



한국정보과학회
KOREAN INFORMATION SCIENCE SOCIETY

제1권 제1호
Vol. 1 No. 1



2005

한국 소프트웨어공학 학술대회 논문집

Proceedings of 2005 Korean Conference on
Software Engineering

- 일시 : 2005년 2월 21일(월)~23일(수)
- 장소 : 무주리조트

주최 : 한국정보과학회, 한국정보처리학회

주관 : 한국정보과학회 소프트웨어공학연구회
한국정보처리학회 소프트웨어공학연구회
한국전자통신연구원

후원 : 한국소프트웨어진흥원
KAIST 소프트웨어프로세스 개선센터
건국대 소프트웨어 연구센터
(주)n3soft

◆ 논문 목차 ◆

| | |
|---|-----|
| 튜토리얼 (2월 21일) | |
| · 은혼로지 공학의 연구 동향 및 적용 사례 김홍기 교수 (서울대 치과대학) | 3 |
| · 정보보증을 위한 소프트웨어 공학 적용 사례 및 연구 필요성 김태훈 사무관(국군기무사령부) | 26 |
| 주제 1 : 심입체 I, II (2월 21일) | |
| · 기능 관점의 임베디드 소프트웨어 아키텍처 프레임워크 설계 방법 오광근, 문전일(LG산전연구소), 박수용(서강대) | 43 |
| · Embedded SW에서 SW Architecture 검증 방법에 관한 연구 박복남 (삼성전자) | 56 |
| · 국방 정보 시스템 간의 상호운용을 위한 LISI 기반 체크리스트 연구 류동국(국방과학연구소), 김우열, 김영철(홍익대) | 67 |
| · 초소형 저전력 센서 모듈을 위한 운영체제의 구조 및 설계 황태호, 송병철 (전자부품연구원), 조위덕(아주대) | 75 |
| · 전자정부참여광장을 통한 전자민주주의 실현 신영진, 이종현(행정자치부) | 84 |
| · J2EE 환경에서 DB2 for z/OS 성능 향상 방안 연구 최병석 (고려대) | 92 |
| 주제 2: 테스트 I, II (2월 22일) | |
| · Function Block Diagrams에 대한 제어 데이터 흐름 테스트 지은경, 유준범, 박수현, 차성덕 (한국과학기술원) | 105 |
| · 소프트웨어 신뢰도 평가 대상의 파라미터 연구 최규식(건양대학교) | 115 |
| · 개방형 서버의 상호운용성 테스트를 위한 테스트 케이스 생성 이지현, 노혜민, 유철중, 장옥배(전북대), 이준옥(한국전자통신연구원) | 122 |
| · 임베디드 소프트웨어 벤치마킹을 위한 BMT 모델 최현미, 성아영, 최병주(이화여대), 신석규(한국정보통신기술협회) | 136 |
| · 펌웨어 펌터링 기능 테스트를 위한 테스트베드 구축 박영대, 국승학, 김현수(충남대) | 145 |
| · 테스트 프로세스 향상을 통한 테스트 성숙도 모델(TMM) 개선에 관한 연구 김기두, 김영철(홍익대), 박병호(국군의무사령부) | 155 |
| 주제 3: 품질관리 I, II (2월 22일) | |
| · 국내 중소 SW 기업을 위한 품질관리 현황 파악 모델 최순황, 이장병, 박수용(서강대), 황만수(신홍대), 신재식, 문장원(소프트웨어진흥원) | 165 |
| · BlueJ 기반 객체지향프로그래밍 교육 및 품질평가 도구에 관한 연구 강유경, 황석형(선문대), 양혜순(호서대) | 180 |
| · ISO/IEC 9126기반의 모바일 게임 소프트웨어 시험을 위한 품질 특성 및 평가항목 최재현, 이우진, 정기원(숭실대) | 186 |
| · Weakest Precondition 함수를 이용한 형상관리 프로세스 활동 수행의 타당성 검증 방법 제안 장옥현, 이병걸(서울여대) | 195 |
| · 영상처리 어플리케이션 자동 생성에 필요한 개발자 지식 추출을 위한 소프트웨어 형상관리 방법 이정현(자르테크), 채옥삼(경희대) | 207 |
| · 대규모 시스템을 위한 ISO/IEC 12207 적용성 및 확장성에 대한 연구 성민숙, 강태현, 백인섭(아주대) | 220 |

테스트 프로세스 향상을 통한 테스트 성숙도 모델(TMM) 개선에 관한 연구

김기두, 김영철, 박병호*

홍익대학교 일반대학원 전자전산 공학과
충남 연기군 조치원읍 신안리 339-701
(kdkim, bob)@won.hongik.ac.kr

*국군의무사령부 전산통신실

요약: 본 연구에서는 현재 개발된 테스트 성숙도 모델 중, 1996년 일리노이 공대(IIT) 테스트 성숙도 모델(Test Maturity Model: TMM)에 테스트 프로세스 개선 모델(Test Process Improvement: TPI) 과 접목하여 테스트 프로세스 단점을 극복하고자 한다.

