



한국정보과학회

제8권 제1호  
Vol. 8 No. 1



2006

# 한국 소프트웨어공학 학술대회 논문집

Proceedings of 2006 Korea Conference on  
Software Engineering

- 일시: 2006년 2월 16일(목) ~ 18일(토)
- 장소: 강원도 용평 리조트

주최 : 한국정보과학회, 한국정보처리학회

주관 : 한국정보과학회 소프트웨어공학 연구회  
한국정보처리학회 소프트웨어공학 연구회

후원 : (주)비트컴퓨터  
소프트웨어진흥원  
ITRC 소프트웨어연구센터  
ITRC 소프트웨어 프로세스 개선 센터

## 목 차

<b>초청특강 [2월 17일 금]</b>	
1. Software Process (Past, Present and Future) .....	3
이단형 교수(ICU)	
2. 임베디드 소프트웨어 개발 방법론 .....	19
천유식 사장(Korea MutalTech)	
3. 한국 S/W 산업의 추세 및 대학의 Software Engineering 교육 현실화 .....	25
김원 상근고문(삼성전자)	
4. Architecture_driven Software Development	
유효식 상무(삼성SDS)	
5. IT Service 業의 Business Model	
박준성 전무(삼성SDS)	
<b>주제 I : 임베디드 소프트웨어 I, II [2월 16일 목]</b>	
1. 임베디드 시스템 디버깅을 위한 동적 슬라이싱 기법 .....	31
서광익, 최은만(동국대)	
2. 임베디드 시스템의 동적 모델링을 위한 정형적 검증 방안 .....	39
양원석, 김민성, 박수용(서강대)	
3. Operational Architecture 에 기반한 임베디드 소프트웨어의 계층적 모델링 (우수논문) .....	49
전상욱, 배두환(KAIST), 홍장의(충북대)	
4. CEC 임베디드 소프트웨어 개발을 위한 최적의 제품개발방법론 선정 .....	57
우동성, 이종이, 윤희병(국방대)	
5. Extreme Programming 방법론을 이용한 하드웨어 소프트웨어 통합디자인상의 분할결정 (우수논문) ...	66
채희서, 이동현, 인호(고려대)	
6. Statechart를 이용한 Ravenscar Profile 기반 실시간 시스템 설계 .....	74
김창진, 최진영(고려대)	
<b>주제 II : 소프트웨어 품질 및 영상관리 I, II [2월 16일 목]</b>	
1. 연구 개발 감사 기반의 산업용 소프트웨어 품질 향상 연구 .....	87
오광근, 박순영, 문건일(LG산전), 박수용(서강대)	
2. Bayesian Inference for Software Growth Reliability Models using Gibbs Sampler .....	91
이상식(송호대학)	
3. 센서노드의 제약사항을 고려한 응용모듈버전 동기화 기법 .....	98
정선우, 김동규, 최재현, 정기원(송실대)	
4. 프레임워크 기반의 가용성 제공 기법 .....	106
박재걸, 하성호, 박진욱, 채홍석(부산대)	
5. 정적 실행시간 분석기의 기반 구조 .....	115
신원, 김태완, 김분희, 장천현(전국대)	
6. IT 시스템 보안수준관리 필요성 및 범위에 관한 연구 .....	124
김태훈, 오동훈(산성공사)	



**주제 III: 설계 및 아키텍처 [2월 17일 금]**

1. UML2.0 기반의 Context-Aware 응용 도메인 전용 아키텍처 모델링 언어 연구 ..... 131  
박용우, 김현성, 전태용(고려대)
2. 영역 전용 아키텍처 기술 언어를 사용한 모바일 상거래 응용 아키텍처의 동적 쿼리 모델링 ..... 139  
이승훈, 전태용(고려대), 신규상(ETRI)
3. 아키텍처 패턴의 분류 및 추상화 방법 ..... 147  
공상환(전안대)
4. 클래스 종속관계 분석을 통한 상위구조 아키텍처 복구 및 재구조화 ..... 157  
박관진, 우치수(서울대)

