니라 운영 테스트 등의 영역에 대해서는 보완이 필요하다. 이 외에도 SOSI (System of Systems Interoperability), LISI (Levels of Information Systems Interoperability)는 이기종 시스템간의 상호운용성을 평가하기 위한 지침을 제공하고 있다.

한편 국방 도메인 관련 국내 표준을 살펴보면 방위사업청에서 개발한 ‘소프트웨어 개발 프로세스’등이 있다. 이 표준은 국방 소프트웨어 개발을 위한 계획, 각 체계에 따른 요구분석, 구조 설계, 개발, 시험평가, 통합 및 인수 등에 대한 지침을 제공하고 있다[9]. ADDMe (Advanced Defense component Development Methodology)는 2003년도에 개발된 국방 CBD 방법론으로서 분석, 설계, 구현 및 테스트, 인도 단계를 포함하는 개발자 관점의 방법론이다. UML 기반이며 테스트 위주의 개발을 지원하여 불필요한 코드를 미연에 방지하고 소프트웨어의 품질을 높이고 도모하고 있다. 2004년부터 국방부 및 각 군 기관에서 적용되고 있다. 그러나 ADDMe는 비무기체계 분야의 컴포넌트 개발 방법론으로서 테스트 관련 활동이 미약하며, 특히 소프트웨어와 하드웨어 모두를 통합한 제품의 테스트에는 많은 한계점이 있다[10]. 또한 최근 국방 무기체계 분야에서 임베디드 소프트웨어 개발 방법론을 도입하고 SOA (Service Oriented Architecture) 기술 도입이 가속화 되는 시점에서 다양한 개발 방법론에 적용할 수 있는 국방 소프트웨어 시험평가 모델이 필요하다[11].

2.3 국방 도메인에서의 시험 평가

국방 소프트웨어 시험 평가란 ‘특정 무기체계가 기술적 측면 또는 운용관리적 측면에서 명시된 제반 요구조건을 충족하는지 확인/검증하는 절

차'를 말한다. 시험 평가는 크게 (a) 요구 성능에 대한 기술적 도달 정도에 중점을 두는 개발시험평가 (Development Test & Evaluation, DT&E)와 (b) 요구 성능 및 운용상의 적합성과 연동성에 중점을 두는 운용시험평가 (Operational Test & Evaluation, OT&E)로 분류할 수 있다[11].

기존의 시험 평가 사례를 보면 개발팀이 주관하여 통합 모듈시험을 미흡하게 한다거나 분산 네트워크 등 운용환경이나 동등 조건에서 실시되지 못하는 한계가 있었다. 외부 기관에 의뢰하여 시험한 경우 역시 단위 모듈 시험, 통합 시험 등이 미흡한 경우가 있었다. 특히 가장 큰 문제점은 운용시험평가에서도 실제 작전환경이 아닌 축소된 환경에서 실시하기 때문에 신뢰성이 떨어진다는 점이었다. 많은 부분 시뮬레이션 테스트로 대체하고 있지만 한계가 있기 때문에 실제 운용 환경과 유사한 테스트 환경을 갖추는 것이 매우 어려우면서도 매우 중대한 요건이다.

국방 소프트웨어의 mission critical 한 특성을 고려해볼 때 현재보다 훨씬 정형화된 시험평가 프로세스가 정립되고 준수되어야 하며, 전문성을 갖춘 인력과 테스트 환경을 위한 예산이 투입되어야 할 것이다. 국방 소프트웨어 시험 평가에 있어서 문제점들을 정리해 보면 다음과 같다[10].

- 독립적인 시험평가 조직의 부재
- 시험평가 관련 인원의 능력 부족
- 시험평가 전략의 부재
- 시험 프로세스의 미성숙
- 시험 기법 적용의 문제
- 운용시험평가 환경 구축이 미흡
- 시험 데이터 구축이 어려움
- 시험 인프라 미흡
- 시험평가를 거친 소프트웨어 또는 정보체계의 품질이 미흡함

앞에 기술한 문제들을 해결하기 위해서는 다양한 개발방법론에 적용할 수 있는 시험평가 프로세스가 정립되어, 현실적으로 적용 가능한 무기체계별 시험 전략 선택, 세부 시험 기법 적용 등의 방법과 지침이 제시되어야 한다. 또한 테스트를 위해 독립된 조직을 갖추고 인력과 예산이 투입되어야 하며 신뢰성을 높일 수 있도록 적절한 시험 환경이 뒷받침 되어야 한다. 이를 위해 테스트 성숙도 모델에서는 각 수준 별로 테스트웨어, 자동화 도구, 시험환경 등 다각적 측면에서 구체적인 테스트 인프라의 요건을 제시해야 한다.

