

테스트 성숙도 모델 기반 시스템 평가 기술

한국정보통신대학교 | 류호연* · 백종문** · 이단형**

서울여자대학교 | 이병걸**

1. 서론

국방 도메인에서의 전장 환경 및 체계가 변화함에 따라 소프트웨어의 비중이 증대되고 있으며[1], 해당 소프트웨어의 품질이 하드웨어의 품질 및 체계의 품질에까지 영향을 미치고 있다. 즉, 컴퓨터가 설치된 무기체계들은 소프트웨어에 의해 성능이 결정된다고 할 수 있기 때문에, 국방 도메인에서 소프트웨어는 그 중요성이 널리 인식되어 있으며 고품질의 소프트웨어를 얻기 위한 노력이 진행되고 있다. 소프트웨어는 하드웨어 단독으로는 제공할 수 없는 기능들을 전투원에게 제공한다. 작가는 탄약의 신관을 제어하는 기능에서부터, 크게는 데이터를 수집하여 가공하고 전송·교환하여 체계들을 연결하는 역할까지 소프트웨어가 수행하기 때문이다[2].

국방 소프트웨어의 품질을 향상시키기 위한 노력 중에서도 개발 비용의 40~50%의 비용을 필요로 하는[3] 소프트웨어 시험 부분을 개선하는 것이 중요한 것으로 인식되고 있다. 개발 과정에서 체계적인 시험을 통해 오류를 미리 제거한다면 신뢰성 있는 고품질의 소프트웨어 체계가 개발될 수 있기 때문이다. 그러나 지금까지는 국방 소프트웨어에 대한 체계적인 접근 방법이 존재하지 않으며 무기체계에 내장된 소프트웨어들은 하드웨어 성능이 군의 요구사항을 만족하는 지를 확인한 다음에 부차적으로 확인하는 정도의 시험만을 수행하며, 정보체계들도 정보기술 분야에 정통한 시험인력을 갖추지 못한 한시적인 시험조직에 의한 제한적인 시험만을 수행되고 있는 실정이다[4]. 그 결과로 소프트웨어 개발 과정에서 발견되지 못한 결함이 운용과정에서 나타나는 경우가 많았으며, 결함들을 수정하고 개선하는 동안에 정상적인 운용이 불가능하여 군 전투력의 저하가 나타났다[5]. 따라서 국방 소프트웨어에서의 이

러한 문제점을 해결하기 위해서는 국방 소프트웨어를 획득하는 과정에서 개발조직의 시험수준을 향상시켜 소프트웨어의 품질을 향상시키기 위한 방법이 필요하다.

현재 CMM, CMMI, SPICE와 같은 소프트웨어 프로세스 성숙도 모델들과 TMM, TPI, TOM, TOM 등과 같은 테스트 성숙도 모델을 적용함으로써 소프트웨어 개발 프로세스와 테스트 프로세스의 지속적인 개선과 품질 향상을 도모하고 있다. 하지만 기존의 프로세스 성숙도 모델들과 테스트 성숙도 모델들은 외국 사례를 기준으로 제작이 되었고 전반적인 시험에 관한 활동들을 포함하고 있지 않으며, 심사방법 또한 체계적이고 구체적이지 못하기 때문에, 국내, 특히 국방 소프트웨어의 테스트 능력을 객관적으로 평가하고 효율적으로 적용하기에는 한계가 있다. 따라서 국방 소프트웨어 개발 조직의 시험 수준을 평가하고 국방 소프트웨어 시험 프로세스를 개선을 통하여 품질을 보장하기 위한 국방 시험 성숙도 모델(MND-TMM; Ministry of Defense-Testing Maturity Model)과 그에 적합한 심사 모델을 개발하여 적용하는 것이 필요하다.

본 논문의 구성은 다음과 같다. 2장에서는 국방 소프트웨어를 소개하고 국방 소프트웨어 시험에서의 문제점을 분석한다. 3장에서는 기존의 성숙도 모델들과 심사 모델에 대해 살펴보고, 그 한계점을 기술한다. 4장은 국방 도메인에서의 시험 프로세스 개선을 통한 국방 소프트웨어의 품질 향상을 위한 국방 시험 성숙도 모델과 그 심사 모델을 소개한다. 마지막으로 5장에서는 결론과 향후 연구사항에 대해 기술한다.

2. 국방 소프트웨어의 분류와 시험

국방 소프트웨어는 크게 무기체계와 비무기체계 소프트웨어로 구분된다. 무기체계 소프트웨어는 전장관리정보체계 소프트웨어와 무기체계 내장형 소프트웨어로 분류된다. 표 1은 국방 소프트웨어의 자세한 분류를 보여준다. 무기체계 내장형 소프트웨어는 전자

* 정회원

** 종신회원

표 1 국방 소프트웨어의 분류[6]

무기체계 소프트웨어	무기체계 내장형 소프트웨어	통전장비, 총포탄약, 기동장비, 항공기/함정 등
	전장관리정보체계 소프트웨어	지휘통제·통신무기체계, 군사정보지원체계, 국방모의훈련체계
비무기체계 소프트웨어	자원관리정보체계 소프트웨어	국방정보관리지원체계, 사무자동화체계
	기반체계 소프트웨어	정보통신망, 컴퓨터체계, 정보보호체계, 상호운용성체계

의 화력통제체계, 항공기의 항법장치 등과 같이 각종 무기체계에 내장되어 해당 장비의 임무에 전용으로 사용되는 소프트웨어를 말한다[7]. 전장관리정보체계는 국방 업무와 관련된 정보를 수집·가공·전달·전시하는 기능들을 수행하는 컴퓨터 하드웨어와 소프트웨어, 데이터 및 통신 수단의 조합이며[8], 전장관리정보체계 소프트웨어는 전시 또는 평시에 지휘관 및 참모에게 필요한 정보를 제공하며, 작전에 중요한 결정을 내리는 것을 보조하는 소프트웨어를 일컫는다. 비무기체계 소프트웨어는 전투상황과 관련 없이 주로 군의 일상적인 업무에 사용되는 소프트웨어로서 자원관리 소프트웨어와 기반체계 소프트웨어로 구분된다.