**주제 VI: 프러덕트 라인 [2월 17일 금]**

1. 권결 지침 프로그래밍의 분별요소를 적용한 프로덕트 라인의 가변 기능 조합 ..... 177  
허승원, 최은만(동국대)
2. 제품 계열을 위한 컨텍스트 기반 특성 지표 도출 ..... 187  
김건삼(ETRI), 엄희균, 안성웅, 황선명(대전대)
3. 워서 결합 단위의 활성화 규칙과 제품라인 행위 모델 간의 일관성 검증 (최우수 논문) ..... 189  
김현탁, 장엽, 양경모, 강교철(포항공대)
4. 워치 구현을 위한 EJB 컴포넌트 설계 기법 ..... 200  
민현기, 이진열, 김성인, 김학동(숭실대)

**주제 V: 재사용 및 소프트웨어 컴포넌트 I, II [2월 17일 금]**

1. 효율적인 소프트웨어컴포넌트 재사용을 위한 비즈니스 프로세스 모델과 컴포넌트 기반 개발의 매핑 지도 ..... 211  
서윤숙, 김영철(홍익대), 김재수(한국과학기술정보연구원)
2. 아키텍처 기반의 자가 성장 로봇 소프트웨어를 위한 저장소 구조 ..... 219  
구형민, 박유식, 김기현, 고인영, 최호진(ICU)
3. 컴포넌트 클러스터링의 실용적 평가 기법 및 지원 도구 (우수논문) ..... 228  
한만걸, 장수호, 김수동(숭실대)
4. 컴포넌트 기반 소프트웨어의 동적인 재구성 프레임워크 설계 ..... 236  
정철호, 이상희, 이은식(성균관대)
5. ESB 를 이용한 SOA 기반 서비스 개발 ..... 245  
이정원, 문은영, 최병주(이화여대)
6. Experience in Developing an Infrastructure for Components Reuse ..... 253  
김성아, 전현선, 김진태(삼성전자)
7. Product Line Engineering(Role-Based CBD Frame)을 적용한 대규모 CBD 프로젝트의 수행 ..... 258  
한상훈, 이상훈(LG CNS)
8. 레거시를 CBD로 변환하기 위한 3 차원 기반의 재공학 통합 메타모델 ..... 266  
조은숙(서일대)

**주제 VI: 요구공학 [2월 17일 금]**

1. 목표와 시나리오 기반 요구사항으로부터 기능 집수 도출방안 ..... 277  
최순향, 박수용, 한지영(서강대), 김진태(삼성전자)
2. 요구사항 협의모델을 위한 계층분석적 의사결정 방법 (우수논문) ..... 287  
김도훈, 이택, 인호(고려대)
3. 블루오션 창출을 위한 요구사항 계 정의 방법론 ..... 295  
김삼수, 임상원, 인호(고려대)
4. 사용자 행위 분석에 관한 연구: S/W 와 HCI 의 접목 ..... 305  
김예진, 김동호, 김영철(홍익대), 김기두(TTA)



**주제 VII: 프로세스 [2월 17일 금]**

- 1. CMMI의 Organizational Training PA를 지원하는 훈련 프로세스 프레임워크의 개발 ..... 315  
이은표, 이병걸(서울여대)
- 2. 요구사항 변경 프로세스의 정의 및 개선 ..... 322  
김한샘, 조성민, 한혁수(상명대)
- 3. 웹 애플리케이션 개발을 위한 기민한 프로세스 (우수논문) ..... 329  
이기열, 정우성, 이훈우, 김택수, 우치수(서울대), 이병정(서울시립대), 김희권(명동대)
- 4. 프로세스 관리 환경 설계 ..... 339  
김정아, 최승용(관동대)
- 5. CMMI 기반의 프로세스 및 제품 품질보증 활동 평가를 위한 Metric에 관한 연구 ..... 346  
양주미, 한혁수(상명대)

**주제 VIII: 소프트웨어 테스트 I, II [2월 18일 토]**

- 1. J2ME 대응 라이브러리 클래스 테스트 도구 개발 ..... 357  
오영노, 국승하, 김현수(중남대), 김철용, 윤석진, 최유희(ETRI)
- 2. 적용 사례를 통한 소프트웨어 테스트 평가모델 분석 ..... 365  
김기두, 신석규(TTA), 김영철(홍익대)
- 3. 테스트 주도 개발을 위한 추적 기반 의사결체 모델 ..... 370  
김택수, 박찬진, 우치수(서울대), 이병정(서울시립대)
- 4. 테스트노력 변화에 의한 소프트웨어의 신뢰도 영향에 관한 연구 ..... 380  
최규식(전양대)
- 5. 다차원적 데이터 분석을 지원하기 위한 소프트웨어 측정 데이터 모델의 확장 ..... 387  
전용호, 김수현, 윤경아, 배두환(KAIST)
- 6. CC 3.0에서 제시된 컴포넌트 복합제품 평가방법 ..... 396  
성윤기, 유연정, 이원식(한국정보보호진흥원)
- 7. 모델 체킹을 이용한 상태도 기반 소프트웨어 검증 ..... 401  
이태훈, 권기현(경기대)