즉, 기존의 V-Model 을 개선한 Double V-Model 을 제안하고 테스트 프로세스 개선 모델이 제공하는 테스트 프로세스 개선 요소를 통해 TMM 테스트 프로세스와 테스트 인프라를 개선 하였다. 또한 현재 조직의 테스트 프로세스를 '테스트 속성과 성숙도 상호 관련 매트릭스'[10] 기반으로 측정된 결과를 통해 조직의 프로세스 개선을 제안 할 수 있다.

핵심어: CMM(Capability Maturity Model), 테스트 성숙도 모델(Test Maturity Model: TMM), 테스트 프로세스 향상 모델(Test Process Improvement: TPI), V-Model, Enhanced TMM

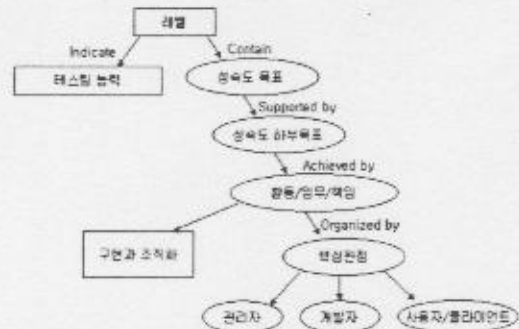
1. 연구배경

소프트웨어 산업의 발전에 따라 소프트웨어 개발 조직에서의 품질 향상은 중요한 이슈이다. 조직 안에서 소프트웨어 개발 프로세스, 완벽한 테스트, 또는 테스트 프로세스의 적용이 요구 되고 있다. 우리는 그 중 테스트 프로세스를 최적화하여 조직에 적용하는 것을 초점으로 하고 있다. 현재의 개발 프로세스 성숙도 모델(Capability Maturity Model: CMM)을 기반으로 개발된 테스트 관점의 테스트 성숙도 모델(Test Maturity Model: TMM)이 있다. 하지만 TMM 은 테스트 프로세스의 성숙도를 보여주기에 는 몇 가지 한계점을 갖고 있다. TMM 은 CMM 기반으로 만들어진 모델이기 때문에 함께 측정하지 않으면 큰 효과를 얻기 힘든 점, 개발 프로세스를 중심으로 만들어 졌기 때문에 테스트 전문조직을 위한 테스트 프로세스 관점이 약한 점 등이 있다.[1] TPI 의 경우는 테스트 프로세스 향상을 위한 제서를 갖고 있지만, CMM

기반에 대한 적용이 어렵다. 이를 해결하기 위해 본 논문에서는 기존의 개발 프로세스인 V-Model 을 확장한 Double V-Model 과 테스트 성숙도 모델과 테스트 프로세스 향상(Test Process Improvement: TPI) 모델을 접목한 Enhanced TMM 를 통해 문제를 해결하려고 한다. 또한, 현재 조직의 테스트 프로세스 문제점을 해결하여 최적 테스트 프로세스를 제안하고자 한다.

2. 관련 연구

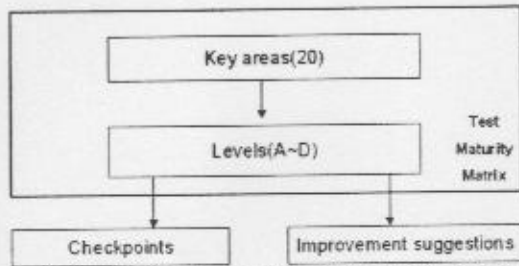
현재 개발되어진 성숙도 모델들 중 테스트 성숙도 모델은 1996년 일리노이 공대의 Burnstein 과 Carlson 의해 개발된 모델로 능력 성숙도 모델을 기반으로 개발된 테스트 성숙도 모델이다. 테스트 성숙도 모델은 능력 성숙도 모델의 테스트 활동에 대해 부족한 부분을 위해 개발되었으며 기존의 심사 모델들 중에서 가장 일관성 있는 모델 구조를 갖고 있다. 테스트 성숙도 모델은 심사 모델 및 절차, 심사 도구 및 절차서, 팀 교육 등에 관한 기준을 제시하고 있다. [2][3]



