

국방 소프트웨어의 시험 프로세스 개선을 위한 국방 시험 성숙도 모델

(MND-TMM for Testing Process Improvements
of Defense Software)

박 준 영[†] 류 호 연^{**} 최 호 진^{***} 백 종 문^{***} 김 진 수^{****}
(Junyoung Park) (Hoyeon Ryu) (Ho-jin Choi) (Jongmoon Baik) (Jinsoo Kim)

요 약 국방 분야의 소프트웨어는 해당 도메인의 특성상 높은 수준의 품질을 요한다. 이를 보장하기 위해서는 잘 정의된 프로세스를 기반으로 하는 개발 및 시험 활동들이 이루어져야 한다. 소프트웨어 획득 과정에서 이들 활동들이 뒷받침되지 못하면 획득된 소프트웨어의 품질이 낮고, 군의 전투력이 저하된다. 본 논문에서는 시험 프로세스 개선을 통해 국방 소프트웨어의 품질을 향상하기 위한 국방 시험 성숙도 모델(MND-TMM: Ministry of National Defense - Testing Maturity Model)을 제안하고 설계된 내용을 소개한다. MND-TMM은 국방 도메인에서 시험과 관련된 문제점을 해결하기 위해 국방 소프트웨어, 국방 소프트웨어 개발 프로세스, 국방 소프트웨어 시험 프로세스를 반영하여 구성되었다. MND-TMM은 5개의 수준의 성숙도와 4개의 범주로 구성되며 각 범주에는 관련 시험 프로세스 영역(TPA: Testing Process Area)들이 포함된다. MND-TMM은 국방 소프트웨어 조직의 시험 성숙도를 평가하는 것을 돕고 소프트웨어 테스트 프로세스를 개선하기 위한 가이드라인을 제공할 것으로 기대된다.

키워드 : 프로세스, 테스트, TMM, TPI, 국방

Abstract Software in defense domain requires high quality since defense specific characteristics. To assure high quality products, development and testing activities based on well defined process must be performed. If those activities cannot support software acquisition process, the quality of acquired software product is low and combat power decreases. In this paper, we propose MND-TMM(Ministry of National Defense - Testing Maturity Model), which can help enhance software quality through testing process improvements. This paper also introduces the contents of MND-TMM architecture. MND-TMM is constituted to reflect the characteristics of defense software, development process, and testing process so as to solve the problems associated with software testing. MND-TMM is comprised of 5 maturity levels and 4 categories which have number of related TPA(Testing Process Area)s. It is expected that MND-TMM can help assess testing maturity of defense software organizations and provides guidelines to improve software testing process.

Key words : Process, Testing, TMM, TPI, Defense

· 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.
(2008-SW-11-DJ-03)

· 이 논문은 2007 한국컴퓨터종합학술대회에서 '국방 소프트웨어의 시험 프로세스 개선을 위한 국방 시험 성숙도 모델'의 제목으로 발표된 논문을 확장한 것임

**** 정 회 원 : 국방과학연구소 4체계본부 연구원
jskim421@add.re.kr

논문접수 : 2007년 10월 2일
심사완료 : 2008년 4월 7일

† 학생회원 : 한국정보통신대학교 공학부
nitra@icu.ac.kr

** 정 회 원 : 한국정보통신대학교 공학부 교수
hoyeon@icu.ac.kr

*** 종신회원 : 한국정보통신대학교 공학부 교수
hjchoi@icu.ac.kr
jbaik@icu.ac.kr

Copyright©2008 한국정보과학회: 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제35권 제5호(2008.5)

1. 서론

정보기술의 급격한 발전에 맞추어 현대전의 전장환경은 전투원, 부대, 그리고 자동화된 무기체계들이 네트워크에 통합되어 전투력을 발휘하는 방향으로 발전하고 있다. 미군이 컴퓨터 기술에 의한 지휘통제체계의 혁신을 통해 NCW(Network Centric Warfare)를 구현하는 것이 대표적이라고 할 수 있다[1].

소프트웨어는 이와 같은 전장환경의 변화를 좌우하는 중요한 요소이다. 국방 환경에서 소프트웨어는 작게는 탄약의 신관을 제어하는 기능에서부터 항공기나 전투함의 전투체계를 운용하고 데이터를 수집/가공/교환하여 다수의 체계들을 연결하는 기능까지 제공한다[2].

소프트웨어가 설치된 무기체계들은 소프트웨어에 의해 성능이 결정되는 반면에 소프트웨어의 사소한 결함도 큰 인적 또는 물적 손실을 가져올 수가 있다. 따라서 고품질의 국방 소프트웨어를 획득하는 것이 매우 중요하다고 볼 수 있으며, 국방 조직들도 소프트웨어의 공학 발전을 통해 소프트웨어의 품질을 개선하기 위해 노력하고 있다.¹⁾

이 중에서도 개발 비용의 40~50%의 비용을 차지하는[3] 시험 프로세스를 개선하는 것이 중요한데, 개발 과정에서 체계적인 시험을 통해 결함을 미리 제거한다면 신뢰성 있는 고품질의 소프트웨어가 개발될 수 있기 때문이다.

그러나 지금까지는 국방 소프트웨어를 어떻게 시험해야 하는 것인가에 대한 체계적인 접근이 부족했다. 무기 체계에 내장된 소프트웨어들은 하드웨어 성능에 대한 시험이 끝난 다음에 부차적으로 기능성을 확인하는 정도의 시험만을 거쳤다. 또한 정보체계들도 정보기술 분야에 정통한 시험 인력을 갖추지 못한 한시적인 시험 조직에 의한 제한적인 시험만을 거치고 있다[4]. 그 결과로 소프트웨어 개발 과정에서 발견되지 못한 결함이 운용 과정에서 나타나고, 결함들을 수정하고 개선하는 동안에 정상적인 운용이 불가하여 군 전투력의 저하가 발생하기도 했다[5]. 이러한 문제점을 해결하기 위해서는 국방 소프트웨어를 획득하는 과정에서 개발/시험 조직의 시험 프로세스를 개선하여 소프트웨어의 품질을 향상시키기 위한 방법이 필요하다.

본 논문에서는 국방 소프트웨어의 시험 프로세스를 개선하기 위한 국방 시험 성숙도 모델(MND-TMM: Ministry of National Defense - Testing Maturity Model)

을 제안한다. MND-TMM은 기존의 프로세스 및 시험 성숙도 모델을 분석하고 문헌조사, 설문, 그리고 인터뷰를 통해 분석된 국방 소프트웨어 개발 및 시험의 특성들과 문제점을 반영하여 국방 소프트웨어 관련 조직에 적합하도록 구성되었다.