2.1 국방 소프트웨어의 특징

국방 소프트웨어는 상용제품 또는 관급제품에 비하여 엄격한 요구사항 또는 품질기준을 적용받으며 다음과 같은 국방 정보체계[9]의 특징에 기인한다.

- 1) 임무 최우선 체계 : 전투지휘를 지원하는 국방 정보체계의 오류는 다수의 전투원에게 생명에 직접적인 영향을 미친다. 이것은 금전적 피해보다 심각하다.
- 2) 체계들의 체계 : 국방 정보체계는 단독으로 구성되지 않고, 다수의 부체계들이 전체 체계를 구성한다. 부체계들은 전체 체계의 기능을 발휘하기 위해 데이터를 생산/교환한다.
- 3) 소프트웨어 집약 체계 : 국방 정보체계는 소프트웨어가 체계의 기능, 비용, 개발 위험, 개발 기간 등에 대부분의 비중을 차지하는 체계이다.
- 4) 하드웨어와 소프트웨어의 복합 체계 : 국방 정보체계는 군 환경에 적합하게 제작된 하드웨어에 기반으로 하는 기능을 발휘하며, 민간 분야의 정보체계와는 운용환경에 큰 차이가 있다.
- 5) 상호운용성 : 국방 정보체계는 부체계간의 데이터 교환이 불가능하다면 전체 체계의 기능을 발휘하지 못한다. 상호운용성은 체계, 부대, 군별, 국가 등 다양한 관점에서 고려되어야 한다.

2.2 국방 소프트웨어 시험

국방 소프트웨어는 획득과정에서 탐색개발, 개발, 시험평가의 과정을 거친다. 그러나 획득과정에서 시험이 체계적으로 수행되지 않는 문제점이 있다. 군은 연구개발을 거친 무기체계에 대한 시험평가를 실시한다. 시험평가는 획득절차의 최종 관문으로서 무기체계의 성능, 기술, 품질, 운용관리적 측면에서의 요구사항 및 개발목표의 충족여부와 운용적합성, 효율성, 안전성 등을 확인 검증하는 절차이다[9]. 그러나 국방 관련 규정/지침[8,9] 등에 무기체계 위주로 시험절차는 잘 정의되어 있지만 소프트웨어 시험에 대한 기술은 미흡한 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 방법론인 ADDMe(Advanced Defense component Development Methodology)[10]와 방위사업청에서 개발한 국방 소프트웨어 개발 프로세스[11] 등에도 소프트웨어 시험의 구체적인 방법이나 활동들이 포함되어 있지 않다. 따라서 유닛 시험, 통합시험, 체계시험 등에서 어떠한 시험 활동들을 수행하는 지는 개발 조직의 역량에 달려 있는 상태이며 산출물 중심으로 관리되기 때문에 실질적인 테스트 프로세스 개선을 위한 노력이 이루어지고 있지 않은 실정이다. 또한, 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 전

표 2 국방 정보체계 시험평가 사례[4]

구분	문제점
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

문 인력이 부족하고, 시험평가의 독립성이 보장되어 있지 않았으며, 사업관리가 효과적으로 이루어지지 않은 사례들이 나타나 있다.

2.3 국방 소프트웨어 시험의 문제점 분석

본 연구과제에서는 보다 정확한 국방 소프트웨어 개발 및 시험에서의 문제점을 파악하기 위하여 국방 조직 및 방산업체를 대상으로 설문과 인터뷰를 수행하였다. 그림 1은 국방 소프트웨어의 시험이 시험 기간 및 시험 인력의 부족으로 인하여 효율적으로 수행되고 있지 못하다는 결과를 보여주고 있으며, 그림 2의 결과는 시험 조직의 전문성 결여와 시험에 투입되는 비용과 할애되는 시험 기간의 부족으로 인하여 대상 국방 소프트웨어에 적합한 시험 기법이 활용되고 있지 못함을 보여주고 있다.

많은 국방 관련 소프트웨어 개발 조직들이 소프트웨어 시험과 관련하여 제기하는 문제점들은 다음과 같다[12].

- 1) 기능 위주의 시험 : 소프트웨어에 대한 시험이 단순한 기능 위주로 진행되어 품질특성에 대한 고려가 부족하다.
- 2) 운용시험에서의 결함 발생 : 개발시험까지 발견되지 않았던 문제들이 운용시험 단계에서 다수 발견됨으로써 이를 해결하기위해 실제 운용이 지연되는 경우가 있다. 시험단계에서 실제 운용환경과 유사한 시험환경 제공이 미흡하다.

- 3) 시험 인력의 부족과 교육의 미흡 : 개발/시험 조직이 전문적인 지식을 가진 시험인력을 가지지 못하고 있으며 체계적인 시험에 관련된 교육을 제공하고 있지 않다.
- 4) 독립된 시험 조직/프로세스의 부재 : 개발 조직에서 시험의 독립성을 보장하기위해 분리된 시험 조직을 갖추지 않아서, 객관적이고 독립적인 시험이 미흡하다.
- 5) 품질 특성 고려가 미흡함 : 요구사항 단계에서 품질 요소에 대한 요구가 식별되지 않기 때문에 시험에도 품질 요소가 반영되지 않는다.
- 6) 운용시험 환경 구축이 어려움 : 운용시험을 위한 데이터 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

3.1 프로세스 성숙도 모델

조직이 가진 소프트웨어 프로세스의 성숙도를 평가하고 개발 프로세스개선을 위한 대표적인 모델로 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 기존에 사용되어왔던 SW-CMM, SE-CMM, SA-CMM, IPD-CMM 등의 모델들이 갖는 중복성으로 인해 소프트웨어 개발조직이 프로세스 개선을 위한 중복투자의 부담을 없애기 위해 통합된 성숙도 모델로서 CMM과 유사한 단계적 표현과 프로세스 영역별로 6개의 능력 단계를 가지는 연속적 표현을 제공한다.