(그림 1) 테스트 성숙도 모델의 프레임워크[4]

테스트 성숙도 모델의 구조를 살펴보면 특정 성숙도 수준을 달성하기 위해서 각 수준에서의 주요 목표와 세부 목표들이 정의되어 있다. 또한 이 목표들을 달성하기 위해 필요한 활동, 테스트, 책임 등이 정의 되어 있다. 책임은 다시 관리자, 개발자/테스터, 사용자/클라이언트 영역으로 분류되는데, 활동, 테스트, 책임 등은 능력 성숙도 모델에서의 활동(Activities)에 해당하는 부분이다.

반면에, CMM 기반이 아닌, 테스트 프로세스 초점의 개선 모델인 테스트 프로세스 향상 모델은 Tim Koomen과 Martin Pol에 의해 1997년 개발된 모델로 테스트 프로세스의 개선에 도와주기 위해 개발된 모델이다. 테스트 프로세스 개선 모델은 현재 조직의 강점과 약점을 파악하고, 체크포인트를 통해 성숙도를 평가하고 개선 사항을 제시한다.



(그림 2) 테스트 프로세스 개선 모델의 구조[5]

(그림 2)에서는 테스트 프로세스 개선 모델은 테스트 프로세스에서 관련 기술, 도구, 보고서 등을 나타내는 핵심영역(Key Areas), 각 핵심영역을 시험하고 성숙도 수준을 분류하는 레벨(Levels), 모든 핵심영역을 통해 결정되는 테스트 프로세스의 수준을 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.[5]

(1) 핵심영역(Key areas)

핵심영역은 모두 20 개의 영역으로 나뉘어 진다. 이 핵심영역은 모든 테스트 프로세스의 범위를 포함한다. 각각의 핵심영역은 서로 다른 등급을 가질 수 있으며, 가장 낮은 등급은 A 에서 C 까지 일반적으로 갖고 있으며 영역에 따라 D 까지 있다.

(2) 레벨(Level)

TPI 레벨에서 핵심영역(Key Area)은 낮은 수의 성숙도 레벨을 갖고 있다. A~D 의 레벨을 갖고 있으며, A 보다는 B 가, B 보다는 C 가 높은 레벨을 의미한다. 레벨의 숫자는 각각의 핵심 영역에서 정의하는 수준이 같지는 않다.

각각의 높은 수준은 시간, 비용, 품질 면에서 이전의 것보다 좋다는 것을 말한다. 레벨을 사용하여 테스트 프로세스와 단계적으로 향상시키기 위한 목표를 제공하는 것을 결정할 수 있다.

레벨이 분류되기 위해서는 체크포인트를 적절하게 만족해야 한다. 레벨의 요구사항 (=체크포인트: Checkpoints)은 하위 레벨에 관련된 요구사항을 포함한다.

(3) 테스트 성숙도 매트릭스(Test Maturity Matrix)

테스트 프로세스 분석에서 각각의 핵심영역의 레벨이 결정된다. 이렇게 결정된 레벨을 기반으로 요구되는 높은 레벨의 형태로 제안될 수 있다.

(3) 체크포인트(Checkpoints)

테스트 프로세스의 핵심영역의 레벨을 객관적으로 결정하기 위해 사용된다. 각각의 레벨은 다수의 체크포인트를 갖는다. 각각의 레벨로 분류되기 위해서는 이런 포인트들을 만족해야 한다. 이런 체크포인트는 누적된다. 레벨 B 가 되기 위해서는 레벨 B 의 체크를 만족해야 하지만 레벨 A 의 체크포인트도 만족해야 한다.

(4) 개선 제안(Improvement Suggestions)

확실한 레벨의 체크포인트는 프로세스 향상을 목표로 한다. 다른 목표는 레벨별 개선 제안이다. 이런 제안들은 조언과 tips 처럼 의미되며, 레벨에 도달하기 위한 강제적인 방법은 아니다.