**주제 IX: 소프트웨어 응용 및 프로젝트 관리 [2월 18일 토]**

- 1. UML 표기법 확장을 위한 언어의 설계와 구현 ..... 409  
이민규, 김현수, 정운태((주)플라스틱소프트웨어 기술연구소)
- 2. SystemC 기반의 가상 소프트웨어 개발 환경 (우수논문) ..... 413  
한상호, 조상영(한국외대), 이정배(선문대)
- 3. 판매자 중심의 다차간 동시협상 방법론 연구 ..... 421  
조민재, 최형림, 김현수, 홍순구, 박영재, 심정훈(동아대)
- 4. 개발자 능력을 고려한 소프트웨어 모듈 할당 자동화 기법 (우수논문) ..... 431  
강동원, 윤경아, 배두환(KAIST)

**주제 X: 소프트웨어 데이터 처리 [2월 18일 토]**

- 1. 온돌로지 기반의 단백질 반응 데이터 품질향상 기법 ..... 443  
장희선, 원민영, 김태경, 조완섭(충북대)
- 2. 바이오 데이터 통합과 데이터웨어하우스 구축 ..... 450  
박소희, 정승현, 남성혁, 정태성, 안명상, 조완섭(충북대)
- 3. 메타데이터 인터페이스를 이용한 DTD 기반 XML 문서 변환기의 골격 원시 코드 생성 (우수논문) ... 457  
최귀자, 남영광(연세대)

..... 131  
..... 139  
..... 149  
..... 157  
..... 171  
..... 180  
..... 188  
..... 200  
매평 시도  
..... 211  
..... 219  
..... 228  
..... 236  
..... 245  
..... 253  
..... 258  
..... 266  
..... 277  
..... 287  
..... 295  
..... 305



# 적용 사례를 통한 소프트웨어 테스트 평가모델 분석

김기두\*, 신석규, 김영철

한국 정보통신 기술협회\*, 홍익대학교 소프트웨어공학 연구실  
서울 분당구 서현동 267-2

kdkim@tta.or.kr\*, skshin@tta.or.kr, bob@selab.hongik.ac.kr

**요약** 본 연구에서는 소프트웨어 테스트를 위한 다양한 평가모델들을 살펴보고 모델들을 적용을 통하여 분석하고자 한다. 여러 모델들 중에서 한국정보통신기술협회(TTA)에서 시행중인 GS 시험평가모델과 테스트 성숙도 모델 및 테스트 향상 모델을 기반으로 개발된 Enhanced-TMM 을 통해 분석해 보았다.

즉, 국내 인증을 수행하고 있는 소프트웨어 평가모델과 소프트웨어 성숙도 측정을 목적으로 개발된 테스트 성숙도 모델을 사용하여 단계 및 평가 결과를 비교하고, 이를 통해 소프트웨어 품질을 향상 시킬 수 있는 방법을 제안하고자 한다.

**핵심어:** 테스트 성숙도 모델(Test Maturity Model : TMM), 테스트 프로세스 향상 모델(Test Process Improvement : TPI), TTA 의 GS 평가모델.

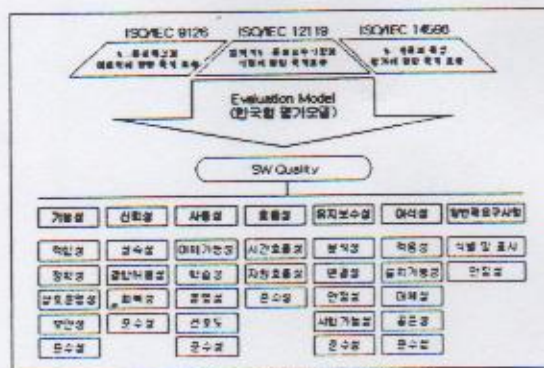
## 1. 연구배경

현재 많은 소프트웨어 조직에서는 개발 뿐만 아니라 소프트웨어 평가에도 많은 비중을 두려고 노력 중이다. 하지만 소프트웨어 시험을 위한 평가 모델이 국내에는 표준화 되지 않았고 적용 또한 일부에서만 이루어 지고 있다. 우리는 그 중에서 국내에서 소프트웨어 시험을 대표적으로 수행하고 있는 한국정보통신 기술협회의 한국형 평가모델과 능력 성숙도 모델을 개선한 Enhanced-TMM 을 적용하고 이를 통해 두개의 모델을 비교 분석하여 소프트웨어 품질을 개선할 수 있는 방향을 제시하고자 한다.