MND-TMM은 5개의 성숙도 수준과 특정 분야의 시험 프로세스를 개선하기 위한 10 개의 시험 프로세스 영역(TPA: Testing Process Area)들로 구성되어 있다. TPA들은 포함하는 내용에 따라 4개의 범주로 구분된다. MND-TMM은 국방 소프트웨어 관련 조직의 시험 성숙도를 평가하여 시험 프로세스 개선의 베이스라인을 제공하고, 나아가 국방 소프트웨어 신뢰성 및 품질 향상에 기여할 것으로 기대된다. 또한 국방 소프트웨어 사업 관리조직은 획득 대상 소프트웨어의 수준에 적합한 시험 성숙도를 가진 개발 조직을 선정하기 위해 MND-TMM을 평가 기준으로 활용할 수도 있다.

본 논문의 구성은 다음과 같다. 2장에서는 국방 소프트웨어의 분류와 특징을 기술하고 국방 소프트웨어 시험에서의 문제점을 분석하였다. 3장에서는 기존의 시험 성숙도 모델들의 특성과 장단점을 분석하고, 4장은 국방도메인에서의 시험 프로세스 개선을 위한 성숙도 모델인 MND-TMM을 제안한다. 5장은 결론과 향후 연구사항을 기술한다.

2. 국방 도메인 분석

본 장에서는 국방 소프트웨어를 분류하고, 민간 분야와 차별화되는 종류인 무기 소프트웨어의 특징을 소개한다. 또한 문헌조사와 설문/인터뷰를 통해 분석된 국방 소프트웨어 시험 단계에서의 문제점을 기술한다.

2.1 국방 소프트웨어

전장 환경의 변화에 따른 무기체계의 발전에 맞추어 새로운 분류체계가 연구되고 있으나, 기존의 분류[6,7]에 따라 다음 표 1과 같이 국방 소프트웨어를 구분할 수 있다.

국방정보체계 소프트웨어는 전장관리 정보체계 소프트웨어, 자원관리 정보체계 소프트웨어, 기반체계 소프트웨어로 분류된다. 국방정보체계는 국방 업무와 관련된 정보를 수집·가공·전달·전시하는 기능들을 수행하는 컴퓨터 하드웨어와 소프트웨어, 데이터 및 통신 수단의 조

표 1 국방 소프트웨어의 분류

국방정보체계 소프트웨어	전장관리 정보체계 소프트웨어
	자원관리 정보체계 소프트웨어
	기반체계 소프트웨어
내장형 소프트웨어	무기체계 내장형 소프트웨어
	비무기체계 내장형 소프트웨어

1) 미국은 1984년에 SEI를 설립하여 국방 조직, 정부 조직, 산업계, 학계가 소프트웨어 집약 체계를 개선하는 것을 돕고 있으며, 한국은 2006년에 방위사업청이 국방소프트웨어 설계특화연구센터를 설립하여 소프트웨어 공학 분야의 연구를 지원하고 있다.

합이며[8], 전장관리 정보체계 소프트웨어는 전시 또는 평시에 지휘관 및 참모에게 필요한 정보를 제공하여 작전에 중요한 결정을 내리는 것을 보조한다. 자원관리 정보체계 소프트웨어는 평시의 기능별 국방 업무를 수행하기 위한 것이며, 기반체계 소프트웨어는 정보통신망, 정보보호, 상호운용성을 지원하기 위한 것이다.

내장형 소프트웨어는 무기/비무기체계에 내장된 소프트웨어이다. 무기체계 내장형 소프트웨어는 전차의 화력 통제체계, 항공기의 항법장치 등과 같이 각종 무기체계에 내장되어 해당 장비의 임무에 전용으로 사용되는 소프트웨어를 말하며[8], 비무기체계 내장형 소프트웨어는 전투상황과 관련 없이 군의 일상적인 업무에 사용되는 내장형 소프트웨어를 말한다.

2.2 무기체계 소프트웨어의 특징

국방 소프트웨어는 민간 분야의 소프트웨어와는 차별화되는 특성들을 가지고 있다. 특히, 전장관리 정보체계 소프트웨어와 무기체계 내장형 소프트웨어를 포함하는 무기체계 소프트웨어[6]에 이 특성들이 뚜렷하게 나타난다. 무기체계 소프트웨어는 상용제품 또는 관급제품에 비하여 엄격한 요구사항 또는 품질기준을 적용받는다. 이것은 다음과 같은 무기체계 소프트웨어가 가진 다음과 같은 특징들에 기인한다.

- 임무 최우선 체계 : 전투지휘를 지원하는 소프트웨어의 오류는 다수의 전투원에게 생명에 직접적인 영향을 미친다. 이것은 금전적 피해보다 심각하다.
- 체계들의 체계 : 소프트웨어 체계가 단독으로 구성되지 않고, 다수의 부체계들이 전체 체계를 구성한다. 부체계들은 전체 체계의 기능을 발휘하기 위해 데이터를 생산/교환한다.
- 소프트웨어 집약 체계 : 소프트웨어가 국방정보체계의 기능, 비용, 개발 위험, 개발 기간 등에 대부분의 비중을 차지한다.
- 하드웨어와 소프트웨어의 복합 체계 : 소프트웨어는 군 환경에 적합하게 제작된 하드웨어에 기반으로 하여 기능을 발휘한다.
- 상호운용성 : 부체계간의 데이터 교환이 불가능하다면 전체 소프트웨어 체계가 기능을 발휘하지 못한다. 체계, 부대, 군별, 국가 등 다양한 관점에서 상호운용성이 고려되어야 한다.

2.3 국방 소프트웨어 시험과 그 문제점

국방 소프트웨어는 획득과정에서 탐색개발, 개발, 시험평가의 과정을 거친다. 그러나 소프트웨어 시험이 체계적으로 수행되지 않는 문제점이 자주 발생하고 있다.

군은 시험평가를 통해 무기체계의 성능, 기술, 품질을 시험하고 운용관리적 측면에서의 요구사항 및 개발목표의 충족여부와 운용적합성, 효율성, 안전성 등을 확인

검증한다[9]. 그러나 국방 관련 규정/지침[7,9] 등에 하드웨어 무기체계의 시험 절차는 잘 정의되어 있지만 소프트웨어 시험에 대한 내용은 매우 부족한 실정이다. 또한, 국방 조직 내의 소프트웨어에 대한 이해와 시험 능력이 미흡하여[6], 소프트웨어 시험이 체계적으로 이루어지지 않고 있다.

체계적인 시험이 부족한 것은 개발 단계에서도 마찬가지이다. 무기체계 소프트웨어를 개발하기 위해 사용되는 국방 CBD 방법론[10], 소프트웨어 개발 프로세스[11] 등에도 소프트웨어 시험의 구체적인 지침과 방법은 포함되어 있지 않다. 따라서 단위시험, 통합시험, 체계시험 등에서 어떠한 시험 활동들을 수행하는 지는 개발 조직의 역량에 달려 있는 상태이다. 그러나 국방 소프트웨어 개발 조직 내에서도 소프트웨어 시험 활동이 체계적으로 수행되지 않고 있으며, 소프트웨어 시험 조직의 역량도 미흡한 실정이다[12].