3. 소프트웨어 프로세스 개선 모델과의 연계

해외의 경우 1990년 중반부터 기존의 테스트 기법, 프로세스, 사례 등을 통합하여 테스트 프로세스 성숙도 모델을 개발하고 이를 적용하고 있다. 시험 프로세스가 성숙될수록 시험 전략, 시험 정책, 시험 기법, 시험 환경 등이 개선되고 이 개선 사항들은 바로 시험 수준의 향상으로 이어질 수 있다. 이러한 움직임은 기존 SW-CMM, CMMI, SPICE 모델 등, 프로세스 성숙도 모델들의 테스트 관련 프로세스 측면에서 미비한 점을 보완하여 상호 연계하는 방향과, 이와는 독립적으로 테스트 프로세스만의 전문화된 프레임워크를 구축하는 연구 방향으로 나뉜다. 미국에서 산학연구로 개발되어 군에서 적용하고 있는 TMM의 경우가 전자이며, 유럽에서 개발된 TPI 모델의 경우는 후자에 해당한다.

국내의 경우 기존 프로세스 성숙 모델의 부족한 점을 보완하는 차원에서 체계적인 테스트 프로세스 프레임워크를 개발하고자 하는 연구들이 진행 중이다. 국내의 방위 산업 관련 업체들을 조사한 결과, 현재 대표적인 소프트웨어 개발 프로세스 개선 모델인 CMMI를 적용하고자 하는 조직이 많

왔고 따라서 테스트에 관한 모델이 보완될 때 CMMI와 논리적, 실무적 연계성을 갖기를 희망하였다. 이는 새로운 모델에 대한 교육 훈련 부담 (learning curve)을 줄이고 용어의 통일과 의사소통, 실무활동에서 일관된 데이터 축적 등을 위해서도 바람직하다.

CMMI는 원래 여러 CMM 모델들 간의 불일치와 중복성을 해소할 뿐 아니라 공식적인 국제 표준인 ISO/ICE15504 (SPICE)와의 호환성을 확보하는 것이 목적이었다. CMM과 비교해볼 때 CMMI의 중요한 특징은 조직 차원의 성숙도 (organizational maturity)를 측정할 수 있는 staged 모델과 함께 각 핵심 영역의 능력 수준 (process area capability)을 측정하고 프로젝트 특성에 맞게 개선하기 용이한 continuous 모델을 제공한다는 것이다[12]. 이처럼 두 가지 측면으로 성숙도 모델을 제시함으로써 조직은 그 특성과 목적에 맞게 모델을 활용할 수 있게 되었다. 그러므로 MND-TMM 개발 시 CMMI 구조의 장점을 살려 조직 전체의 테스트 성숙도를 정의하는 모델과 특정 영역별 성숙도를 정의하는 모델로 나누어 각 조직과 프로젝트의 특성에 따라 유연하게 적용할 수 있도록 한다.

4. 테스트 프로세스 모델들의 특징점 비교

CMMI나 SPICE 등은 소프트웨어 개발 활동 전체의 프로세스 성숙도를 평가하는 모델이다. 그러나 이들 모델은 테스트와 관련된 이슈들을 상세히 다루지 못하고 있다. 이들 모델과 연계되면서도 테스트를 위한 프로세스 요건과 가이드라인, 평가 및 개선 방법을 다루는 모델이 필요하다. 또한 군체계와 국방 소프트웨어의 특성들을 감안하여 국방 도메인을 위한 특화된 테스트 프로세스 모델을

개발해야 할 필요가 있다. 본 절에서는 기존의 테스트 프로세스 개선 모델들의 특성과 장단점을 살펴봄으로써 국방 소프트웨어 시험 평가 프로세스 정립을 위한 요소들을 고려하고자 한다. 테스트 프로세스 성숙도 모델을 활용함으로써 다음과 같은 효과를 얻을 수 있다[4].