## 2. 관련 연구

국내 소프트웨어 개발 업체의 대부분은 중소·벤처 기업으로 그 규모가 영세하여 우수한 제품을 개발하여도 업체의 낮은 지명도와 마케팅

능력의 부족 등으로 시장개척에 어려움을 겪고 있다. 또한, 세계적인 추세로 볼 때, 사회 및 경제 발전과 더불어 소비자의 인식은 제품의 가격보다는 품질 및 안정성 등으로 그 관심이 변해가고 있으며, 기업에서는 얼마나 고객을 만족시킬 수 있는 고품질 제품을 제공하는 지의 여부가 주요한 성공 요인이 되고 있다. 산업체를 중심으로 소프트웨어 제품의 품질 향상을 위한 시험·인증 제도의 필요성이 제기되어 왔으나 시험·인증을 위한 장비투자 및 평가 기술 개발 등 막대한 초기 투자 비용과 수익성 불투명 등의 원인으로 민간 기업에서는 시험·인증 사업이 전혀 실행되지 못함으로 국가적 차원의 제 3 자 시험·인증 서비스 제도 도입 요구가 꾸준히 제기되어 왔다. 이런 계기로 정보통신부에서 소프트웨어 산업진흥법 제 13 조 및 동법 시행령 제 9 조, 제 10 조 등 관련 법령을 마련하고, 정보통신부장관 고시를 통하여 2000 년 9 월 ETRI(한국전자통신연구원)에 제 3 자 시험·인증 기관으로 SW 시험센터를 설립하게 되었으며, 2001 년도에 TTA(한국정보통신기술협회)로 조직을 이관하여 현재까지 SW 시험인증센터에서 시험·인증 서비스를 제공하고 있다.



<그림 1> GS 평가 기준[1]

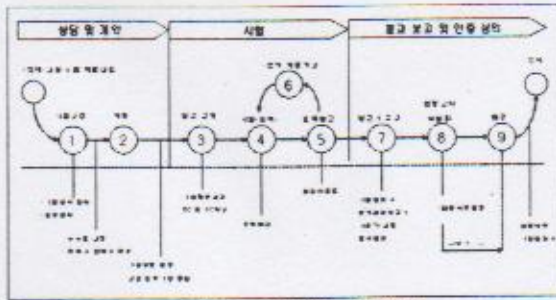
GS(Good Software) 인증제도는 문서심사만으로



인증을 부여하는 여다 인증제도는 달리 고객이 사용하게 될 실제 운영환경의 테스트베드를 갖추어 제품 품질을 인증하는 제도이다.

<그림 1>과 같이 소프트웨어 시험·인증을 위한 평가모듈은 국제 표준인 ISO/IEC 9126, ISO/IEC 12119, ISO/IEC 14598 에 근거하여 크게 7 가지 품질 특성으로 구성되어 있으며, 품질특성에 기반하여 시험·인증을 수행한다.

GS 시험·인증은 아래의 <그림 2> 와 같은 절차로 수행된다.

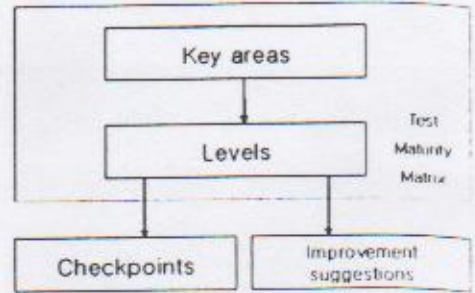


<그림 2> GS 평가 절차[1]

처음 소프트웨어 개발업체가 SW 시험인증센터에 시험·인증서비스를 신청하게 되면, 시험·인증 상담 후 수수료 및 일정을 산출하여 계약을 하게 된다. 시험 대상을 위한 시험환경을 설정한 뒤 시험을 수행한다. 시험 중 발생하는 결함을 개발업체에게 보내주어 제품을 개선하게 한다. 개선된 제품을 회귀시험을 하는 일련의 반복을 수행한다. 시험이 끝나면, 시험결과를 소프트웨어 분야별 전문가로 이루어진 인증심의위원회에 상정하여 품질인증 여부를 결정한다. 마지막으로 품질인증결과를 해당 개발업체에게 통보한 후 인증마크 및 인증서를 교부함으로써 절차가 끝나게 된다.[1]