표 2는 과거의 국방 소프트웨어 시험평가 사례와 문제점들을 보여준다. 과거의 시험평가 사례는 수준별 시험이 생략되거나 미흡하게 수행되고, 시험환경이 제대로 구축되지 않았다. 조직 측면에서는 시험평가 전문인력이 부족하고, 시험평가의 독립성이 보장되어있지 않았으며, 사업관리가 효과적으로 이루어지지 않은 사례들이 나타나 있다.

국방 소프트웨어 조직들의 시험 성숙도와 프랙티스 등을 수집하기 위해 실시한 설문 및 인터뷰의 결과[12]에 따르면, 국방 조직들이 개발이 완료된 소프트웨어를 시험하는 과정에서 어려움을 겪는 것에 비해, 국방 소프트웨어 개발 조직들은 개발 과정에서의 문제점들을 보였다.

그림 1과 2는 국방 소프트웨어 개발업체를 대상으로 이루어진 설문 및 인터뷰의 결과를 나타낸 것이다.

설문 응답자들과 인터뷰에 응한 조직들은 시간과 비용의 필요, 유용성의 부재, 전문기술 또는 지식의 부족

표 2 국방 정보체계 시험평가 사례[4]

구분	문 제 점
00 정보체계	부적절한 단위시험 구조적 시험 기법의 적용이 미흡 형식적 기술시험으로 기술적 신뢰도 저하 운용시험 환경과 데이터 구축이 어려움
00 위게임	단위시험 미흡 미흡한 통합시험 운용시험환경 구축이 부적절
00 정보처리체계	시험평가 전문인력의 부재 시험평가의 독립성이 매우 저조함
00 지휘통제체계	기술시험 미흡 사업관리의 문제 (기간, 인력, 예산 등) 시험평가 전문인력의 부재

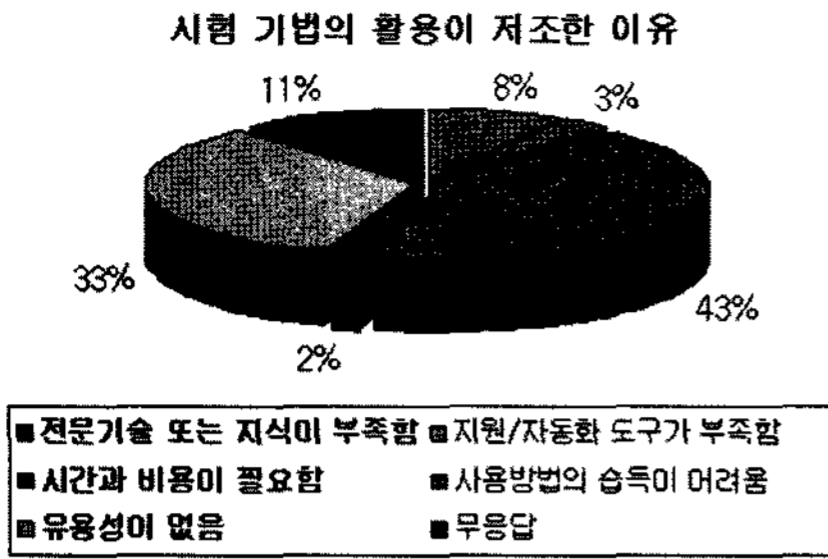


그림 1 시험 기법의 활용이 저조한 이유

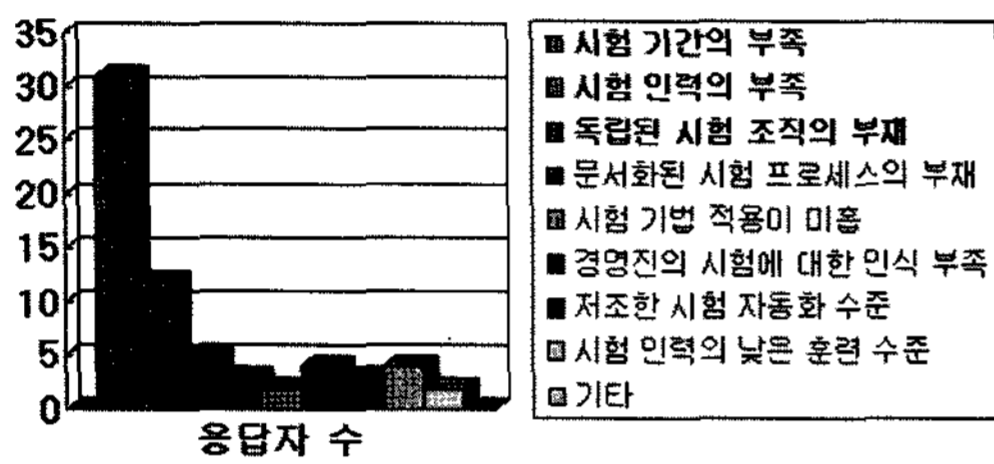


그림 2 소프트웨어 시험에서의 어려운 점

등을 시험 기법의 활용이 저조한 주요한 문제점으로 꼽았으며, 소프트웨어 시험에서 겪는 어려움들로는 시험 기간 및 시험 인력의 부족, 독립된 시험 조직의 부재 등이 대표적이었다.

시험 기법 외에 국방 소프트웨어 개발 조직들이 소프트웨어 시험과 관련하여 제기하는 문제점들은 다음과 같다.

- 기능 위주의 시험 : 소프트웨어에 대한 시험이 단순한 기능 위주로 진행되어 품질특성에 대한 고려가 부족하다.
- 운용시험에서의 결함 발생 : 개발시험까지 발견되지 않았던 문제들이 운용시험 단계에서 다수 발견됨으로써 이를 해결하기 위해 실제 운용이 지연되는 경우가 있다. 시험단계에서 실제 운용환경과 유사한 시험환경을 제공하려는 노력이 미흡하다.
- 시험 인력의 부족과 교육의 미흡 : 개발/시험 조직이 전문적인 지식을 가진 시험인력을 가지지 못하고 있으며 체계적인 시험에 관련된 교육을 제공하고 있지 않다.
- 독립된 시험 조직/프로세스의 부재 : 개발 조직에서 시험의 독립성을 보장하기 위해 분리된 시험 조직을 갖추지 않아서, 객관적이고 독립적인 시험이 미흡하다.
- 품질 특성 고려가 미흡함 : 요구사항 단계에서 품질 요소에 대한 요구가 식별되지 않기 때문에 시험에도 품질 요소가 반영되지 않는다.
- 운용시험 환경 구축이 어려움 : 운용시험을 위한 데이

타 입력, 운용자 확보, 시험 장비 설치 등이 미흡하여, 운용환경에서의 문제점을 찾아내는 시험이 불충분하다.

“시험의 수준이 바로 체계의 수준이다”[13]라는 관점을 고려할 때, 프로세스의 개선을 통한 체계적인 시험이 국방 소프트웨어 사업 성공의 열쇠라고 할 수 있다.