또 다른 TPAM 에서는 기존의 TPI 와 TMM 그리고 소프트웨어 평가와 테스트 KPA(Key Process Area)의 단점을 보완하기 위해 노력으로 만들어진 모델이다.[6]

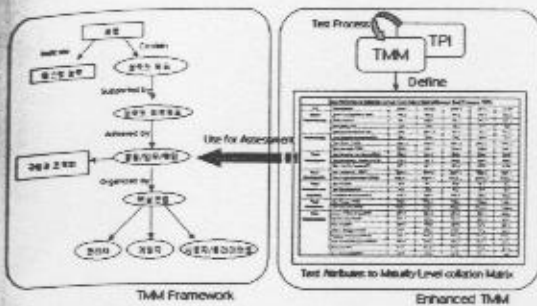
TPAM 은 동일한 CMM 구조와 단지 레벨 2 에서 3 으로 향상시키는데 도움을 줄 수 있는 모델이다.

TPAM 의 장점은 첫째, CMM 기반의 레벨 2 와 3 에서 능력 평가를 소프트웨어 성숙도로 보여주기에 용이한 것이다. 둘째는 CMM 과 TPAM 의 평가 데이터 수집과 분석의 일관성을 줄 수 있다는 것이다. 셋째, TPAM 을 통해 CMM 레벨 2 에서 레벨 3 으로 향상하기 위한 기초적인 소프트웨어 테스트 실재를 확립하는데 도움을 줄 수 있다.[6]

3. Enhanced TMM 프레임워크

이 장에서는 기존 TMM 상의 테스트 프로세스를 보완하기 위해 TPI 모델을 적용하는 확장된 TMM 을 언급한다. 아래 (그림 3)는 Enhanced TMM 의 개발 프레임워크를 보여준다.

Enhanced TMM 은 앞에서 언급한 TMM 의 테스트 인프라와 테스트 프로세스 개선에 대한 부족을 해결하기 위해 TPI 의 모델을 이용하여 해결하였다.

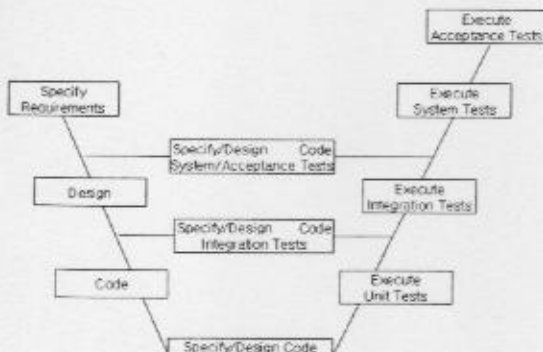


(그림 3) Enhanced TMM Framework

3.1 Double V-Model

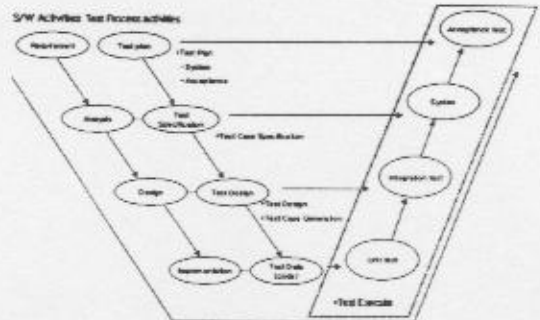
테스트 성숙도를 보여주는 기존의 TMM 에서는 테스트 활동에 적용될 수 있는 V-Model 을 사용하고 있다.

(그림 4)는 기존의 V-Model 테스트 프로세스와 개발 프로세스를 매칭하고 실제 테스트에서 각 단계에서의 활동 이후 문서적으로 확인하거나 리뷰하는 것을 제시하고 있다. 따라서, 본 연구의 TMM 의 테스트 활동 요소를 심사하기 위한 Enhanced TMM 을 개발하기 위해 V-Model 을 보다 개선된 모델을 제안한다.



(그림 4) 기존의 V-Model [1]

아래 (그림 5)은 확장한 Double V-Model 이다. 요구사항, 분석, 설계, 구현, 테스트의 각 단계에서 테스트 프로세스를 수행하기 위한 활동들을 제시하고 있다.



(그림 5) Double V-Model

Double V-Model 에서는 기존의 V-Model 과 같이 개발 프로세스 단계에서의 validation 과 테스트 인스펙션 활동들 그리고 그와 함께 테스트 조직이 따로 수행할 수 있는 테스트 프로세스를 함께 정의하고 있다. Double V-Model 의 오른쪽 시각영부분이 실제 테스트를 수행하는 부분이 되고 왼쪽부분이 병렬적으로 소프트웨어 활동(s/w activities)과 테스트 프로세스(test process activities) 함께 각 단계에서 이루어지게 되어 있다.