프로세스 성숙도 모델들은 소프트웨어 개발의 전 과정에 대해 기술하고 있음에도 불구하고 전체 프로젝트 비용의 40~50%에 달하는 시험 분야가 차지하는 비중은 매우 낮다[16]. 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 CMM 보다 시험에 대해 많은 내용을 담고 있으나, 소규모의 개발조직의 시험 프로세스에 적용하기에는 부담이 크다는 단점이 있다[17].

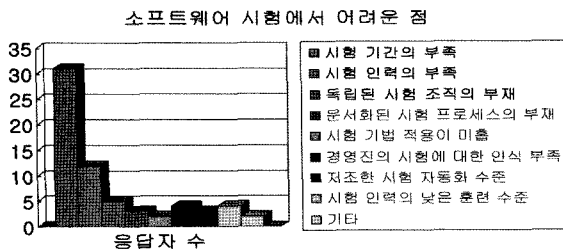


그림 1 국방 소프트웨어 시험에서의 어려운 점

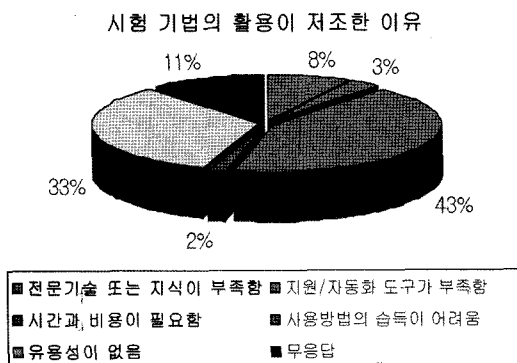


그림 2 다양한 시험 기법의 활용이 저조한 이유

3.2 시험 성숙도 모델

시험 성숙도 모델은 CMM 등과 유사한 프로세스 개선 모델로서 지금까지 여러 형태의 시험 성숙도 모델, TMM, TPI, TIM, TOM 등이 시험 프로세스의 성숙도를 평가하고 개선하기 위한 참조 모델로 개발되어 왔다.

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[18]. 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.

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. 그러나 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 중요 영역을 다루고 있지 않다[19]. 마지막으로 TMM은 단계적 표현을 사용하기 때문에 조직 전체의 프로세스 수준만 나타나고, 개선이 필요한 프로세스 영역별 수준이 나타나지 않는다.



그림 3 TMM 내부 구조[18]

3.2.2 Test Process Improvement(TPI)

TPI는 Kooman과 Pol에 의해 1997년에 개발된 시험 성숙도 모델로서 CMMI와 유사한 연속적 표현을 제공한다[20]. TPI는 3개의 기본 성숙도 수준과 14 단계의 스케일을 가지며, 시험 성숙도 매트릭스에 핵심 영역별 성숙도 수준을 정의한다. 성숙도 수준에 필요한 요구사항들을 담은 체크포인트와 프로세스 개선을 위한 팁과 아이디어를 담은 개선 제안사항이 추가로 제공된다. 그림 4는 TPI의 프레임워크를 보여준다. 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 3은 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. 그러나 모든 핵심영역에 공통적으로 적용되는 성숙도 수준의 목적이 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.

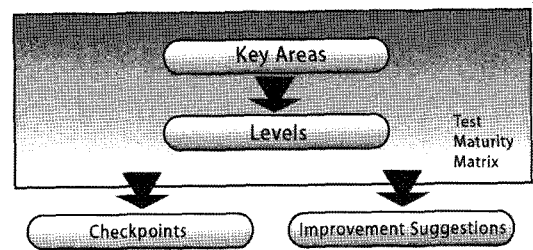


그림 5 TPI 프레임워크[20]

		Scale	0	1	2	3	4	...	13
20 Key Areas	Key Area								
	Test strategy		A						
	Life Cycle		A		B				
	Reporting		A			B			
	etc.								

그림 6 시험 성숙도 매트릭스[20]

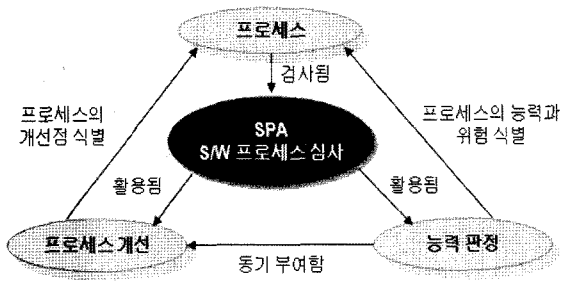


그림 7 소프트웨어 프로세스 심사의 목적과 활용

3.3 심사모델

국제 표준화 기구인 ISO에서는 소프트웨어 프로세스 심사를 ‘프로세스가 품질, 비용, 일정 목표를 달성하는 능력을 결정하기 위하여 기준을 가지고 조직에서 사용하는 프로세스를 검사하는 것’으로 정의하고 있다[5]. SEI(Software Engineering Institute)에서는 이를 ‘조직의 현행 소프트웨어 프로세스의 상태를 결정하고, 조직이 갖고 있는 높은 우선 순위의 프로세스 관련 문제점을 결정하고, 프로세스 개선에 대한 조직적인 지원을 얻기 위하여, 훈련된 소프트웨어 전문가에 의해 행해지는 심사(Appraisal)’라고 정의하고 있다[22]. ISO 정의가 심사의 목적을 조직의 프로세스 능력을 결정하는데 중점을 두고 있는 반면에 SEI에서는 프로세스의 현 상태와 문제점을 식별하고 개선점을 찾고자하는데 중점을 두고 있음을 알 수 있다. 그림 7은 ISO 문서에 나타난 소프트웨어 프로세스 심사의 목적과 활용을 도식화 하고 있다[22].