3. 관련 연구

본 장에서는 기존의 프로세스 성숙도 모델들과 시험 성숙도 모델들을 소개하고 민간 분야에서 사용되고 있는 이 모델들을 통한 시험 프로세스 개선의 한계에 대해 기술한다.

3.1 프로세스 성숙도 모델

조직이 가진 소프트웨어 프로세스의 성숙도를 평가하고 프로세스를 개선하기 위한 대표적인 프로세스 성숙도 모델로는 CMM(Capability Maturity Model)[14]과 CMMI(Capability Maturity Model Integration)[15]가 널리 사용되고 있다. CMM은 다섯 단계의 성숙도 수준을 가지며, 핵심 프로세스 영역별로 목적과 세부 행위들을 정의한다. CMMI는 소프트웨어 개발조직들이 프로세스 개선에 중복하여 투자하는 부담을 없애기 위해 SW-CMM, SE-CMM, SA-CMM, IPD-CMM 등의 다수의 모델들을 통합한 성숙도 모델로서, CMM과 유사한 단계적 표현과 프로세스 영역별로 6개의 능력 단계를 가지는 연속적 표현을 제공한다.

3.2 시험 성숙도 모델

본 절에서는 제안하고자 하는 MND-TMM의 근간이 되는 주요 시험 성숙도 모델들을 소개하고 이들의 문제점 및 제약사항에 대해 기술한다.

3.2.1 TMM(Testing Maturity Model)

TMM은 1996년에 일리노이 공대의 Burnstein 교수에 의해 개발된 시험 성숙도 모델이다. 이 모델은 CMM과 같은 개념의 성숙도 수준을 갖고 있으며, 시험 프로세스 향상을 위한 체계적 접근방법을 제공한다[16]. 즉, TMM은 CMM과 결합하여 사용될 수 있는 보조적 모델로서 시험 프로세스의 성숙도 수준을 확인하고, 개선될 수 있는 시험 분야를 식별하고, 개선을 구현하기 위한 로드맵을 제공하며, 개선 결과를 측정할 수 있는 수단을 제공한다.

TMM은 5단계의 성숙도 수준을 가지며, 각 수준마다 달성해야 하는 성숙도 목표를 가지고 있다. 성숙도 목표는 부목표들에 의해 지원되며, 부목표들은 활동·작업·책임을 수행함으로써 달성된다. 활동·작업·책임을 다시 관리자, 개발자·테스터, 사용자·클라이언트 영역으로 분류된다. 그림 3은 TMM의 내부 구조를 보여준다.

3.2.2 TPI(Test Process Improvement)

TPI는 Kooman과 Pol에 의해 1997년에 개발된 시험

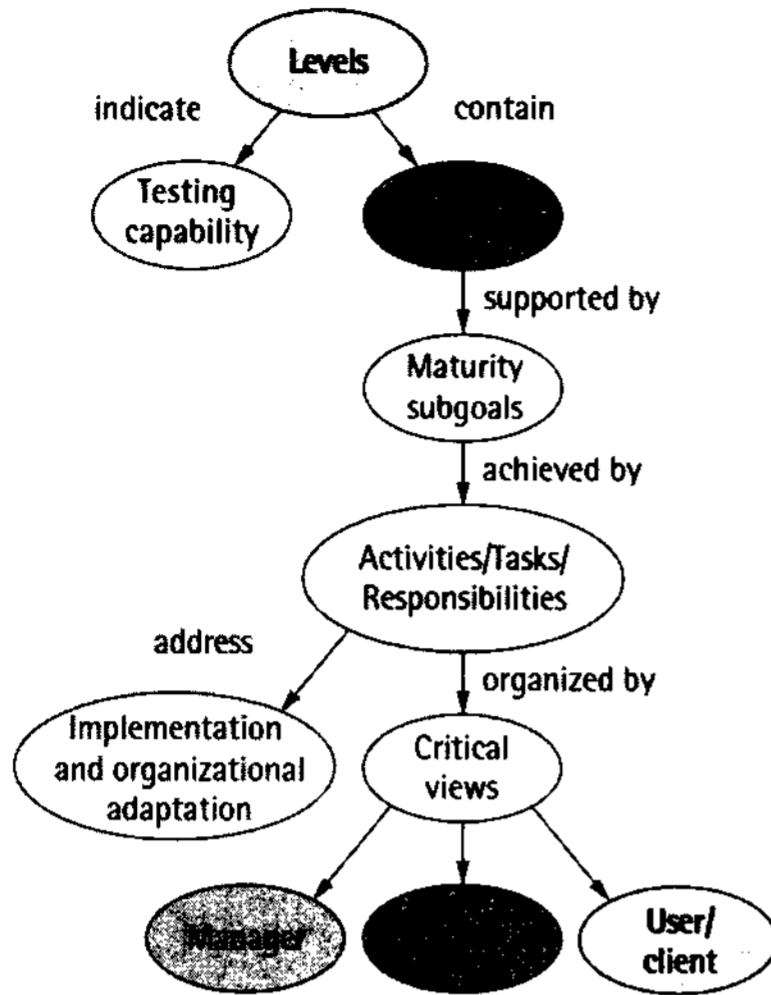


그림 3 TMM 내부 구조[16]

성숙도 모델로서 CMMI와 유사한 연속적 표현을 제공한다[17]. TPI는 3개의 기본 성숙도 수준과 14 단계의 스케일을 가지며, 시험 성숙도 매트릭스에 핵심 영역별 성숙도 수준을 정의한다. 또한, 성숙도 수준에 필요한 요구 사항들을 담은 체크포인트와 프로세스 개선을 위한 팁과 아이디어를 담은 개선 제안사항이 추가로 제공된다.

TPI는 시험환경, 사무환경, 보고, 결함관리, 테스트웨어 관리 등과 같이 TMM이 다루지 않는 영역을 포함하는 20개의 핵심 영역을 정의한다. 핵심 영역에 필요한 시험 활동들은 A~D의 단계로 구분되어 시험 성숙도 매트릭스의 상응하는 스케일에 할당된다. 소프트웨어 조직은 시험 성숙도 매트릭스를 이용하여 조직의 시험 프로세스의 수준을 파악하고, 개선 전략을 수립/실행할 수 있다. 그림 4는 TPI의 프레임워크와 시험 성숙도 매트릭스의 개념을 설명한다.

TPI는 조직이 가진 시험 프로세스의 수준을 평가하여 강점과 약점을 인식하고 개선사항을 식별하고, 구현 전략을 수립/시행할 수 있게 한다. TPI는 성숙도 수준을 연속적 표현으로 나타내기 때문에 특정한 시험 영역을 선택하여 성숙도를 향상시키는 데에 적합하다.