이와는 달리 국외에서는 다양한 평가모델들을 개발하여 적용하고 있다. 그 중에서 테스트 프로세스 개선 모델(Test Process Improvement: TPI)은 1997년 Tim Koomen 과 Martin Pol 에 의해 개발된 테스트 프로세스 개선을 보다 쉽게 수행하기 위한 모델이다.

<그림 3>은 테스트 프로세스 개선 모델의 프레임워크를 보여준다. 테스트 프로세스 개선 모델은 현재 조직의 테스트 프로세스 강점과 약점을 파악할 수 있다. 또한 체크포인트를 통해 프로세스 성숙도를 평가하고, 개선 사항을 제시할 수 있다.



<그림 3> 테스트 프로세스 개선 모델 프레임워크

테스트 프로세스 개선 모델은 테스트 프로세스와 관련된 기술, 도구, 보고서 등을 다루는 핵심영역(Key Areas), 각 핵심영역을 측정하는 성숙도 수준을 분류하는 레벨(Levels), 각 핵심영역을 통해 결정되는 테스트 프로세스의 성숙도를 보여주는 테스트 성숙도 매트릭스(Test Maturity Matrix), 핵심영역을 객관적으로 평가할 수 있는 체크 포인트(Check Point), 측정된 수준보다 테스트 프로세스를 향상시키기 위한 개선 제안(Improvement Suggestions)으로 구성된다.

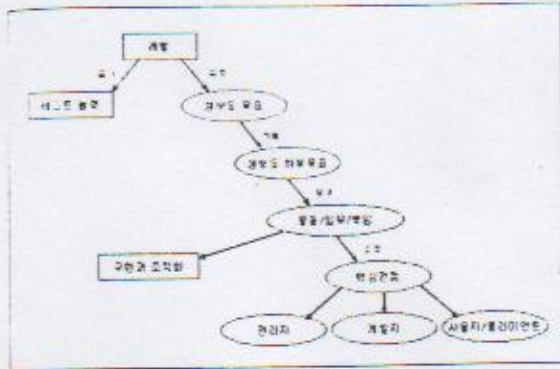
테스트 프로세스 향상 모델에서 핵심영역은 모두 20 개의 영역으로 구분되어 있으며 모든 테스트 프로세스를 커버한다. 핵심영역은 서로 다른 레벨을 받을 수 있으며 A-D 까지 존재한다. 이들 레벨은 테스트 프로세스를 단계적으로 향상시키기 위한 목표를 제공한다. 레벨의 분류를 위해서는 레벨이 갖고 있는 요구사항을 만족해야 한다. 이를 확인하는 것이 체크포인트이다. 체크포인트는 레벨을 객관적으로 결정할 수 있게 도와준다. 마지막으로 개선제안은 체크포인트에서 측정된 결과를 갖고 다음 레벨로 향상시키기 위한 방법을 제안한다. 개선제안은 강제적인 방법은 아니고 다만 다음 목표를 이루기 위한 조언이나 비결과 같은 의미로 사용된다.[2]

기존의 테스트 프로세스의 성숙도를 향상시키는 테스트 프로세스 향상 모델과 달리 과거 소프트웨어 공학에서 소개하는 소프트웨어 개발 프로세스를 테스트 초점으로 분석하여 테스트 관점의 프로세스 성숙도를 향상시키는 방법을 제안하는 것이 테스트 성숙도 모델(Test Maturity Model : TMM)이다. 테스트 성숙도 모델은 일리노이 공대의 Burnstein 교수팀에 의해 테스트 성숙도 모델은 능력성숙도 모델(Capability Maturity Model : CMM)을 기반으로 능력성숙도 모델이 갖고 있는 테스트 관점의 문제를 향상시킨 모델로 테스트 관리에서 제안된 이슈의 중요성, 성숙도의 각 레벨에서 권고된 실습, 조직의 테스트 프로세스를 평가하고 향상시키기 위한 평가모델 등을 제안하고