소프트웨어 프로세스 심사를 위해서는 먼저 심사 참조 모델과 심사 방법이 정의되어 있어야 한다. 심사 참조 모델은 기본적으로 심사의 대상이 되는 프로세스가 무엇이 될 수 있는가를 정의하여야 하고(프로세스 목표로 대상이 되는 프로세스가 무엇인지를 정의함), 각 프로세스에 대하여 능력이 어느 정도인지를 표시할 수 있어야 한다. 또 심사를 하는 과정에서 명확한 기준을 제공하기 위하여 프로세스가 수행되거나 그 능력이 어느 정도인지를 판별하기 위한 지표가 정의되어 있어야 한다.

프로세스 심사의 방법은 모든 프로세스 영역에 대해 공통적으로 적용될 수 있다. 기존 테스트 성숙도 모델의 경우에는 심사 모델에 대한 언급이 부족하므로 가장 널리 사용되고 있는 ISO 국제 표준인 SPICE와 해외 시장 표준인 CMMI의 심사방법에 대한 비교 분석을 통해 테스트 프로세스 심사 모델의 기본 프레임워크를 구성한다.

3.3.1 CMMI 기반 심사 모델

CMMI 기반 심사 방법인 Standard CMMI Appraisal

Method for Process Improvement(이하 SCAMPI) A는 현재 SEI(Software Engineering Institute)로부터 유일하게 승인된 클래스 A 심사 방법이다[22]. SEI에서는 Appraisal Requirement for CMMI(이하 ARC) 문서를 통해 CMMI 기반 프로세스 심사 방법이 만족해야 할 요구사항을 정의하고 있으며, 심사 방법의 엄격 정도에 따라 만족해야 할 요구사항의 범위를 다르게 함으로써 클래스 A, B, C의 3가지 유형의 심사 방법으로 구분한다. 그 중 SCAMPI A는 가장 엄격한 수준의 심사 방법인 클래스 A의 심사 유형에 속한다.

SCAMPI A에 의한 심사결과는 프로세스 개선을 위해 활용되는 것이 보통이며, 획득 및 공급업체 선정 또는 공급업체에서 수행하는 프로세스 모니터링(계약 관리)을 목적으로 하는 심사 활동에 활용할 수 있다. SCAMPI A는 CMMI의 단계적 표현 방법이나 연속적 표현 방법 모두를 기준으로 심사할 수 있다. SCAMPI A는 CMMI에서 정의하고 있는 구성요소들을 다음의 3가지로 구분해서 심사에 적용하도록 권고한다. 1) 필수 요건: 공통 목표, 고유 목표 2) 권고 요건: 공통 프랙티스, 고유 프랙티스 3) 선택 요건: 하위 프랙티스, 상세 설명, 산출물 예시 등

SCAMPI A 심사의 판단 기준은 특정 프로세스 활동에 대한 목표 달성 여부이다. 따라서 심사 팀은 목표와 관련된 프랙티스들을 조사하고, 하위 프랙티스, 상세 설명, 산출물 예시 등을 통해 CMMI의 프랙티스 구현에 대한 구체성을 보완함으로써 심사 대상 조직이나 프로젝트가 공통 목표와 고유 목표를 만족하고 있는지에 대한 근거를 찾는다. SCAMPI A는 이러한 검증 활동을 지원하기 위해 인터뷰와 문서 검토, 설문서 분석 활동들을 포함하고 있다. 최종 심사 결과에는 조직의 성숙도나 프로세스의 능력도, 조직의 강·약점 프로세스, 그리고 개선 계획들이 포함된다.

3.3.2 SPICE 기반 심사 모델

SPICE(ISO/IEC 15504)는 프로세스 개선을 위한 조직 내의 프로세스 상태 파악, 조직의 특정 요구사항을 만족시키기 위한 조직 내의 프로세스 적합성 결정, 계약 관리를 위한 공급자의 프로세스 적합성 결정을 목적으로 사용할 수 있다. SPICE 심사 모형의 기본 구조는 SPICE-Part 2의 참조 모형의 구조와 동일하다. SPICE 심사 모형과 참조 모형의 프로세스 범주, 프로세스, 목적 기술문, 프로세스 능력 수준, 프로세스 속성은 서로 1:1로 사용된다[23].

SPICE 심사 모형은 프로세스 속성 달성을 통해 프로세스의 능력을 심사할 수 있다는 원칙에 근거하고 있다. 따라서 프로세스 차원에서의 각 프로세스는 기본 수

행활동을 가지고 있으며, 그 활동들의 수행을 통하여 프로세스 목적의 달성 정도에 대한 척도를 제공한다. 이와 유사하게, 능력 차원의 각 프로세스 속성은 관련된 관리 수행활동을 가지고 있으며, 그 활동의 수행을 통하여 구현된 프로세스 속성의 달성 정도를 나타내는 척도를 제공한다. SPICE 심사 모형에서 정의된 지표들은 프로세스를 사례화시켰을 때 발견될 수 있는 객관적인 증거가 되며, 그 프로세스 능력의 성과를 판단하는데 사용될 수 있다.

4. 국방 시험 성숙도 모델 : MND-TMM

본 장에서는 MND-TMM의 개발 배경 및 설계된 기본 구조를 설명하고 그 활용방안을 제시한다.