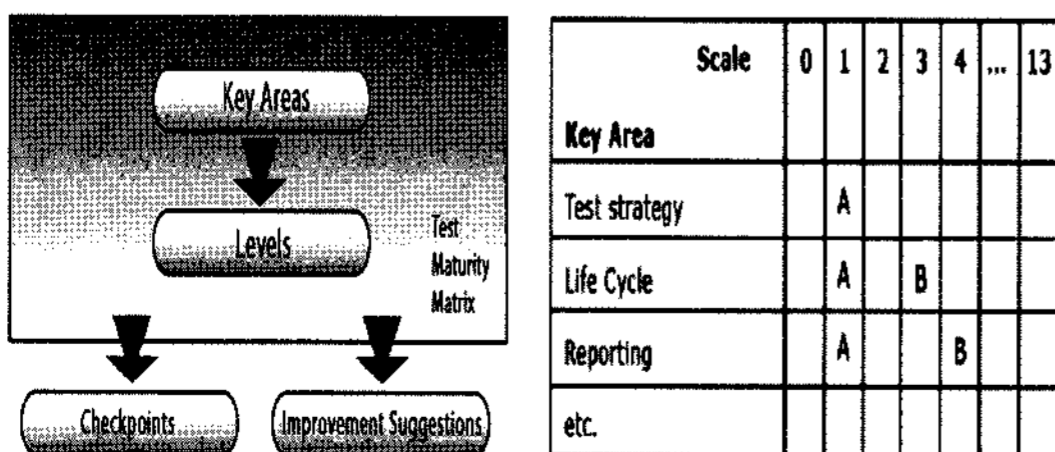


그림 4 TPI 프레임워크와 시험 성숙도 매트릭스의 개념[17]

3.3 기존 모델들의 문제점

프로세스 성숙도 모델들은 소프트웨어 개발의 전 과정에 대해 기술하고 있지만, 전체 프로젝트 비용의 40~50%를 차지하는 시험 분야를 다루는 비중은 매우 낮다 [18]. 따라서 프로세스의 성숙도 수준이 시험 프로세스의 성숙도 수준을 의미하지는 않는다. CMM은 품질보증, 결함관리 등의 프로세스 영역을 가지지만, 이것들은 시험을 직접적으로 다루지는 않고 시험 분야에 대한 기술이 부족하다. CMMI는 CMM보다 시험에 대해 많은 부분을 고려하고 있으나, 소규모의 개발조직의 시험 프로세스에 적용하기에는 부담이 크다는 단점이 있으며 [19] 국방 소프트웨어 개발 및 시험 조직에 적용하기에는 국방 분야의 개발 프로세스 및 조직 특성과 같은 것들이 반영되지 않았다는 문제가 있다.

TMM은 시험 프로세스 개선을 위한 완전한 참고서는 아니며 시험 프로세스 구현에는 경험 많은 리더가 필요하다. 또한, TMM은 인적 관리와 시험 조직에 대한 기술이 비교적 부족하고, 시험 장비, 시험 체계, 테스트베드 등의 시험 기반시설에 대한 영역을 다루고 있지 않다[20]. 마지막으로 TMM은 단계적 표현을 사용하기 때문에 조직 전체의 프로세스 수준만 나타나고, 개선이 필요한 프로세스 영역별 수준이 나타나지 않는다. 따라서 앞서 언급한 시험 인력의 부족과 교육의 미흡, 조직 및 프로세스의 부재 등과 같은 국방 소프트웨어 시험에서 식별된 문제점을 해결하기에는 한계가 있다.

TPI에는 모든 핵심영역에 공통적으로 적용되는 성숙도 수준이 정의되지 않았고, 성숙도 수준의 개념이 CMMI 등과 다르기 때문에 기존의 모델과 함께 사용하기에는 부적합하다. 또한 TMM이 CMM이나 Beizer가 제시한 테스트에 대한 태도[21] 등을 참조한 수준 정의를 가지는 데 비해서, TPI는 성숙도 수준을 구분하는데 있어서 충분한 근거를 제공하지 못한다.

4. MND-TMM의 설계

본 장에서는 MND-TMM에 대해 소개하고 설계가 완료되거나 진행중인 구성요소들에 대해 기술한다. 또한 식별된 TPA들을 소개하고 모델의 활용방안을 제시한다.

4.1 MND-TMM 개요

MND-TMM은 국방 소프트웨어 시험 프로세스를 개선시키고 체계적이고 효과적인 시험을 거친 소프트웨어의 품질과 신뢰성을 향상시키기 위한 성숙도 모델이다. 국방 분야의 소프트웨어 개발 및 시험에 적합한 모델을 개발하기 위하여 다음과 같은 다양한 관련 모델 및 표준들이 조사 및 분석되었다.

- 방법론 및 표준 : ISO/IEC 12207, MIL-STD-498, 국방 CBD 방법론(ADDMe), 방사청 소프트웨어 개발

프로세스, DO-178B 등을 바탕으로 국방 소프트웨어 개발 프로세스와 시험기법 및 활동에 대해 조사하고 신뢰성 확보 기법을 파악하였다.

- 성숙도 모델 : CMM, CMMI와 같은 프로세스 성숙도 모델과 TMM, TPI 등과 같은 테스트 성숙도 모델을 분석하여 테스트 관련 프로세스 영역을 추출하고 MND-TMM의 TPA를 정의하는 데에 활용하였다.
- 상호운용성 모델: 군의 중요한 관심사인 상호운용성을 반영하기 위하여 LISI(Level of Information Systems Interoperability)와 SoSI(System of Systems Interoperability) 등의 상호운용성 모델들의 성숙도 정의와 평가절차를 조사 분석하였다.
- 관련 핵심 요소기술: VBSE(Value-Based Software Engineering), COQ(Cost of Quality), ROI(Return on Investment) 등에 대한 조사 분석을 통하여 가치 기반의 비용대비 효과를 모델 설계에 반영하였다.

4.2 MND-TMM의 구성

MND-TMM은 국방 소프트웨어 시험 프로세스의 성숙도를 5개의 수준과 4 개의 범주로 구분된 시험 프로세스 영역들을 가진다. 각각의 TPA는 성숙도 수준을 결정하는 시험/지원/관리 활동들을 포함한다.

4.2.1 성숙도 수준

MND-TMM은 TPA들에 적용되는 5 단계의 성숙도 수준을 가진다. 각각의 성숙도 수준과 해당 성숙도 수준에서 달성해야 하는 일반 목표는 다음과 같다.

1 수준에서 조직은 국방 분야에서 사용되는 표준들을 알고 있으며 시험의 필요성을 인식한다. 하지만, 시험 프로세스가 정립되지 않았으며, 시험 활동들이 임의적으로 이루어진다.

2 수준에서는 개별 프로젝트에 시험 프로세스가 사용된다. 시험 프로세스에 따라 기본적인 시험 활동이 이루어지며, 국방 분야의 개발 프로세스에 따른 개발 및 시험이 가능하다.