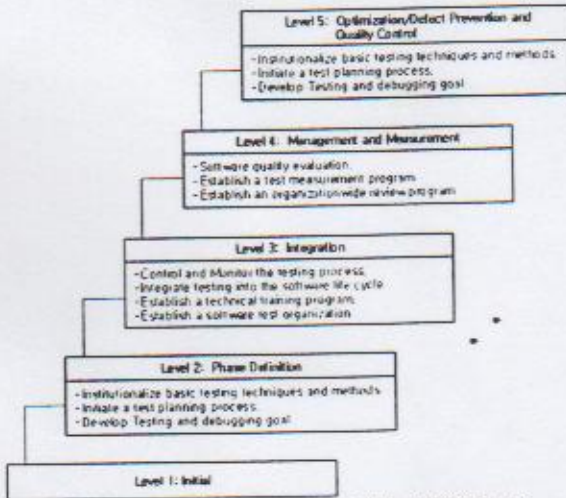
있다.

테스트 성숙도 모델의 프레임워크는 <그림 4>와 같다. 테스트 성숙도 모델은 테스트 성숙도를 표시하는 레벨이 존재하며, 레벨에서 이루어야 할 성숙도 목표(Maturity Goal)를 포함한다. 성숙도 하위 목표는 성숙도 목표를 지원하고, 활동/임무/책임에 의해 성숙도 하위목표가 달성된다. 활동/임무/책임은 관리자, 개발자, 사용자/클라이언트 관점으로 조직된다.



<그림 4> 테스트 성숙도 모델 프레임워크[3], [4]

테스트 성숙도 모델은 능력 성숙도 모델을 기반으로 만들어져서 능력성숙도 모델과 같이 5 개의 성숙도 레벨로 나누어 진다. 각각의 레벨은 성숙도 목표, 세부목표, 범위, 경계, 주요 활동, 필요 산출물 등을 정의하고 있다. 각각의 성숙도 레벨과 목표는 아래의 그림 5와 같다.



<그림 5> 테스트 성숙도 모델 레벨[3], [4]

-초기 수준 : 테스트 프로세스가 정의가 없는 단계이다. 초기 단계에서는 테스트가 코딩이 이루어진 후에 비계획적으로 이루어지고 테스트와

디버깅의 차이가 구분되지 않는다. 또한 테스트의 목적이 단순히 소프트웨어가 작동한다는 것을 보여준다.

-정의 수준 : 테스트와 디버깅이 구분되며 테스트가 소프트웨어 생명주기에서 하나의 독립된 단계로 정의된다. 정의 수준에서는 테스트가 계획되고, 정의되며, 기초적인 테스트 기술 및 도구들이 지원된다. 테스트의 목적은 주로 소프트웨어가 명세를 만족하는지 검증하는 것에 있다. 하지만 여전히 코딩이 이루어진 후에 수행된다.

-통합 수준 : 테스트가 더 이상 코딩 이후의 활동이 아닌 소프트웨어 생명주기 전체에 걸쳐 이루어진다. 테스트 목적이 사용자나 의뢰인의 요구사항에 맞춰 정립되며 이러한 요구사항이 테스트 케이스 설계에 반영된다. 또한 별도의 테스트 조직이 존재하며, 테스트 교육 훈련을 포함한 전문적 활동으로 인식된다.

-관리의 측정 수준 : 테스트 활동이 측정되고 정량화 되는 단계로 모든 개발활동에 대한 검토가 테스트와 품질제어 활동의 필수 요소로 인식한다. 검토는 검증과 확인을 위한 동료검토, 인스펙션, 워크스루 활동을 포함하며, 개발 제품뿐만 아니라 테스트 관련 문서들, 즉 테스트 계획 테스트 설계, 테스트 절차 등도 검토의 대상이 된다. 이런 활동들의 주요 목적은 오류들을 가능한 일찍 발견하려는데 있다.

-최적화 수준 : 이 단계에서는 결함 예방과 품질제어 활동이 핵심이 된다. 테스트 프로세스가 정의/관리되며, 비용과 효과가 추적/감시 된다. 최적화 단계에서는 테스트 프로세스가 지속적으로 개선되고, 조정되며, 결함 예방과 품질 제어 활동이 수행된다. [3], [4]

앞에서 언급한 테스트 프로세스 개선모델을 기반으로 테스트 성숙도 모델을 향상시킨 모델이 Enhanced-TMM 이다. Enhanced-TMM 은 테스트 성숙도 모델을 기반으로 개발된 모델이기 때문에 동일한 구조를 갖고 있다. 특히 테스트 성숙도 모델 프레임워크의 활동/임무/책임 단계에서 Test Attribute to Maturity Level Correlation Matrix를 사용하여 테스트 성숙도 수준 측정을 지원한다.