아래에서 그림의 왼쪽의 단계에서 수행되는 테스트 프로세스에 대해 정의하였다.

(가) 요구사항(Requirement)단계

- 테스트 계획(Test Plan) 수행: 테스트 활동에 있어서 포함되는 테스트 범위(Scope), 방법(Approach), 자원(Resource), 스케줄들을 문서로 기술한다. 아래는 Test Plan 단계에서 이루어져야 할 사항들이다.[7]
 - 테스트 계획 식별(Test plan identifier)
 - 테스트 아이템(Test Item)
 - 테스트해야 할 사항
 - 테스트 하지 말아야 할 사항
 - 방법(Approach)
 - 테스트 실패 기준
 - 테스트를 정지해야 할 기준/재테스트 할 때의 요구 사항
 - 테스트 임무
 - 테스트 요구 환경
 - 의무
 - 요구되는 팀원과 교육
 - 스케줄
 - 위험성과 우연성
 - 승인

(나)분석(Analysis)

- 테스트 명세(Test Specification): 테스트 명세 단계는 테스트 개발 문서로부터 테스트 케이스를 끌어내는 표준화된 방법이다.[5]

(다) 설계(Design)

- 테스트 설계(Test Design): 소프트웨어의 단일 특징이나 복합된 특징 그리고 관련된 테스트를 식별하는 것을 상세히 기술하여 문서화하는 것이다. 간단히 설계단계에서 이루어진 문서들을 검증하고 테스트 활동에 대해 구체적인 활동들을 문서화 하는 단계이다.[8]
 - 단위 테스트(Unit Test)에 대한 설계
 - 통합 테스트(Integration Test)에 대한 설계
 - 시스템 테스트(System)에 대한 설계
 - 적합성 테스트(Acceptance Test)에 대한 설계
 - 이례적인 것에 대한 보고서작성

(라) 구현(Implementation) 단계

- 테스트 데이터(Test Data(Code)): 개발 프로세스에서 실제 코딩이 이루어지는 단계로 이 단계에서는 테스트에 필요한 데이터와 코드, 혹은 테스트 스크립 등이 실제 생성되는 단계이다.

(마) 단위 테스트(Unit Test) 단계

- 단위테스트 단계(Unit Test)에서의 목적은 모듈이 제대로 구현되었는지 시험하는 것으로 모듈이 어떤 작업을 수행하는지를 나타낸 설계명세를 근거로 블랙박스 테스트를 하거나 프로그램 내부를 살펴서 논리 흐름을 체크하는 화이트 박스 테스트를 할 수도 있다.[9]

(바) 통합 테스트(Integration Test) 단계

- 통합 테스트 단계(Integration Test)에서는 소프트웨어 요소, 하드웨어 요소, 혹은 두개 모두에 대해 소프트웨어 프로그램의 증가된 부분들이 테스트의 순차에 따라 전체 시스템이 될 때 까지 묶여지고 테스트된다.[8]

(사) 시스템 테스트(System Test) 단계

- 통합된 하드웨어와 소프트웨어 시스템의 테스트 활동들이 시스템의 원래의 목적에 만족하는지 검증하고 확인하는 테스트 단계이다.[8]

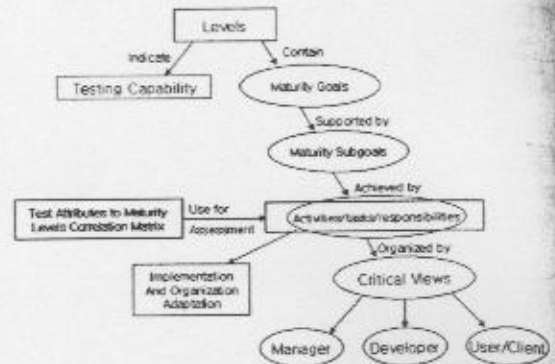
(아) 적합성 테스트(Acceptance Test) 단계

- 소프트웨어 시스템을 수용하기 전 단계에서 이루어지는 테스트로 실제 고객에게 전달되기에 적합관지를 시험한다.