3 수준에서는 조직의 표준 시험 프로세스가 정의되어 있다. 각각의 프로젝트 팀은 조직의 표준 시험 프로세스를 테일러링 지침에 따라 변경하여 사용한다. 즉, 국방 표준을 근간으로 하여 세부 목표 및 활동들을 정의 및 변경하여 사용할 수 있다.

4 수준과 5 수준은 지속적인 시험 프로세스 개선과 품질 향상을 위한 단계이다. 4 수준은 시험 프로세스가 다양한 수준의 매트릭에 따라 정량적으로 평가, 관리되고 5 수준에서는 지속적인 시험 프로세스 개선 활동이 이루어진다.

4.2.2 시험 프로세스 영역

MND-TMM는 4 개의 범주와 각각의 범주에 속하는 10개의 TPA들을 가진다. 이 TPA들은 관련 모델 및

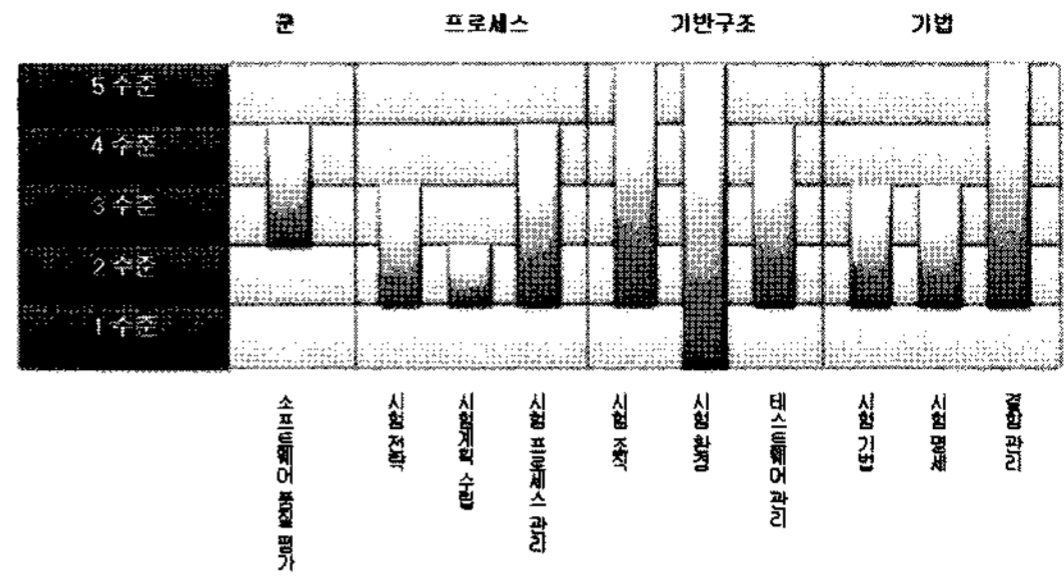


그림 5 MND-TMM의 범주와 TPA

표준과 국방 소프트웨어 시험의 특징을 바탕으로 TPI의 핵심 영역 중에서 선택되었는데, 이것은 TPI가 소프트웨어 시험을 20개의 자세한 항목으로 구분하고 있기 때문이다. 추가적으로 국방 소프트웨어만의 차별화되는 특성은 군 범주의 TPA로 추가하였다. 범주에 따른 TPA의 분류는 다음의 그림 5와 같다.

- 프로세스(Process): 시험 프로세스와 관련된 TPA들을 포함한다. 이 범주는 시험 전략, 시험계획 수립, 시험 프로세스 관리 TPA를 가진다.
- 기법(Technique): 시험에 사용하는 기법들과 관련된 TPA들을 포함한다. 시험 명세 기법, 시험 기법, 결함 관리가 포함된다.
- 기반구조(Infrastructure): 테스트 수행을 위해 필요한 제반 환경과 관련된 TPA들을 포함한다. 시험 환경, 테스트웨어 관리, 시험 조직이 해당된다.
- 군(Military): 민간 영역과 다른 국방 소프트웨어 시험에 관련된 TPA들을 포함한다. 현재는 소프트웨어 품질 평가가 포함되어 있다.

이들 TPA들은 앞에서 언급된 국방 소프트웨어 시험과 그 문제점을 개선하는 것을 고려하여 선정하였다. 국방 조직들이 국방 소프트웨어 시험에서 부족한 시험활동과 이를 개선하기 위한 TPA가 표 3에 제시되어 있다.

소프트웨어 품질 평가 TPA는 품질 특성들을 식별하고 정량화하여 시험의 목적을 명확히 한다. 시험 전략은 수준별 시험 대상과 노력의 할당을 결정하여 효율성을 높인다. 시험계획 수립, 시험 기법, 시험 명세는 전략을 구체화하여 수준별로 수행해야 할 시험 활동들을 구체

표 3 국방 소프트웨어 시험의 이슈와 관련된 TPA

이슈	선정된 TPA
시험의 목적	소프트웨어 품질 평가
시험의 효율성	시험 전략
시험의 내용	시험계획 수립, 시험 기법, 시험 명세
시험의 관리	시험 조직, 시험 프로세스 관리, 테스트웨어 관리, 결함 관리
시험의 환경	시험 환경

화하기 위해 선정되었으며, 시험 활동의 수행을 관리하기 위해 시험 조직, 시험 프로세스 관리, 테스트웨어 관리, 결함 관리 TPA가 선정되었다. 마지막으로 실제 환경과 유사한 시험 환경을 구축하기 위해 시험 환경 TPA를 선정하였다.

4.2.3 시험 프로세스 영역의 구조

TPA의 구조는 그림 6에서 나타난 것처럼 연속적 표현과 단계적 표현이 혼합된 형태이다. 각각의 TPA는 달성해야 하는 특정 목적을 가지며, 특정 목적들은 능력