4.1 MND-TMM 개요

MND-TMM은 국방 소프트웨어 시험 프로세스를 개선시키고 체계적이고 효과적인 시험을 거친 소프트웨어의 품질을 향상시키기 위한 성숙도 모델이다. 국방도메인에 적합한 모델을 개발하기 위하여 그림 8에 나타난 것과 같이 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, LISI, DO-178B 등을 조사 분석하여 국방 소프트웨어 개발 절차를 분석하고 해당 절차상에서의 시험기법 및 활동에 대해 조사하고 신뢰성 확보 기법을 파악하였다.
- 성숙도 모델: CMM, CMMI와 같은 프로세스 성숙도 모델 및 TMM, TPI 등과 같은 테스트 성숙도 모델을 분석하여 테스트 관련 프로세스 영역을 추출하고, MND-TMM의 시험 프로세스 영역 정의에 활용하였다.

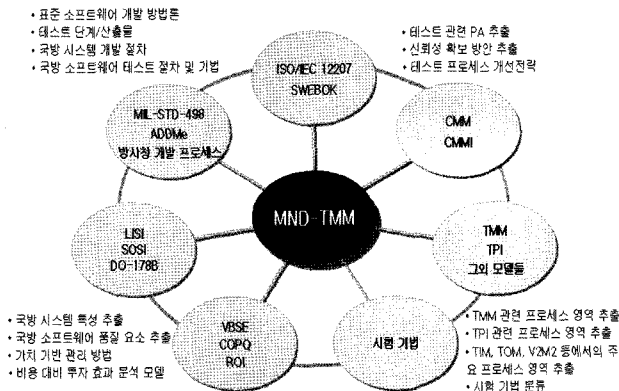


그림 8 MND-TMM 개발 배경

- 관련 핵심 요소 기술 연구 : 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 4개의 범주로 구분된 시험 프로세스 영역(TPA; Testing Process Area)들을 가진다. 각각의 TPA는 성숙도 수준을 결정하는 시험 활동들을 포함한다. 그림 5는 MND-TMM의 구조를 나타낸 것이다.

4.2.1 성숙도 수준

MND-TMM은 모든 프로세스 영역에 적용되는 5 단계의 성숙도 수준을 가진다. 성숙도 수준과 해당 성숙도 수준에서 달성해야 하는 일반 목표는 다음과 같다.

- 1 수준 - “인식”: 조직은 시험의 필요성을 인식한다. 하지만, 시험 프로세스가 정립되지 않았으며, 시험 활동들이 임의적으로 이루어진다.
- 2 수준 - “이해”: 개별 프로젝트에 시험 프로세스가 사용된다. 시험 프로세스에 따라 기본적인 시험 활동이 이루어진다.
- 3 수준 - “정의”: 조직의 표준 시험 프로세스가 정의되어 있다. 각각의 프로젝트 팀은 조직의 표준 시험 프로세스를 테일러링 지침에 따라 변경하여 사용한다.
- 4 수준 - “정량화”: 시험 프로세스는 다양한 수준의 메트릭에 따라 정량적으로 평가, 관리된다.
- 5 수준 - “최적화”: 시험 프로세스 개선이 이루어진다. 시험 프로세스는 결함 예방을 중점으로 지속적으로 시험 프로세스가 개선된다.

4.2.2 시험 프로세스 영역

MND-TMM은 4개의 범주로 분류되는 15개의 TPA들을 가진다. TPA들은 TPI의 20개 핵심 영역 중에서 선정되었으며, 상호운용성, 그 외, ADDMe, 방위사업청 개발 프로세스, 그 외 관련 표준등에서의 국방 소프트웨어 특성이 포함된다. 국방 소프트웨어의 특징을 반영하는 TPA가 추가되었다. 각각의 TPA들은 성숙도 수준 별로 달성해야 하는 일반 목표와 특수 목표를 가진다. 범주에 따른 TPA의 분류는 다음과 같다.

- 프로세스(Process): 시험 프로세스와 관련된 TPA들을 포함한다. 이 범주는 시험 전략, 수명주기 모델, 프로세스 관리 TPA를 가진다.

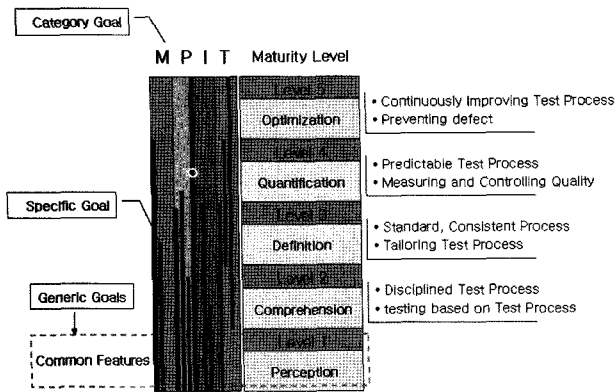


그림 9 MND-TMM의 성숙도 수준 정의

- 기법(Technique): 시험에 사용하는 기법들과 관련된 TPA들을 포함한다. 시험 명세, 검사, 기법 적용, 품질 통제, 평가가 포함된다.
- 기반구조(Infrastructure): 테스트 수행을 위해 필요한 제반 환경과 관련된 TPA들을 포함한다. 시험 환경, 테스트웨어 관리, 시험 추정, 시험 조직, 시험 도구가 해당된다.
- 군(Military): 민간 영역과 다른 국방 분야의 특성과 관련된 TPA들을 포함한다. 상호운용성, 그 외, ADDMe, 방위사업청 개발 프로세스, 그 외 관련 표준등에서의 테스트 관련 활동, 산출물, 품질 특성들이 고려된다.

특히, 군만의 특성을 반영한 TPA를 위해 국방 소프트웨어의 특성을 알아보기 위한 조사[12]에서는 긴 사업기간, 정형적 개발 프로세스, 시험의 강조, 요구사항의 변동, 시험환경 구축의 어려움 등이 국방 소프트웨어의 특성으로는 식별되었는데, MND-TMM은 이와 같은 특성을 시험 프로세스에 반영할 수 있도록 군 범주에 국방 소프트웨어 특성을 시험 프로세스 영역으로 포함하고 있다.