- 테스트 활동이 조직화되며, 전문적이고 공학적인 활동으로 인식된다.
- 개발 활동과 독립적인 테스트가 수행된다.
- 성숙도가 높아질수록 테스트를 통해 개발 초기에 오류를 탐지하거나 결함을 방지하기 위한 활동으로 변화한다.
- 테스트 활동이 정량적으로 측정되고 관련 데이터가 축적되어 테스트 프로세스의 문제점을 파악하고 개선, 최적화하게 된다.
- 테스트 성숙도 레벨이 증가하면서 조직의 테스트 비용이 절감되고 최종 제품이나 서비스의 품질이 향상되어 고객만족을 기할 수 있다.

본 연구에서 비교 분석할 주요 테스트 프로세스 모델들은 다음과 같다.

- Testing Maturity Model (TMM)
- Testing Capability Maturity Model (TCMM)
- Test Improvement Model (TIM)
- Test Process Improvement Model (TPI)
- Test Organization Maturity Model (TOM)
- Testing Assessment Program (TAP)
- Maturity Model for Automated Software Testing (MMAST)
- Testing Maturity Model Integration (TMMi)
- Matrix-Based Test Maturity Model (MB-TMM)

4.1 Testing Maturity Model (TMM)

TMM은 일리노이 공대의 Ilene Burnstein 교수팀에 의해 개발된 것으로 SW-CMM에 호환될 수 있도록 설계된 테스트 프로세스 성숙도 모델이다. 즉, CMM에서 테스트 활동에 대한 프로세스를 상세화한 것이다. TMM은 조직 전체의 테스트 프로세스 성숙도 수준을 정의하고 현 상태를 평가하여 개선해 나가기 위한 심사 모델 및 절차, 심사 도구 및 질문서, 팀 교육 등에 관한 기준을 제시한다. CMM과 명칭과 개념구조가 유사하므로 CMM에 대한 배경 지식과 경험이 있는 조직에서는 쉽게 접근할 수 있다.

그러나 TMM은 다음과 같은 문제점을 가진다.

- TMM 적용을 위해서는 경험 있는 프로세스 개선 책임자 및 인력이 필요하게 되는데, 이에 대한 자세한 설명이나 지침이 없다.
- 인력이나 조직구성을 위한 목표나 활동에 대한 설명이 없다. 테스트 조직을 정립하고 훈련 프로그램을 확립하라는 내용은 언급하지만 그 활동들을 어떻게 수행해야 하며, 이를 어떻게 추적/감시할 것인지에 대한 구체적 설명이 없다.
- 테스트 도구, 테스트 시스템, 테스트 환경 등의 테스트 인프라에 대한 지침이 없다.
- 조직의 테스트 프로세스의 문제점이나 현 성숙도를 파악하는 데는 도움이 되지만 프로세스 '개선'에 대한 지침은 부족하다.
- CMM에 기반한 모델이므로 CMM과 같이 적용되지 않으면 큰 효과를 기대할 수 없다 [13].

이상에서 볼 때 TMM 모델을 통해서도 조직 전체의 테스트 프로세스 성숙도를 심사하고 평가하여 국방 소프트웨어 개발 프로젝트를 위해 관련

업체들을 평가하거나 일정 수준 이상의 성숙도를 필요로 할 때 활용할 수 있는 구조를 시사해 준다.

4.2 Testing Capability Maturity Model (TCMM)

TCMM은 1996년에 개발된 모델로 많은 부분에서 CMM과 유사하게 구성되어 있다. TCMM의 핵심영역은 CMM의 핵심영역의 많은 부분을 그대로 도입하고 테스트 관련 활동들이 추가된 형태이다. TCMM은 미 연방 항공 운영 위원회의 ACT-200 프로그램의 테스트 성숙도를 심사하는데 성공적으로 적용된 바 있다[13].

TCMM 모델은 CMM과 유사한 성숙도 수준과 함께 테스트 관련하여 심사를 위한 서면 및 인터뷰 질문서, 체크리스트 지원하고 있다. 또한 평가 데이터를 처리하기 위한 템플릿 도구 지원한다. 그러나 TCMM은 CMM의 내용을 테스트 영역으로 바꾸어 놓은 것에 지나지 않기 때문에 테스트 프로세스만의 독립적 평가가 어렵고 조직이 CMM을 적용하고 있을 때에만 효과를 기대할 수 있다는 단점이 있다.