3.2 테스트 속성과 성숙도 상호관련 매트릭스 (Test Attributes to Maturity Levels Correlation Matrix)

테스트 속성과 성숙도 상호관련 매트릭스는 TMM 프레임워크의 Activities/Task/Responsibilities

(활동/업무/책임) 평가를 위해 사용할 수 있는 매트릭스이다. 아래의 (그림 6)은 테스트 속성과 성숙도 상호관련 매트릭스가 TMM 프레임워크에서의 적용을 보여준다.[1]



(그림 6) 테스트 속성과 성숙도 상호 관련 매트릭스의 적용 [1]

테스트 속성과 성숙도 상호관련 매트릭스는 3 장에서 언급한 TMM 의 문제점 중 테스트 프로세스와 테스트 전문조직을 위한 성숙도들 제공할 수 있고, 성숙도 심사를 보다 용이하게 하도록 도움을 줄 수 있다.

가. 테스트 속성과 성숙도 상호관련 매트릭스

테스트 속성과 성숙도 상호관련 매트릭스[1][10]는 테스트 요소들을 7 개의 카테고리로 구성하며 카테고리 내에 평가를 위한 테스트 속성들로 구성된다. 각각의 속성들은 TPI 와 TMM 에서 제시하고 있는 테스트 활동들을 재정의 하였다. 각 속성들은 앞에서 언급한 TMM 의 레벨을 기준으로 다시 5 개의 레벨로 정의하였다.

<표 1>은 테스트 속성과 상호관련 매트릭스를 보여준다. 각각의 레벨을 만족하는 수준이 1-5 까지 정의되어 있는데 각 요소마다 다른 수준을 갖는다. 요소별로 레벨의 요구를 충족시키는 수준이 다르기 때문이다.

| Test Attributes to Maturity Levels Correlation Matrix | | | | | | |
|---|----------------------------------|---------|---------|---------|---------|---------|
| 구분 | Test Attribute | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 |
| Test Process | Test Process Definition (TPD) | TPD1 | TPD2 | TPD3 | TPD3 | TPD3 |
| | Test Reviews (TREV) | TREV1 | TREV2 | TREV3 | TREV4 | TREV4 |
| | Test Strategy(TS) | TS1 | TS2 | TS3 | TS4 | TS5 |
| | Test Moment of Involvement (TOI) | TOI1 | TOI2, 3 | TOI4 | TOI4 | TOI4 |
| Test Quality | Test Feedback Improvement (TFI) | TFI1 | TFI2 | TFI3 | TFI4 | TFI4 |

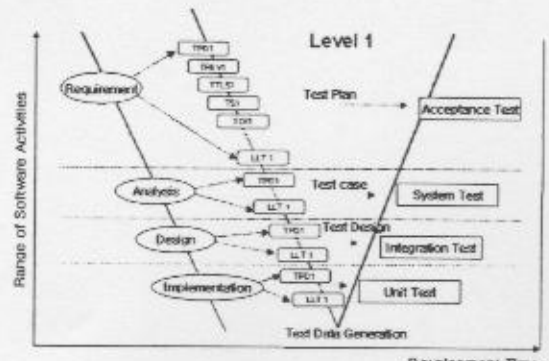
| | | | | | | |
|-----------------|-------------------------------------|--------|--------|--------|--------|--------|
| | Test Metrics (TMTC) | TMTC1 | TMTC2 | TMTC3 | TMTC4 | TMTC5 |
| | Test Standards (TSTD) | TSTD1 | TSTD2 | TSTD3 | TSTD4 | TSTD5 |
| Test Techniques | Test Estimating and planning (TEAP) | TEAP1 | TEAP2 | TEAP3 | TEAP4 | TEAP5 |
| | Test Specification Techniques(TSI) | TST1 | TST2 | TST3 | TST4 | TST5 |
| | Static Test Techniques (SIT) | SIT1 | SIT2 | SIT3 | SIT4 | SIT5 |
| Test Documents | Test Documents (TDOC) | TDOC1 | TDOC2 | TDOC3 | TDOC4 | TDOC5 |
| | Test Process Documents (TPDOC) | TPDOC1 | TPDOC2 | TPDOC3 | TPDOC4 | TPDOC5 |
| Infrastructure | Test Tools(TT) | TT1 | TT2 | TT3 | TT4 | TT5 |
| | Test Environment (TE) | TE1 | TE2 | TE3 | TE4 | TE5 |
| Test Education | Test Office Environment (TOE) | TOE1 | TOE2 | TOE3 | TOE4 | TOE5 |
| | Test Training (TTRN) | TTRN1 | TTRN2 | TTRN3 | TTRN4 | TTRN5 |
| Organization | Test Mentoring (TMEN) | TMEN1 | TMEN2 | TMEN3 | TMEN4 | TMEN5 |
| | Scope of Methodology (SOM) | SOM1 | SOM2 | SOM3 | SOM4 | SOM5 |
| | Communication (COM) | COM1 | COM2 | COM3 | COM4 | COM5 |
| | Reporting (RET) | RET1 | RET2 | RET3 | RET4 | RET5 |
| | Defect Management (DM) | DM1 | DM2 | DM3 | DM4 | DM5 |
| | Testware Management (TWM) | TWM1 | TWM2 | TWM3 | TWM4 | TWM5 |
| | Test Process Management (TPM) | TPM1 | TPM2 | TPM3 | TPM4 | TPM5 |
| | Evaluation (EVL) | EVL1 | EVL2 | EVL3 | EVL4 | EVL5 |
| | Low-level Testing(LLT) | LLT1 | LLT2 | LLT3 | LLT4 | LLT5 |