4.3 MND-TMM 활용 방안

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.
- 사업관리 조직이 정보체계 개발 조직을 선정하기 위한 평가 기준으로 사용할 수 있다.

· 개발 조직은 사업관리 조직으로부터 조직의 시험 능력을 인증 받을 수 있으며, 경쟁자들과의 차별되는 장점을 인정받을 수 있다.

4.4 MND-TMM의 심사모델

4.4.1 심사 방법의 목표 및 개선 방안

(1) 심사 간소화

심사 결과의 정확성을 위해서는 심사 시 심사 대상 조직에서 충분하고 정확한 심사 자료를 제공하는 것이 중요하다. 심사 자료는 주로 문서의 형태로 제공하며, 인터뷰나 설문 작성을 통해서도 자료를 제공할 수 있다. 심사 대상 조직에서는 심사를 받는 동안 자기 본연의 업무 이외에 심사 자료 준비와 제공을 위한 업무도 맡아야 하는 부담을 갖는다. 중소기업의 소프트웨어 개발 업체가 주를 이루는 국내 현실에서 심사 기간 동안 발생하는 이중 업무는 심사를 기피하게 만드는 원인이 된다. 중소기업의 소프트웨어 개발 업체가 심사비용과 심사 준비 업무의 어려움으로 인해 프로세스 심사를 통한 개선점 파악의 기회를 얻지 못한다면 국내 소프트웨어 산업계의 역량 증대에도 큰 손실을 준다. 따라서 본 연구에서 제안하는 심사 방법은 심사 자료 준비에 필요한 일정과 대상 자료를 최소화할 수 있는 방안으로 다음의 내용을 적용한다.

① 일정: 문서 확인 및 검증 활동의 비중 조절

- 설문서 심사를 통한 문서 확인 절차 간소화
- 세부 심사 활동 중 각종 검토 활동의 통폐합
- 심사원에 대한 교육 활동 생략
- 내부 심사 시 문서 확인 및 검증 활동의 옵션화

② 대상 자료: 심사 자료 준비를 위한 문서 작업 조절

- 기본적으로 직접 산출물만 심사하는 것을 원칙으로 한다. 단, 문서 테이블과 비교하여 일관성이 결여된 부분은 직, 간접 산출물을 모두 심사한다.

(2) 심사의 객관성 및 정확성 확보

일반적으로 프로세스 심사는 심사 대상 조직에서 제공한 문서들을 바탕으로 프로세스 영역 수행 여부를 판단하게 된다. 이러한 문서 심사는 문서의 종류가 다양하고 그 수가 많기 때문에, 심사에 소요되는 시간 중 가장 많은 시간이 소요되는 부분이다. 심사를 간소화하기 위해, (1)번 항목에서 문서 확인 절차를 간소화하고, 심사 대상 자료의 범위를 직접 산출물로 제한하였으나, 이 때문에 심사의 정확성이 결여될 수 있는 문제점이 있다. 또한 기존 CMMI나 SPICE 심사의 가장 큰 문제점으로 지적되는 것이 심사원의 주관적 판단이 많이 개입된다는 것이다. 국내의 심사 희망 조직 중 대부분은 레벨 획득을 목적으로 심사를 받

는다. 이러한 잘못된 목적 달성을 위해 일부 조직에서는 내재화된 프로세스를 통해 산출되는 문서가 아닌, 심사를 위한 문서를 별도로 작성하여 심사를 준비하는 사례가 있기도 하다. 하지만 증거(직, 간접 산출물)의 유무에 따라 프로세스 수행 여부를 판단하는 방식의 기존 심사 방법들은 이런 점을 발견하는데 심사원의 주관적인 판단이 개입될 수밖에 없다. 심사 모델이 증거의 진위 여부를 판단하기 위한 객관적인 기준을 제시하지 않기 때문이다. 심사를 받은 조직은 목표했던 레벨을 획득하더라도 정확하지 않은 심사가 이루어졌기 때문에 제대로 된 강, 약점을 발견하기 어렵고, 결국 기대했던 프로세스 개선의 효과를 얻지 못하는 결과를 낳게 된다. 따라서 본 연구에서 제안하는 심사 방법은 이런 문제점을 해결하기 위해 다음과 같은 두 가지 방법을 제안한다.

- ① 문서 활용 기록 검토 : 중간 산출물들에 대한 활용 기록을 확인함으로써, 심사를 위해 만들어진 문서가 아니라 조직에서 실제로 프로세스 활동을 하면서 작성된 문서인지(진위 여부)를 파악할 수 있다. 문서 활용 기록에는 그 문서를 활용한 시기, 장소, 이유, 방법 등이 포함될 수 있다. 예를 들어, 요구사항정의서 문서에 대한 심사를 하는 경우, 설 문서를 통해 문서의 존재 여부를 파악한 후, 심사원의 판단에 의해 주요산출물로 결정되면 요구사항 문서의 변경기록이나 이 문서가 다른 활동에서 참고 되거나 이용된 기록 등을 추가적으로 확인하도록 한다.
- ② 문서 테이블 적용 : 심사가 객관적이기 위해서는 심사 자료를 심사하고, 심사 결과를 도출하고, 등급을 결정하는 등 심사원이 의사결정을 할 때 객관적인 판단 기준을 근거로 해야 한다. 그런 면에서 볼 때, 기존의 심사 방법들은 심사원들이 심사 자료를 심사하고, 자료의 충분성 정도를 결정하는 것이 심사원의 판단에만 맡겨지기 때문에, 자칫 심사원의 경험과 지식, 기술에 따라 그 판단기준이 달라질 수 있는 위험성이 있다. 따라서 본 연구에서 제안하는 방법을 이를 해결하기 위해 문서 테이블을 적용하여 객관적인 판단 기준을 제시한다.