4.3 Test Improvement Model (TIM)

TIM은 CMM과 TSM에 근간을 두고 있다. 핵심영역과 평가 프로시저로 구성되어 핵심 영역에서 테스트 활동의 현 상태를 파악하고 장점은 살리고 약점은 제거하는 방안을 제시한다. TIM은 주요 핵심 영역에 대해 초기수준, 베이스라인 수준, 비용효과 수준, 위험감소 수준, 최적화 수준이라는 5개의 성숙도 수준을 정의하고 있다.

TIM은 구조에 있어서 CMM의 핵심영역과 유사하지만 (Organization, Planning & tracking, Testware, Test cases, Reviews), CMM의 핵심

영역이 하나의 수준에 한정된 것에 비해 TIM은 0~4의 성숙도 수준에 걸쳐 확장하였다. 또한 TMM이 테스트 프로세스를 조직 차원에서 보고 테스트 조직, 위원회 구성, 정책 및 책임에 대한 내용을 강조하는 반면 TIM은 테스트 조직과 다른 조직 간의 의사소통 측면을 강조한다. 예를 들어 테스트 조직을 만들고 테스트 조직과 개발자 조직 간에 job orientation을 실시하도록 지침을 주고 있다[13].

4.4 Test Process Improvement Model (TPI)

TPI는 Kooman과 Pol에 의해 1997년에 개발된 모델로 테스트 프로세스 개선을 더욱 용이하게 할 수 있도록 고안되었다. TPI는 조직의 테스트 프로세스의 장점과 약점 영역을 결정할 수 있게 하고 프로세스 성숙도 심사, 개선 사항을 제안한다. 중요한 특징이 핵심 영역별 성숙도 수준 매트릭스를 제공하고 구체적인 체크리스트, 개선 제안 사항 등을 제공한다는 점이다. 성숙도 모델 구조는 크게 제어되는 수준, 효율적으로 이루어지는 수준, 최적화 수준 등 3가지 성숙도 수준과 14개의 상세 등급을 갖는다[14].

- 제어수준 : 0~5 등급. 테스트 프로세스가 제어되고 있음을 의미한다. 수준별 특성을 가져야 하고 정의된 테스트 전략과 조화를 이루어야 한다. 테스트 명세, 결점 보고, 의사소통이 일어나고, 테스트웨어 및 테스트 운영이 관리된다.
- 효율수준 : 6~10 등급. 효율적 테스트 프로세스의 정립을 의미한다. 테스트 프로세스가 자동화되고, 통합되며, 개발조직에 정착된 수준이다.

- 최적화 수준 : 11~14 등급. 최적화 상태를 의미한다. 지속적인 개선이 조직의 일상적인 활동으로 인식된다.

TPI는 20가지 핵심영역들마다 등급 별로 점검되어야 할 체크포인트들을 상술하고 있다. TPI에서 정의하는 20개의 핵심영역은 다음과 같다[14].

- 1) 테스트 전략, 2) 생명주기 모델, 3) 참여 시점, 4) 산정과 계획, 5) 테스트 명세, 6) 정적 테스트, 7) 척도, 8) 테스트 도구, 9) 테스트 및 10) 사무실 환경, 11) 약속과 동기, 12) 테스트 기능과 훈련, 13) 방법론의 범위, 14) 의사소통, 15) 보고, 16) 결점 관리, 17) 테스트웨어 관리, 18) 테스트 프로세스 관리, 19) 평가, 20) 하위레벨 테스트

TMM이 조직 전체의 성숙도를 정의하는 반면 TPI에서는 카테고리 별로 핵심영역별 성숙도를 정의하고 있다. 이는 CMMI에서 제공하는 Continuous Representation과 비슷한 측면이라 할 수 있다. 또한 상세한 체크리스트를 제시하여 핵심 영역별로 현재 수준에 대한 진단 및 개선 전략을 세우는 데 구체적인 지침을 제시한다.

TMM에 비해 시험환경, 사무환경 등의 요소가 독립적으로 다루어지고 있다. 이에 더하여 TPI 모델이 정의하는 대로 수준을 진단할 수 있는 TPI Scoring Tool도 제공되고 있다. 이 틀은 스프레드시트의 매크로 기능을 이용하여 각 핵심 영역마다 체크포인트의 요건을 만족시키는지의 여부를 입력하면 해당 영역에 대한 수준을 자동으로 계산해 준다. 그러므로 TPI를 참조하여 국방 소프트웨어 테스트를 위한 핵심 영역을 선별하고 수준마다 요구되는 구체적인 활동 사항들을 체크리스트로 작성하여 실제적인 가이드라인과 수준 진단의 근거로 활용할 수 있을 것이다. TPI는 특히 실제적

이고 유용한 자료들이 많이 공개되어 있어 테스트 모델의 상세 설계 시에 테스트 실무 측면에서 참조하기 용이하다.