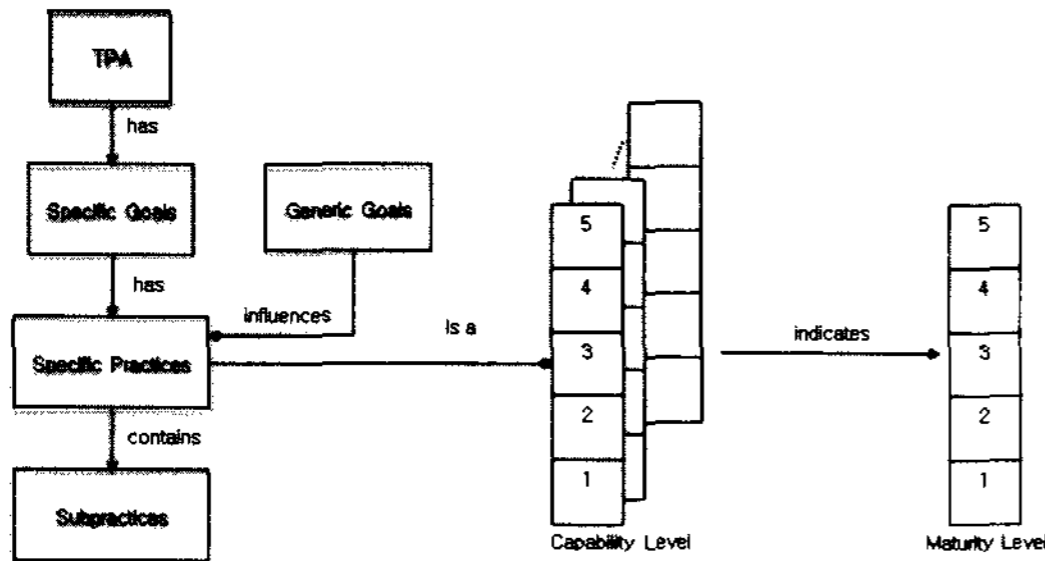


그림 6 MND-TMM의 구조

표 4 TPA별 특정 목적의 목록

TPA	특정 목적
소프트웨어 품질 평가	1. 소프트웨어 요구사항을 파악한다. 2. 소프트웨어 품질 모델을 개발한다.
시험 전략	1. 리스크를 식별한다. 2. 품질 특성을 분류한다. 3. 시험 견적을 수행한다. 4. 시험 자원을 할당한다.
시험계획 수립	1. 소프트웨어 프로세스 생명주기를 식별한다. 2. 전체 시험 계획을 작성한다.
시험 프로세스 관리	1. 시험 프로세스의 실행을 감독하고 조정한다. 2. 시험 프로세스의 성능을 측정한다. 3. 시험 프로세스를 재활용한다.
시험 조직	1. 시험 프로세스를 정립한다. 2. 시험 조직을 체계화한다. 3. 시험 지원 조직을 구성한다. 4. 다양한 수준의 의사소통 체계를 정립한다. 5. 시험 인원을 위한 훈련을 제공한다. 6. 경영진의 참여와 지원을 확보한다.
시험 환경	1. 시설과 장비를 제공한다. 2. 시험 도구를 제공한다. 3. 시험 데이터를 준비한다. 4. 운용요원을 준비한다.
테스트웨어 관리	1. 테스트웨어를 정의한다. 2. 테스트웨어를 수집한다. 3. 테스트웨어를 재활용한다.
시험 기법	1. 요구되는 수준의 시험 기법을 적용한다
시험 명세	2. 시험 명세 기법을 적용한다.
결함 관리	1. 결함의 내용을 보고한다. 2. 결함 보고를 관리한다. 3. 결함예방 활동을 수행한다.

수준별로 구분된 특정 프랙티스들에 의해 달성된다. 특정 프랙티스들은 결정된 능력수준에 해당하는 일반목표를 반영하여 기술되며 세부 프랙티스에 의해 구체적으로 수행된다. 한 개의 TPA의 능력수준을 개선하기 위해서는 상위 수준의 특정 프랙티스들을 달성하고, 조직 전체의 성숙도를 얻기 위해서는 해당 수준 이하의 모든 특정 프랙티스를 달성한다.

하나의 TPA는 하나 이상의 특정 목적을 가지며, 특정 목적은 하나 이상의 특정 프랙티스를 가진다. 각각의 TPA에 선정된 특정 목적의 목록은 표 4에 정리되어 있다.

특정 프랙티스들은 능력수준이 부여되며, 능력수준별로 하나 이상의 특정 프랙티스가 할당된다. 표 5는 소프트웨어 품질 평가 TPA의 특정 목적들과 능력수준별 특정 프랙티스, 그리고 특정 프랙티스를 달성하기 위한 세부 프랙티스들을 보여준다. SG는 Specific Goal(특정 목적)을 말하며, SP는 Specific Practice(특정 프랙티스)를 뜻한다. SP에는 특정 목적별로 일련번호가 부여되며, 마지막 숫자는 특정 프랙티스의 능력수준을 뜻한다. 표 5

표 5 소프트웨어 품질 평가 TPA

SG 1 소프트웨어 요구사항을 파악한다.	SP 1.1-2 소프트웨어가 만족해야 하는 요구사항을 수집한다. Subpractices 1. 체계 개발 요구사항을 수집한다. 2. 소프트웨어 개발 요구사항을 수집한다. 3. 야전 요구사항을 수집한다. 4. 운용환경이 가진 제한사항을 조사한다. 5. 운용시험평가 사례를 수집한다.
SP 1.2-3 소프트웨어 요구사항을 문서화한다.	Subpractices 1. 시험 대상인 소프트웨어의 요구사항을 종합/분류하여 문서화한다. 2. 테스트 케이스 작성에 요구사항이 활용된다.
SP 1.2-4 소프트웨어 요구사항을 재사용한다.	Subpractices 1. 요구사항 수집 결과를 보존/관리한다. 2. 과거 자료 참조하여 체계의 요구사항을 쉽게 식별한다.
SG 2 소프트웨어 품질 모델을 개발하여 사용한다.	SP 2.1-3 품질 모델을 개발한다. Subpractices 1. 소프트웨어 품질 특성의 종류와 내용을 이해한다. 2. 소프트웨어 품질 특성을 선정한다. 3. 시험 대상의 품질 요소와 메트릭/평가표를 개발한다.
SP 2.2-4 품질 목표를 개발한다.	Subpractices 1. 품질 모델에 기초하여 시험 대상인 소프트웨어가 달성해야 하는 품질 목표를 설정한다.
SP 2.3-4 도메인 품질 모델을 개발한다.	Subpractices 1. 도메인에 최적화된 품질 모델을 개발 또는 수집한다. 2. 품질 모델을 시험에 적용한다 3. 주기적으로 품질 모델을 검토하고 갱신한다.

에서 SG 1은 세 가지의 SP를 가지며, SP들은 각각 2수준, 3수준, 4수준에 할당되어 있다. SG 2는 3수준에 하나의 특정 프랙티스를 가지지만, 4수준에는 두 개의 특정 프랙티스를 가지고 있다.

4.3 MND-TMM 활용 방안 및 사용예상 효과

본 논문에서 제시한 MND-TMM은 국방 소프트웨어 획득에 참여하는 개발 조직과 사업관리 조직에 의해 다음과 같이 사용될 수 있다.

- 개발 조직은 CMMI 등의 성숙도 모델을 보완하여 시험 분야의 개선을 위한 보조적 모델로 사용할 수 있다.
- 개발 조직은 조직이 가진 시험 능력을 파악하기 위한 평가 기준으로 사용할 수 있다.
- 개발 조직 또는 사업관리 조직이 시험 프로세스 개선을 위한 전략을 수립하기 위해 사용할 수 있다.
- 개발 조직은 사업관리 조직으로부터 조직의 시험 능력을 인증받을 수 있으며, 경쟁자들과의 차별되는 장점을 인정받을 수 있다.
- 사업관리 조직이 정보체계 개발 조직을 선정하기 위한 평가 기준으로 사용할 수 있다.