산출물은 한번 만들어지면 어떤 다른 활동에서 적어도 한번 이상은 반드시 사용되게 돼있다. 프로젝트 수행 중에 만들어진 문서가 그 이후에 한 번도 사용되지 않는다면, 그 문서는 작성할 필요가 없는 문서인 것이다. 심사 시에 대상 조직의 개발 프로세스를 파악하고, 각 활동 별 입력물과 출력물을 파악하여 그 관

계를 문서 테이블로 정의한 후, 이를 근거로 일관성이 없는 부분을 선별해 내도록 한다. 예를 들어, 어떤 활동의 출력물이 다른 활동의 입력물로 이용되지 않거나 어떤 활동의 입, 출력물이 실제로는 존재하지 않는 등 일관성이 없는 부분을 발견하도록 한다. 이 문서 테이블은 증거의 존재 여부(누락된 문서)를 판단하기 위한 기준으로도 사용할 수 있다.

4.4.2 심사 참조 모델

본 심사 방법에서 참조모델로 이용하는 테스트 성숙도 모델은 다음 그림 9와 같이, 전체적으로 CMMI와 유사한 구조를 갖는다. 전체적으로 5개 수준의 레벨로 구성되며, 각 레벨은 4개의 범주 별 SG와 GG로 구성된다. 4개의 범주는 각각 테스트 인프라, 테스트 기법, 테스트 프로세스, 도메인 특성(예, 국방, 의료 산업, 금융 등등)을 의미하며, 수준 별 Goal이 포함된다. 이렇듯 영역별로 나누어 Goal을 제시하는 것은 CMMI와 차별화된 부분이라 할 수 있다. 심사의 핵심은 CMMI 기반 심사와 마찬가지로 Goal 달성여부에 있게 된다.

4.4.3 심사 방법 절차

본 연구에서 제안하는 심사 방법은 크게 3단계로 구분되며 그 절차는 그림 10과 같다. 심사의 대상 프로세스는 테스트 프로세스로 한정된다.

각 단계는 여러 프로세스로 구성되며, 하나의 프로세스는 여러 활동들을 포함한다. 그림과 같이 각 단계를 구성하는 활동들은 기존의 심사 모델인 SCAMPI와 SPICE에서 공통적으로 수행하는 활동들을 바탕으로 선정하였다. 각 활동에 대한 세부적인 내용은 앞에서 파악된 요소들을 적용한 수행 기법에 대한 설명을 포함하여, 활동의 목적, 착수기준, 완료기준, 입력물, 출력물, 수행 절차 등으로 구성된다.

각 단계의 활동은 기존 심사 모델인 CBA-IPI와 SPICE, SCAMPI에서의 공통 활동들을 기반으로 하되, 심사 현황 분석을 통해 파악한 요소들을 보완하여 구성한다. 먼저 심사의 객관성 확보를 위해 문서테이블과 문서 활용 기록을 검토하는 방법을 적용함으로써 심사원에게 객관적인 판단 기준을 제시한다. 그리고 세부 심사

1. 평가 계획 및 준비	2. 문서 평가	3. 현장 평가
평가 요구사항 분석	문서 평가 착수	현장 평가 착수
계획서 작성	문서 평가 수행	현장 평가 수행
평가팀 준비	문서 평가 결과 작성	현장 평가 결과 작성
문서 평가 준비	현장 평가 준비	평가결과 보고 준비

그림 10 테스트 프로세스 심사 방법의 3단계

활동 중 각종 검토 활동을 통폐합하고, 문서 확인 시 설문서를 이용한 심사를 통해 문서 확인 절차를 간소화한다. 또한 문서 심사 수행 시, 기본적으로 직접 산출물만을 심사 하되, 문서 테이블을 기준으로 일관성이 결여된 부분에 대해서는 간접 산출물까지도 심사하여, 심사의 기간을 단축하면서도 객관적인 심사가 이루어질 수 있도록 한다. 이러한 요소들을 적용하여 구성한 각 단계의 활동은 다음과 같다.

- (1) 1단계: 심사 계획 및 준비 - 심사를 수행하기 위한 모든 준비를 완료하는 단계로, 심사 요구사항 분석, 계획서 작성, 심사 팀 준비, 참여자 준비, 문서 심사 준비 활동을 수행한다. 이 단계에서는 심사 일정을 단축하기 위한 방법으로 문서 목록작성과 설문 조사 방법을 적용한다. 문서 목록작성 및 설문 조사를 통해 문서의 존재 여부를 파악할 수 있고, 이 결과를 이용해 1차적인 판단이 가능하다.
- (2) 2단계: 문서 심사 - 문서 검증과 확인을 중심으로 심사하는 단계로, 문서 심사 착수, 문서 심사 수행, 문서 심사 결과 작성, 현장 심사 준비 활동을 수행한다. 이 단계에서는 심사 일정을 단축하고, 심사 대상 자료를 최소화하기 위해 문서 확인 및 검증의 비중을 조절한다. 심사 팀은 1단계에서 파악된 문서 테이블과 문서 활용기록을 바탕으로 중점을 두어 확인하고 검증해야 할 문서들을 정할 수 있고, 심사 팀이 판단할 때 중요도가 낮은 문서의 경우 확인과 검증 절차를 간소화 할 수 있다.
- (3) 3단계 : 현장 심사 - 문서 심사 결과 추가 심사가 필요하다고 판단되는 부분을 중심으로 인터뷰와 추가 자료 조사를 통해 심사를 수행하고, 최종적인 심사 결과를 생성하는 단계로, 현장 심사 착수, 현장심사 수행, 현장 심사 결과 작성, 결과 보고 준비 활동을 수행한다. 심사 일정을 단축하고자 하는 개념이 전체적인 단계의 정의에 적용되었다.

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 제시하였으며, 또한 이를 위한 테스트 프로세스 심사 모델은 CMMI나 SPICE 등의 기존 심사 모델의 단점을 국내 실정을 반영하여 보완하고 간소화시켜 테스트 프로세스 성숙도를 심사할 수 있도록 구성하였다.