4.5 Test Organization Maturity Model (TOM)

TOM은 System Evolution 사에 의해 개발, CMM에서 다루지 못하는 테스트 조직에 대한 이슈들을 강조하여 개발된 모델이다. 다른 모델들이 사후 문제 해결에만 관심을 둘 뿐 문제의 원인, 목적, 제약 사항 등에 대한 요소를 충분히 다루지 못하여 이를 보완하기 위한 방법들을 제시하였다. TOM에서는 프로세스 개선을 실현하는 데에 가장 큰 장애 요소는 관리자의 인식변화 및 구성원의 저항감이라고 보고, 이러한 조직 차원의 문제들을 파악하고 해결책을 제시한다. 이를 위해 문제를 파악하고 우선 순위화하기 위한 질문서와 개선·제안점 등을 포함한다[1].

질문서는 20개의 질문/증상으로 구성되었고, 주로 기술적인 이슈들보다 조직적인 이슈에 초점을 두고 있다. 각각의 질문/증상에 대한 점수가 계산되면 해당 질문과 관련하여 조직이 최우선적으로 해결하고자 하는 증상이 순위화 되며, 상위 증상을 우선적으로 해결하도록 제안하는 방식이다. 모든 점수와 순위는 5개 등급으로 구성되어 누적·합산되며 합산된 점수와 순위는 데이터베이스에 저장되어 개선 모델을 제안하는데 사용된다. 데이터베이스는 83개의 테스트 프로세스의 개선 사항을 유지하며, 입력 심사 데이터를 바탕으로 최대 70개의 개선 제안 사항을 생성하고 제시할 수 있다.

지금까지 약 120개의 조직이 TOM을 사용하여 테스트 프로세스의 개선을 시도했다고 알려져 있다[13]. TOM은 조직의 문제를 해결하기 위한 솔루션 제공을 위한 모델로서 조직의 문제 및 증상

파악에 초점을 둔다. 질문서를 제공하고 이 질문서에 대한 응답과 채점이 완료되면 이를 바탕으로 프로세스 개선안이 도출된다. 따라서 조직의 테스트 프로세스에서 특정 문제를 진단하고 해결할 수 있는 지침을 얻을 수 있다.

4.6 Testing Assessment Program (TAP)

TAP 모델은 CMM에 기반을 두고 테스트 프로세스를 어떻게 개선할 것인가를 돕기 위해 Software Future사와 IE Testing Consultancy 사에 의해 1995년 개발된 모델이다. CMM과 유사한 5단계의 성숙도 레벨과 goals, people, management, techniques이라는 4개의 핵심 영역을 정의하였다[13].

TAP의 특징은 특별한 테크닉이나 툴, 조직의 표준 등을 언급하지 않고 CMM을 비롯하여 다른 모델들에서는 간과된 '사람'에 관련된 이슈들을 다룬다는 것이다. 목적을 이룰 수 있는 적절한 스킬과 능력, 동기를 가진 사람들을 갖추어야 함을 강조하고 있다.

4.7 Maturity Model for Automated Software Testing (MMAST)

MMAST는 의료기기 제작에 필요한 소프트웨어 테스트 자동화를 위해 개발되었으며 다음과 같이 4개의 자동화 수준으로 나뉘어 있다[13].

1. Accidental Automation
2. Beginning Automation
3. Intentional Automation
4. Advanced Automation

MMAST는 개별 수준에서 자동화를 달성하는데 소요되는 비용에 대한 정보, 체크리스트 등을 제공한다. 체크리스트는 개발 프로젝트의 규모,

제품의 복잡도, 조직의 재정적 리스크, 제품이 한자와 의사에게 미치는 위험도 등을 묻고 4등급의 점수 중 최고점으로 자동화 수준을 판단한다[13].