아래 (표 1)의 Test Attribute to Maturity Level Correlation Matrix 는 테스트 프로세스의 측정을 위한 매트릭스로 각 성숙도 레벨에서의 테스트 프로세스에서 도달해야하는 요소들을 보여준다. 각 테스트 요소들은 테스트 성숙도 모델을 기반으로 성숙도 목표와 테스트 프로세스 향상 모델의 테스트



프로세스 영역을 통해 제안되었다. 각 요소들은 30 개로 구성되어 있고, 각 요소별로 1-5 까지의 레벨에 도달하기 위한 요구들이 간략히 기술되어 있다. 특정 요소들은 레벨 5 까지가 아니지만 이전 레벨을 만족하면 이후 상위레벨까지 만족하게 된다.[6]

구분	Test Attribute	Level 1	Level 2	Level 3	Level 4	Level 5
Test Devise	Test Plan (TP)	0.00	0.00	0.00	0.00	0.00
	Test Strategy (TS)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
Test Quality	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
Test Technique	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
Test Document	Test Plan (TP)	0.00	0.00	0.00	0.00	0.00
	Test Strategy (TS)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
Test Infrastructure and Tool	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
Test Execution	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
Test Organization	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00

(표 1) Test Attribute to Maturity Level Correlation Matrix[5]

3 적용사례

평가대상을 GS 평가모델을 통해 평가한 결과 아래 (표 2)와 같은 결과를 얻었다. 7 개의 품질특성항목에 대한 평항목수가 있으며, 실제로 평가를 하여 측정된 결과가 총점이 되며, 마지막으로 총점을 백분율로 전환하여 결과를 얻는다. 해당 소프트웨어의 평가결과 일반적 요구사항(100%), 기능성(99.85), 신뢰성(99.95), 효율성(100%), 사용성(98.37), 유지보수성(100%), 이식성(100%)로 평가항목을 만족하였다. 각 항목의 기준은 GS 평가 기준을 따르고 있으며 테스트에 의해 평가된다.

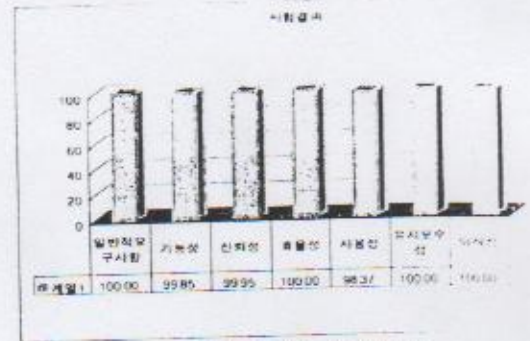
실제 시험대상에 대한 구체적인 사항에 대한 언급은 사내 대외비이기 때문에 언급하지 않았다.

품질특성	평항목수	총점	평균
일반적요구사항	2	2.00	100.00
기능성	8	7.99	99.85

신뢰성	4	4.00	99.95
효율성	5	5.00	100.00
사용성	12	11.80	98.37
유지보수성	2	2.00	100.00
이식성	8	8.00	100.00

(표 2) GS 평가결과

(표 2)에서 얻은 결과를 그래프로 나타내면 다음과 같다.



<그림 6> GS 점진결과

측정결과를 바탕으로 <그림 2>에서와 시험결과서와 평가결과보고서를 작성하여 받게 된다.

구분	Test Attribute	Level 1	Level 2	Level 3	Level 4	Level 5
Test Devise	Test Plan (TP)	0.00	0.00	0.00	0.00	0.00
	Test Strategy (TS)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
Test Quality	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
Test Technique	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
Test Document	Test Plan (TP)	0.00	0.00	0.00	0.00	0.00
	Test Strategy (TS)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
Test Infrastructure and Tool	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
Test Execution	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
Test Organization	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00
	Test Report (TR)	0.00	0.00	0.00	0.00	0.00
	Test Case (TC)	0.00	0.00	0.00	0.00	0.00
	Test Script (TS)	0.00	0.00	0.00	0.00	0.00

<그림 7> Enhanced-TMM 평가결과

같은 대상을 Enhanced-TMM 을 통해 평가하면 7과 같은 결과를 얻을 수 있었다.