향후 과제는 국방 소프트웨어의 특성을 반영하도록 TPA와 시험 활동들을 정제하고, 각각의 시험 성숙도 수준을 필요로 하는 국방 소프트웨어를 분류하는 것이다. 또한, 그 효용성을 입증하기위해 국방 무기체계와 비무기체계를 개발하는 조직에 시험 운용하여 미비점을 보완해 나갈 예정이다. 또한 제안하는 심사 모델을 적용한 심사 사례를 구축하여 모델의 실효성을 검증하고, 계속적으로 심사 모델을 개선해 나갈 계획이다.

후기

본 논문은 국방 소프트웨어설계 특화연구센터의 연구과제로 수행중인 테스트 성숙도 모델 기반 시스템 평가 기술의 연구내용을 정리한 것으로, 한국정보통신대학교와 서울여자대학교 연구원들에 의해 수행중입니다.

본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다(2007-SW-11-DJ-02).

참고문헌

- [1] Ludwig, Lt. Gen Robert H., "The Role of Technology in Modern Warfare", speech presented to the Software Technology Conference, 1992.4.14.
- [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] "무기/비무기체계 내장형 소프트웨어 개발관리 지침", 국방부, 2002.1.1.
- [8] "국방전력발전업무 규정", 국방부 훈령 제793호,

- 2006.6.29.
- [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 인터뷰 결과”, DSRC 기술보고서, 2007.4.7.
- [13] 신현인, 박수현, “모델링 및 시뮬레이션에 의한 시험 평가”, 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] Erik van Veenendaal, “Guidelines for Testing – The Test Maturity Model”, TMMi-Foundation, <http://www.tmmifoundation.org/downloads/resources/TestMaturityModel.TMMi.pdf>.
- [17] Margaret Kulpa, Kent A. Johnson, “Interpreting the CMMI: A Process Improvement Approach”, CRC Press, 2003.
- [18] Ilene Burnsteine, “Practical Software Testing”, Springer, 2003.
- [19] Jef Jacobs, et. al., “The Process of Test Process Improvement”, XOOTIC Magazine, pp. 23~29, 2000. 11(November 2000, Volume 8, Number 2: Testing).
- [20] Tim Koomen, Martin Pol, “Test Process Improvement”, Addison-Wesley, 1999.
- [21] Boris Beizer, “Software Testing Techniques 2nd edition”, International Thomson Computer Press, 1990.
- [22] CMMI Upgrade Team, Standard CMMI Appraisal Method for Process Improvement(SCAMPI)A, Version 1.2 Method Definition Document, (CMU/SEI-2006-HB-002, ADA339225). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006.
- [23] Dunaway, D. K, CMMI®-Based Appraisal for Internal Process Improvement(CBA-IPI) Lead Assessor’s Guide(CMU/SEI-96-HB-003). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996.
- [24] Byrnes, P. & Phillips, M. Software Capability Evaluation, Version 3.0, Method Description(CMU/SEI-96-TR-002, ADA309160). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996.
- [25] Electronic Industries Association, Systems Engineering Capability Model, Part 2: EIA/IS-731-2 Appraisal Method, Washington, D.C.: 1998.
- [26] AFMC Pamphlet 63-103, Software Development Capability Evaluation(SDCE), Version 1.0. United States Air Force Material Command(AFMC), 1994.
- [27] Ibrahim, L.; LaBruyere, L.; Malpass, P.; Marciniak, J.; Salamon, A.; & Weigl, P. The Federal Aviation Administration Integrated Capability Maturity Model® (FAA-iCMM) Appraisal Method(FAM), Version 1.0. Federal Aviation Administration, 1999.
- [28] SCAMPI Upgrade Team, Appraisal Requirements for CMMI, Version 1.2.(CMU/SEI-2006-TR-011). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006.



류호연

2000 경상대학교 정보시스템 석사
 2005 경상대학교 정보시스템 박사
 2005 경성대학교 초빙외래교수
 2005~2006 한국과학기술연구원 Post-Doc
 2006~현재 한국정보통신대학교 공학부 연구교수
 관심분야 : 소프트웨어 테스트, 소프트웨어 신뢰성, 소프트웨어 보안, 소프트웨어 프로세스 개선, 온톨로지 공학

E-mail : hoyeon@icu.ac.kr



백종문

1993 조선대학교 컴퓨터과학 및 통계학과 학사
 1996 University of Southern California Computer Science 석사
 2000 University of Southern California Computer Science 박사
 2001~2005 미국 모토로라 SSERL (Software and Systems Engineering Research Lab) 수석연구원

2005~현재 한국정보통신대학교 공학부 조교수
 관심분야 : 소프트웨어 매트릭스, 소프트웨어비용추정, 소프트웨어 동적모델링, 소프트웨어 프로세스개선, 소프트웨어 품질보증, 소프트웨어 6 시그마

E-mail : jbaik@icu.ac.kr



이단형

1971 서울대학교 핵공학 학사
 1983 Arthur D Little School Of Management 석사
 1990 Virginia Commonwealth University 정보시스템 박사
 1993~1998 한국과학기술연구원 시스템 공학 연구소, 선임연구부 부장

1998~2000 LG-EDS 시스템, 부사장
 2000~2003 한국소프트웨어진흥원 원장
 2003~현재 한국정보통신대학교 공학부 교수
 관심분야 : 요구공학, 소프트웨어 프로젝트 관리, 소프트웨어 프로세스 개선

E-mail : danlee@icu.ac.kr



이병길

1984 University of Bridgeport 물리학과 학사
 1993 Auburn University 전산학과 석사
 1996 Auburn University 전산학과 공학 석사
 1998~현재 서울여자대학교 정보미디어대학 컴퓨터학부 부교수

관심분야 : 소프트웨어 아키텍처, 형상관리, 소프트웨어 프로세스 개선
 E-mail : byongl@swu.ac.kr