테스트 자동화는 많은 노력과 비용을 절감시키는 방법이지만 국방 소프트웨어의 경우 연관된 하드웨어에 종속성이 강하고 기종이 복잡하여 자동화가 쉽지 않은 문제이다. 특히 잘못된 프로세스를 자동화하거나 세부 차이점을 간과하고 잘못 적용된 자동화는 더 큰 문제를 낳을 수 있으므로 목적과 특성에 맞도록 세심한 주의가 필요하다. MMAST 모델을 이용하면 국방 소프트웨어 테스트를 위해서는 자동화 수준을 판단하는 기준이나 이를 구현하기 위한 예상 비용을 산정하는 방법 등을 참고할 수 있을 것이다.

4.8 Testing Maturity Model Integration (TMMi)

CMMI에서는 성숙도 수준의 3레벨까지는 테스트 활동이 명시되지 않는다. 3레벨에서부터 Engineering 카테고리에서 review 활동과 software quality management, defect prevention 등의 영역이 정의되어 있다[13]. TMMi는 CMMI에서 다루는 전체 개발활동 중 테스트 영역을 독립적으로 평가하기 위한 성숙도 모델이다. TMMi는 CMMI 구조와 유사하게 Generic Goals, Generic Practices를 정의하고 프로세스 영역별로 Specific Goals, Specific Practices를 정의하였다. TPI 등의 모델보다 간단하면서도 공식화되고 제도화될 것을 강조한다. 영국의 TMMi Foundation을 중심으로 활동 중이다[15].

4.9 Metrics-Based Test Maturity Model (MB-TMM)

MB-TMM은 1996년부터 Burnstein에 의해 개

발된 모델로서 TMM을 근간으로 여러 테스트 프로세스 개선 모델들을 비교 분석하여 장점을 살리고 강화하는 취지로 만들어졌다. TMM과 유사하게 성숙도 수준, 프로세스 영역, 목표와 common features, Generic Practices, Activities, Sub-practices 등을 구성요소로 한다.

현재 MB-V2M2 (Metrics Based Verification & Validation Maturity Model) 프로젝트를 진행 중이며 사이트를 통해 MB-TMM의 전반적인 구조 등, 산출물을 공개하며 계속 갱신해 가고 있다.

이상으로 주요 테스트 모델들의 특징을 살펴본다. 다음 표 1은 MND-TMM 개발에 적용할 수 있는 장점들을 위주로 하여 각 모델들을 비교·분석한 것이다.

표 1. 테스트 프로세스 모델들 특징점 비교

모델명	MND-TMM에 적용할 사항
TMM	CMMI 관련구조 (성숙도 수준, staged, 단계적 표현)
TCMM	CMM 기반 구조 질문서, 체크리스트, 템플릿
TIM	다양한 심사 모델, 질문서
TPI	핵심영역별 성숙도 매트릭스 상세한 체크리스트 시험/사무 환경
TOM	조직의 문제 진단 질문서 DB
TAP	사람 관련 이슈와 해결책
MMAST	자동화 수준 정의 비용 예측
TMMi	CMMI 관련구조 (성숙도 수준, staged, continuous)
MB-TMM	매트릭 기반 구조

5. 결론: MND-TMM의 기본 구조

현재까지 MIL-STD-498, ADDMe 등 국방 소프트웨어 개발을 위한 방법론 및 프로세스는 많이 연구되고 적용되었지만 테스트 분야에 대한 상세한 연구와 지침은 아직 부족하다. 국방 소프트웨어의 경우 테스트의 비중이 높고 소프트웨어의 복잡성, 테스트의 어려움, 그리고 매우 높은 신뢰성이 요구되는 특성을 감안할 때 국방 소프트웨어 시험 평가와 관련한 요구사항을 재검토하고 이를 토대로 적합하고 효율적인 테스트 프로세스를 정립해야 한다.

현재 개발되고 있는 국방 소프트웨어 테스트 성숙도 모델(MND-TMM)은 상기한 바와 같이 통합적 개발 프로세스 성숙도 모델인 CMMI의 구조를 근간으로 테스트 영역을 독립적으로 진단하고 개선안을 도출하기 위한 모델이다. 조직 전체의 테스트 프로세스 성숙도 및 핵심 영역별 성숙도 진단을 위한 구조를 제공하고, 우리나라 국방 시스템의 특성과 요구사항을 수용하는 한편 북미와 유럽 선진국들의 테스트 프로세스 모델들의 특징점을 살려서 국방 도메인에 최적의 테스트 성숙도 모델을 개발하고 있다.