<표 1> Test Attributes to Maturity Levels Correlation Matrix

4.2 평가방법

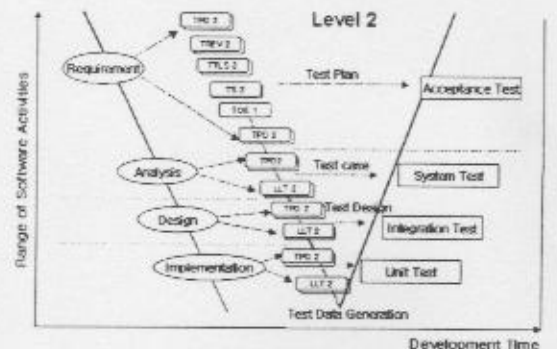
이 장은 Enhanced TMM 의 테스트 프로세스 성숙 Level 을 평가하는 방법에 대해 언급한다. Double V-Model 의 각 개발단계에서 평가 요소들의 측정한다. 모든 요소들의 평가된 수준에서 공통된 최대 레벨이 그 조직의 테스트 성숙도 수준이 된다.

(그림 7)은 Double V-model 을 이용하여 Enhanced TMM 요소의 적용을 설명하고 있다.



(그림 7) Double V-Model 을 통한 레벨측정(Level 1)

하나의 테스트 프로세스가 레벨 1 을 얻기 위해서는 각각의 단계, 즉 Analysis, Design, Implement 단계 내에서 모두 테스트 속성 레벨 1 의 요구를 만족해야 한다. 예를 들어 Requirement 단계(TMETH, TPD, TREV, TTLS, TS, TOI) , Analysis 단계의 Test Case, Design 단계의 Test Design, Implementation 단계의 Test Data Generation 역시도 모두 최소한의 수준이 레벨 1 을 만족해야 한다. 이 모델의 중요한 점은 각각의 개발 프로세스 단계에서 테스트 활동들이 병렬적으로 이루어지는 점에서 그만큼의 길함의 증식을 막을 수 있을 것이다.



(그림 8) Double V-Model 을 통한 레벨측정(Level 2)

(그림 8)은 평가 모델이 레벨 2 의 수준으로 측정됨을 보여준다. 레벨 2 에서는 해당 속성이 레벨 2 의 요구를 만족할 뿐만 아니라 레벨 1 의 정의도 함께 만족하고 있어야 한다. 나머지 레벨 측정방법 사례는 생략한다.[10]

수 있는 소프트웨어 속성과 임의에서의



매트릭스는 중 테스트 성숙도들이 용이하게

7 개의 기를 위한은 TPI 와 을 재정의의 레벨을

매트릭스를 1~5 까지 을 갖는다. 이 다르기

| | |
|---------|---------|
| Level 4 | Level 5 |
| TPD3 | TPD3 |
| TREV 4 | TREV 4 |
| TS4 | TS5 |
| TO4 | TO4 |
| TF14 | TF14 |

4.3 적용사례

SSTP 를 실제 개발 조직을 대상으로 적용한 결과는 아래와 같다. 적용대상의 표기는 적용 대상의 보안상의 문제로 회사명과 관련 문서에 대해서는 언급 하지 못하였다. 각 요소별로 크게 분류하여 결과를 아래와 같이 심사되었다.