또한, MND-TMM을 적용하는 경우 다음과 같은 사용예상 효과를 기대할 수 있다.

- 소프트웨어 개발 조직의 시험 프로세스를 개선
- 소프트웨어 시험평가 조직의 시험 프로세스 개선
- 체계적인 시험평가 기법 적용
- 시험환경 개선
- 시험을 거친 제품의 성능과 품질의 향상

5. 결론 및 향후 연구

전장관리정보체계 소프트웨어와 무기체계 내장형 소프트웨어를 포함하는 무기체계 소프트웨어는 무기체계의 성능 발휘에 대부분의 비중을 차지하는 중요한 구성요소이다. 그러나 현재의 국방 소프트웨어 시험은 프로세스의 미성숙에서 오는 문제점들을 가지고 있으며, 이러한 문제들은 국방 소프트웨어 자체의 품질이 미흡한 문제점 뿐 만이 아니라, 전력화의 지연으로 인한 군 전투력의 손실로 나타나기도 한다. 따라서 국방 소프트웨어 시험 프로세스의 개선이 절실히 필요하다.

본 논문에서는 국방 소프트웨어를 개발하는 조직이 가진 시험 프로세스의 성숙도를 평가하고 개선하기 위한 모델인 MND-TMM을 제시하였다. MND-TMM은 5개의 성숙도 수준을 가지며, 4개의 범주 내에 관련 TPA들을 포함하고 있으며 각 수준별 TPA별 목표를 가지고 심사 및 개선활동을 지원한다.

향후 과제는 국방 소프트웨어의 특성을 반영하도록 TPA와 시험 활동들을 정제하고, 각각의 시험 성숙도 수준을 필요로 하는 국방 소프트웨어를 분류하는 것이

다. 또한, 그 효용성을 입증하기 위해 국방 무기체계와 비무기체계를 개발하는 조직에 시험 운용하여 미비점을 보완해 나갈 예정이다.

참고 문헌

- [1] 박휘락, "네트워크 중심전의 이해와 추진 현황", 국방정책연구, 2000년 가을, pp. 156-181, 2000.
- [2] Rick Barbour, "CMMI: The DoD Perspective," SEI Presentation, 2006.10.
- [3] "Guidelines for Successful Acquisition and Management of Software-Intensive Systems Version 3.0," Department of the Air Force Software Technology Support Center, pp. F-7, Appendix F, 2003.5.
- [4] "국방정보체계의 체계적인 S/W 시험평가(검증) 방안 연구", 국방대학원, 1999.12.
- [5] 송영선, "공식 시평단 판정결과 무시한 지상전술 C4I 사업", 국방부/합참 국정감사 보도자료, 2005.9.23.
- [6] "무기체계 내장형 SW 개발관리 실무편람", 방위사업청 사업관리본부, 2006.12.12.
- [7] "국방전력발전업무 규정", 국방부 훈령 제793호, 2006.6.29.
- [8] "무기/비무기체계 내장형 소프트웨어 개발관리 지침", 국방부, 2002.1.1.
- [9] "방위력개선사업관리규정", 방위사업청 훈령 제 13호, 2006.5.1.
- [10] "국방 CBD 방법론", 국방부, 2006.8.
- [11] "소프트웨어 개발 프로세스", 방위사업청, 2006.1.25.
- [12] "국방 소프트웨어 조직 설문/인터뷰 결과", DSRC 기술보고서, 2007.4.7.
- [13] 신현인, 박수현, "모델링 및 시뮬레이션에 의한 시험평가", 한국국방연구원, 국방정책연구 2000년 가을.
- [14] Mark C. Paulk, et al., "Capability Maturity Model for Software Version 1.1," Technical Report CMU/SEI-93-TR-024, 1993.
- [15] "Capability Maturity Model Integration (CMMI), Version 1.1 Staged Representation," CMU/SEI-2002-TR-12, 2002.
- [16] Ilene Burnsteine, "Practical Software Testing," Springer, 2003.
- [17] Tim Koomen, Martin Pol, "Test Process Improvement," Addison-Wesley, 1999.
- [18] Erik van Veenendaal, "Guidelines for Testing - The Test Maturity Model," TMMi-Foundation, <http://www.tmmifoundation.org/downloads/resources/TestMaturityModel.TMMi.pdf>.
- [19] Margaret Kulpa, Kent A. Johnson, "Interpreting the CMMI: A Process Improvement Approach," CRC Press, 2003.
- [20] Jef Jacobs, et al., "The Process of Test Process Improvement," XOOTIC Magazine, November 2000, Vol.8, No.2, pp. 23-29, 2000.11.
- [21] Boris Beizer, "Software Testing Techniques 2nd edition," International Thomson Computer Press, 1990.



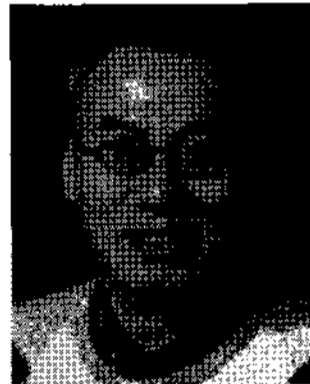
박 준 영

1997년 세종대학교 전산학과 학사. 1999년~2003년 육군 전산장교. 2003년~2005년 육군 C4I 시험평가장교. 2006년~2007년 한국정보통신대학교 공학부 석사. 2007년~현재 육군본부. 관심분야는 소프트웨어 프로세스, 소프트웨어 테

스팅, 품질 모델

류 호 연

정보과학회논문지 : 소프트웨어 및 응용 제 35 권 제 4 호 참조



최 호 진

1982년 서울대학교 컴퓨터공학과 학사. 1985년 영국 뉴캐슬대학교 컴퓨팅학과 석사. 1995년 영국 임페리얼공대 컴퓨팅학과 박사. 1982년~1989년 (주)데이콤 중앙연구소 선임연구원. 1995년~1996년 영국 임페리얼공대 박사후연구원. 1997

년~2002년 한국항공대학교 전자공학부 교수. 2003년~2003년 미국 카네기멜론대학교 방문교수. 2002년~현재 한국정보통신대학교 공학부 교수. 2003년~현재 카네기멜론대학교 겸임교수. 관심분야는 유비쿼터스 컴퓨팅, 인공지능, 소프트웨어 공학, 그리드 컴퓨팅, 데이터 마이닝

백 종 문

정보과학회논문지 : 소프트웨어 및 응용 제 35 권 제 4 호 참조



김 진 수

1999년 전남대학교 컴퓨터공학과 학사. 2002년 KAIST 컴퓨터공학과 석사. 2002년~2005년 한국생명공학연구원 연구원. 2006년~현재 국방과학연구소 연구원. 관심분야는 인공지능, 보안, 데이터 통신, 데이터베이스, 패턴 인식, 분산 시스템