평가 결과 Enhanced-TMM 에서 세팅하



대해 레벨 3 이상을 만족하였다. 또한 두 모델의 평가특성을 비교하여 아래(표 3)과 같았다.

비교	GS 평가모델	Enhanced-TMM
심사기반	-ISO/IEC 9126 -ISO/IEC 12119 -ISO/IEC 14598	-테스트성숙도모델 -테스트프로세스 개선모델 - 테스트프로세스평가모 델
평가영역	-7개 품질특성 -29개의 평가영역	-7개의 핵심영역 -30개의 Test Attribute
과거예심	-평가항목에 따른 소프트웨어 구현 평가	-개발 단계 전반에 따른 테스트 프로세스 평가
프로세스	- 순차적(부분반복적)	-Double V-Model
관여시점	-코딩 이후	-소프트웨어 생명주기와 동합
평가형태	-점검표	-체크리스트
심사요소	-평가항목	-심사절차 -질문서 -교육훈련 등
평가결과	점수(백분율)	레벨
인증여부	유	무

(표 3) GS 평가모델과 Enhanced-TMM 의 비교

우리는 두 모델의 모두 평가 항목에 대한 핵심분야는 유사하였으며, GS 인증이 부여되는 수준이 Enhanced-TMM 에서 제시하는 3 레벨을 수준을 만족되었다. 또한 개선 할 요소들을 제시 할 수 있다 즉, Enhanced-TMM 를 적용하여 개발한 소프트웨어의 품질은 국내 인증을 받을 수 있는 수준에 가까움을 알 수 있었다.

#### 4. 결론

높은 품질의 소프트웨어를 완성하는데 있어 개발 모델의 적용 만큼 소프트웨어 평가 모델 적용 또한 많은 비중을 차지 하고 있다. 우리는 국내에서 GS 시험·인증을 수행하고 있는 한국정보통신기술협회의 소프트웨어 개발제품 및 조직에 한국형 평가모델 과 Enhanced-TMM 을 적용해 보았다. 심사기반, 평가영역, 평가 프로세스 등의 항목과 평가 결과에 대해 비교하였다. 평가를 위해 참여하는 시점이 각각, GS 평가모델의 경우 제품의 코딩이 이루어진 이후에 적용하며 평가결과가 백분율로 환산된 점수로 나타나지만, Enhanced-TMM 은 개발주기 전반에 적용되며 평가결과가 1-5 까지의 레벨로 평가된다는 차이가 있었다.

하지만 두 모델 모두 평가 항목에 대한 핵심분야는 유사하였으며, GS 인증이 부여되는 수준이 Enhanced-TMM 에서 제시하는 3 레벨이상과 매칭되었다. 즉, Enhanced-TMM 를 적용하여 개발한 소프트웨어의 품질은 국내 인증을 받을 수 있는 수준에 가까움을 알 수 있었다. 만약 조직 개발단계에서 Enhanced-TMM 을 개발프로세스와 함께 적용한다면 국내 GS 시험인증 또한 받을 수 있을 것으로 예상된다. 이를 통해 테스트 기간과 비용을 절감되는 효과를 기대한다.

향후 Enhanced-TMM 을 기반으로 개발된 소프트웨어를 시험하여 결과를 정량화하는 연구가 필요하고, 다양한 소프트웨어의 적용을 통해 현재 진행 중인 임베디드 소프트웨어 분야에 적용하려고 한다.

#### 참고문헌

- [1] 한국정보통신 기술협회, "GS 인증제품 목록집", 8, 2005.
- [2] Tim Koomen, Martin Pol, "Test Process Improvement" Addison-Wesley, 1999
- [3] Ilene Burnstein, Taratip Suwannasart, C.R. Carlson "Developing a Testing Maturity Model: Part I", Crosstalk, August 1996
- [4] Ilene Burnstein, Taratip Suwannasart, C.R. Carlson "Developing a Testing Maturity Model: Part II", Crosstalk, September 1996
- [5] 김기두, 김영철, "테스트 프로세스 향상을 통한 테스트 성숙도 모델(TMM) 개선에 관한 연구", 홍익대, 2004
- [6] 김기두, 김영철, "테스트 프로세스 향상을 통한 테스트 성숙도 모델 개선에 관한 연구", 한국소프트웨어 공학 학술대회 논문집, 제 1 권 제 1 호, 2005년 2월