| Test Quality | 문서번호 | No.2 |
|--------------|------|--|
| | 단계 | TFL, TMT, TSTD |
| 분석 | | -테스트 활동에서 보고서나 리뷰를 통해 피드백을 정확히 수행하고 있음 -테스트를 위한 측정 매트릭스가 존재 하며 이를 우선적으로 측정함 -테스트 표준화 작업이 활성화 되어 있어 대부분의 활동 들이 표준 문서를 기준으로 수행되고 있음 |
| 개선 | | -피드백 활동이 작업을 수행한 조직에서만 해당되며, 테스트 활동에 대한 히스토리가 문서화되어 있지 못해 외부의 조직과 결과를 공유하지 못하고 있음 -매트릭스의 내용이 테스트 측정을 위한 내용만을 우선으로 언급하여 테스트 프로세스에 대한 평가가 이루어 지고 있지는 못하고 있음 |

<표 2> Test Quality 에서 적용결과

| Test Techniques | 문서번호 | No.3 |
|-----------------|------|--|
| | 단계 | TEAP, TST, STT |
| 분석 | | -테스트 평가와 계획이 조직의 결과 보고서 를 통해 수행되며, 이는 표준 문서에 명시 되어 있음 -KS 안전 인증 표준을 기준으로 수행하고 있음 -테스트 체크리스트를 사용 하여 테스트를 보다 용이 하게 수행하고 있음 |
| 개선 | | -테스트웨어가 충분히 사용되고 있지만 재사용하지 못 하고 있는 실정, 이를 위해 테스트웨어의 관리가 필요시됨 -KS 안전인증을 따르고 있지만 테스트가 완료되었는지에 중점을 두기 때문에 대상 산출물들에 대한 테스트가 완전한지에 대한 평가가 어려운 실정 |

<표 3> Test Techniques 에서 적용결과

| Test Documents | 문서번호 | No.4 |
|----------------|------|--|
| | 단계 | TDOC, TPDOC, |
| 분석 | | -표준 문서에 테스트 문서에 대한 형식, 템플릿 등이 정의 되어 있어 이를 수행 하고 있음, -테스트 결과만을 기록하지 않고 테스트 과정에서 나오는 결과물에 대해서도 기록 하고 있음 |
| 개선 | | -문서에 대한 진산화가 부족하여 재사용에 어려움 |

<표 4> Test Document 에서 적용결과

| Test Infrastructure and Tool | 문서번호 | No.5 |
|------------------------------|------|--|
| | 단계 | TT, TE, TOE, |
| 분석 | | -테스트 도구에 대한 지원이 충분 하게 이루어지고 있으며 독립적인 테스트 팀에 의해 관리되고 있음 -여러 부분에 걸쳐 테스트 도구를 사용하고 있음 -테스트 환경이 개발 초기 시점에 정의 되어 최적화된 구성 -사무적 환경도 적절히 구성되어 있음 |
| 개선 | | -테스트 설계에 대한 도구 사용과 인식이 부족 -개발 초기에 구성된 테스트 환경이 테스트 요구조건에 따라 중간 단계에서 변화 하기 어려움, 이를 위해 적절 하게 테스트 요구를 수용할 준비를 하고 있어야 함 |

<표 5> Test Infrastructure and Tool 에서 적용결과

| Test Education | 문서번호 | No.6 |
|----------------|------|--|
| | 단계 | TTRN, TMEN |
| 분석 | | -테스트에 대해 전체적인 교육이 수행되고 있음 -업무에 대해 기능과 기술을 분류하고 있음 |
| 개선 | | -교육을 통해 테스트 프로세스의 향상을 도모하지만 상황에 따라 다른 결과를 얻고 있음 -정형화된 테스트 프로세스에 대한 교육이 필요함 -개발의 다른 단계에서도 사용할 수 있는 멘토링 프로그램이 필요 |

<표 6> Test Education 에서 적용결과

| Test Organization | 문서번호 | No.7 |
|-------------------|------|--|
| | 단계 | SOM, RET, DM, TWM, TPM, EVL, LLT |
| 분석 | | -테스트 표준 기술 문서를 통해 테스트의 영역과 방법, 보고에 대해 정의를 명시하고 있음 -발견된 결함을 모두 보고서에 작성하고 있음 -주간, 월간 보고를 통해 진행사항에 대해 보고함 -결함에 대한 관리가 충실히 이루어짐 |
| 개선 | | -테스트 분야이외에서도 여러 보고들이 이루어져야 함 -테스트웨어의 재사용문제를 수행하기 위해 전산화가 이루어 지서 관리 되어야함 |

<표 7> Test Organization 에서의 적용결과