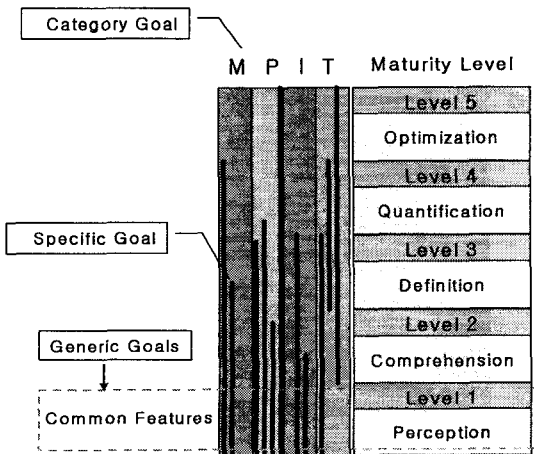


그림 1. MND-TMM 구성도

MND-TMM의 기본 구조는 그림 1과 같다. 5 단계의 성숙도 수준을 정의하여 현 수준을 진단하고 테스트 성숙도 수준을 개선하기 위해 각 단계별로 갖추어야 할 목적과 활동들을 정의한다.

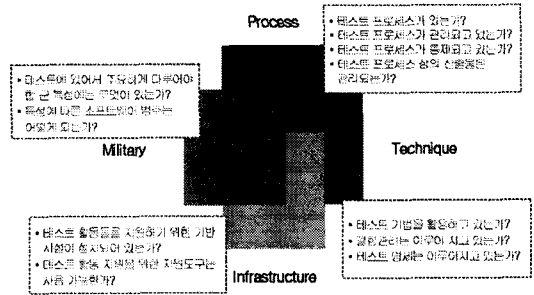


그림 2. MND-TMM의 범주

그림2는 네 가지 범주와 각 범주별 핵심 영역 (Key Process Area)들을 보여준다. 범주는 크게 군 도메인 특성(Military), 소프트웨어 테스트의 Process, Infrastructure, Technique으로 분류된다. 각 영역에서는 CMMI의 continuous representation과 같이 영역의 성숙도 수준을 정의할 수 있다[4].

상위 범주에서 특징적인 것이 Military 범주로서 국방 도메인에서의 특성을 담은 테스트 영역과 품질 특성들이 반영되는 부분이다. 예를 들어 국방 소프트웨어에 있어서 이 기종 간의 상호운용성이 중요한 핵심 영역으로 요구되고 있다. 국방 소프트웨어 개발 업체들과 군 조직 설문조사에 따르면 다양한 무기체계들마다 각각에 맞는 소프트웨어들이 개발되어 있는 상태이며 상호간의 데이터 교환 및 수집, 분석이 어렵고 재사용성도 매우 낮은 것으로 나타났다. 방위력 개선사업 관리 규정에서는 무기체계간 정보유통을 보장하기 위하여 상호운용성 획득정책 및 표준 데이터를 관리할 것과 시험평가를 통해 무기체계의 상호운용성을 검증해야 함을 규정하고 있다[17].

Process 카테고리에는 테스트 프로세스 관련된

것으로 테스트 전략과 생명주기 모델, 테스트 프로세스 관리 등의 핵심 영역을 아우른다. Infrastructure 카테고리는 테스트 환경, 테스트의 비용과 일정 산정, 테스트웨어 관리와 자동화 수준, 테스트 조직 등을 다룬다. Technique 카테고리는 수준별, 도메인별 테스트 기법과 테스트 명세 활동, 결함 관리 및 품질 제어 활동을 다룬다. 테스트 기법과 관련해서는 국방 소프트웨어의 특성상 임베디드 소프트웨어 테스트를 위한 기법이 깊게 연구되어야 할 것이다.

MND-TMM은 CMMI의 구조와 TMM, TPI 등 테스트 프로세스 개선 모델들의 특징점을 살리고 한국군 소프트웨어 개발을 위해 필요한 특징과 요건들을 연구함으로써 개발 전체적인 과정의 개선과 함께 테스트 프로세스의 개선을 꾀하고 있다. 특히 현실적이고 실제적인 모델 개발을 위하여, 문헌연구 외에도 국방 소프트웨어 개발 조직들을 방문·인터뷰하고 설문을 실시하여 각 조직의 소프트웨어 시험 성숙도 수준과 practice, 개선할 사항, 개발 중인 모델에 대한 요구사항을 수집하였다. 이를 통해 현재의 테스트 성숙도에 대한 인식과 개선점 등을 도출하고, 이를 바탕으로 모델의 기본 구조, 범주와 세부 영역들을 확인하고 세부 설계에 반영되어야 할 사항들을 파악하였다. 이렇게 만들어진 MND-TMM의 기본 설계를 바탕으로 세부 설계를 진행해가는 단계에 있다.

MND-TMM는 성공적인 국방 소프트웨어 개발 및 운용을 위해 국방 소프트웨어에 특화된 테스트 성숙도 모델을 제시하고 구체적인 지침을 중 우리나라 국방 소프트웨어의 품질 개선을 위한 실제적인 가이드라인을 제시하는 모델이 될 것이다.

참 고 문 헌

[1] TMM 적용방안에 관한 연구, 서울여자대학교,

Dec. 2003.

- [2] ISO/IEC 9126-1, International Organization for Standardization, 2001.
- [3] 정영은, 신석규, “소프트웨어 품질평가 표준 기술 및 동향,” TTA SW 시험인증센터, Nov. 2006.
- [4] 테스트 성숙도 모델기반 시스템 평가 기술 연구 1차년도 보고서, DSRC, Dec. 2006.
- [5] J-STD-016-1995(Standard for Information Technology Software Life Cycle Processes - Acquirer-Supplier Agreement), Sep. 1995.
- [6] Lewis Gray, “A Comparison of IEEE/EIA 12207, ISO/IEC 12207, J-STD-016 and MIL-STD-498 form Acquirers and Developers,” 1999.
- [7] MIL-STD-498 (Military Standard Software Development and Documentation), Dec. 1994.
- [8] An Introduction to IEEE/EIA 12207, Software Engineering Process Office.
- [9] 방위사업청 지침소프트웨어 개발 프로세스, 방위사업청, Jan. 2006.
- [10] 정연대 임진수 윤희병, “국방 CBD 방법론의 현재와 미래”, 정보과학회지 24권 9호, Sep. 2006.
- [11] 김남희, “소프트웨어 컴포넌트 시험기술 동향,” TTA 저널 제 88호, 2003.
- [12] CMMI for Development, Version 1.2 CMU/SEI-2006-TR-008, Aug. 2006.
- [13] Ron Swinkels, “A Comparison of TMM and other Test Process Improvement Models,” Frits Philips Institute, Nov. 2000.
- [14] Koomen and Pol, Test Process Improvement: a Practical Step-by-step Guide to Structured Testing, Addison-Wesley, 1999.
- [15] Jaap Cannegieter, An Introduction to TMMi: what to expect, SYSQA.
- [16] J. Jacobs, et al. Towards a Metrics Based Verification and Validation Maturity Model, Springer Berlin, 2002.
- [17] 방위력개선사업관리규정, 방위사업청, 2006.5.



송 경 희

- 1997년 고려대학교 영어교육과(문학사)
- 2002년~2005년 ㈜쓰리에스피 개발팀
- 2005년 ㈜에피토미 개발팀
- 2006년~현재 서울여자대학교 컴퓨터학과 (석사과정)
- 관심분야 : 소프트웨어 테스트, 소프트웨어 분석/설계, 컴퓨터 교육, C-IT융합



류 동 국

- 1994년 중앙대학교 컴퓨터공학과(학사)
- 1996년 중앙대학교 컴퓨터공학과(석사)
- 1996년~1999년 국방정보체계연구소 연구원
- 1999년~현재 국방과학연구소 선임연구원
- 관심분야 : 상호운용성 시험평가, 시스템 통합, 임베디드 소프트웨어 개발 방법론



이 병 길

- 1984년 University of Bridgeport 물리학과 졸업 (학사)
- 1993년 Auburn University 전산학과 졸업 (공학 석사)
- 1996년 Auburn University 전산학과 졸업 (공학 석사)
- 1998년~현재 서울여자대학교 정보미디어대학 컴퓨터학부 부교수
- 관심분야 : 소프트웨어 아키텍처, 형상관리, 소프트웨어 프로세스 개선



김 진 수

- 1991년~1999년 전남대학교 컴퓨터공학과(학사)
- 2000년~2002년 KAIST 컴퓨터공학과(석사)
- 2002년~2005년 한국생명공학연구원 연구원
- 2006년~현재 국방과학연구소 연구원