

테스트 프로세스 개선모델을 통한 테스트 성숙도 모델 (Test Maturity Model) 확장에 관한 연구

김 기 두[†] · 김 영 철^{**}

요 약

소프트웨어 산업의 발전에 따라 소프트웨어 개발조직에서는 소프트웨어 품질이 중요한 이슈로 부각되었다. 특히 여러 종류의 테스트 성숙도 모델을 통해 조직의 소프트웨어 테스트 성숙도 향상에 대해 다양한 방법으로 시도하고 있다. 하지만 현재 테스트 성숙도를 측정할 수 있는 모델들은 기존의 개발 성숙도 측정 모델을 기반으로 개발하여 테스트 부분에 대한 언급이 부족하고 단지 테스트 성숙도 레벨만을 측정하였다. 이러한 문제점을 해결하기 위해 소프트웨어 개발프로세스와 함께 테스트 프로세스를 수행할 수 있도록 이중 V모델과 테스트 성숙도의 용이한 평가를 위한 테스트 평가 속성과 레벨을 정의한 테스트 속성과 상호관련 매트릭스를 제안하였다. 테스트 프로세스 개선(Test Process Improvement: TPI)의 개선제안(Improvement Suggestion) 방법을 통해 테스트 성숙도 모델 확장하고, 측정된 성숙도를 향상 시킬 수 있는 개선방향을 제시한다. 또한, 본 연구에서 제안하는 테스트 성숙도 상호관련 매트릭스는 테스트 성숙도를 정량적으로 평가하기위해 사용하였다. 결과적으로는 확장된 테스트 성숙도 모델을 통해 조직의 테스트 성숙도를 높이는 데 기여 할 수 있으리라 생각한다.

키워드 : 테스트 성숙도 모델, 테스트 프로세스 개선, 테스트 속성과 성숙도 상호관련 매트릭스, 이중 V 모델

A Study of Enhanced Test Maturity Model with Test Process Improvement

Kidu Kim[†] · Young chul Kim^{**}

ABSTRACT

Organizations of Software development are very important issue on enhancement of a software quality as rapid progress of software industry. Especially there are diverse attempts for enhancement of test maturity of the software organization through some kinds of the test maturity model. But the current test maturity models based on CMM(Capability Maturity Model) lack part of actual testing measurement and only measure level of test maturity. To solve these problems, we suggest 'double V-model' to execute both software development process and test process simultaneously, and also 'test attributes to Maturity Levels Correlation Matrix' for evaluating level of test maturity included with definitions of test attribute and level. That is, we enhance TMM(Test Maturity Model) adopted with 'Improvement Suggestion' of TPI(Test Process Improvement) which is easy the evaluation of test maturity of organization and gives the direction of improvement to level up the test maturity for the measured organization. As a result, we will contribute to level up the test maturity of the organization.

Key Words : Test Maturity Model, Test Process Improvement, Test Attributes to Maturity Levels Correlation Matrix

1. 서 론

오늘날 소프트웨어 산업은 보다 높은 소프트웨어 품질을 획득하기 위해 많은 노력을 기울이고 있다. 이를 위한 방법으로 소프트웨어 개발 프로세스와 테스트 프로세스를 향상하려고 노력하고 있다. 완벽한 테스트를 통해 소프트웨어의 품질 보장을 확보할 수 있지만 완벽한 테스트를 수행하는

것은 불가능하기 때문에 정형화된 테스트 프로세스를 초점으로 연구하였다.

또한, 테스트 성숙도 모델의 성숙도를 향상시키기 위한 방법으로 테스트 프로세스 개선모델(Test Process Improvement: TPI)에서 제시하는 개선제안방법을 접목하였다.

기존의 소프트웨어 개발 프로세스의 성숙도를 보여주는 모델로는 능력 성숙도 모델(Capability Maturity Model: CMM), SPICE(Software Process Improvement & Capability dEtermination) 등이 있고, 테스트 프로세스의 성숙도를 보여주는 모델은 일리노이 공대의 테스트 성숙도 모델(Test Maturity

[†] 정 회 원: 한국정보통신기술협회 전임 연구원

^{**} 정 회 원: 홍익대학교 교수

논문접수: 2006년 11월 1일, 심사완료: 2006년 12월 26일

Model: TMM)이 있다[1-3].

CMM의 경우 소프트웨어 개발 프로세스의 성숙도를 측정할 수 있는 모델로 5개의 성숙도 레벨로 정의되어 있다. 각 단계는 초기(Initial)단계에서 최적화(Optimize) 단계로 정의된다. 하지만 개발 프로세스를 중심의 능력 성숙도 모델은 소프트웨어 테스트에 대한 언급이 부족하다. 이것을 보충하기 위해 일리노이공대에서 능력 성숙도 모델에서 일부 다루지 못한 테스트활동에 대한 프로세스 능력을 평가하기 위해 개발된 모델이 테스트 성숙도 모델이다.

테스트 성숙도 모델은 능력 성숙도 모델과 같이 성숙도 레벨이 5단계로 나뉘어진다. 각 레벨도 능력 성숙도 모델과 유사하게 처음 초기단계에서 마지막으로 최적화단계로 정의하고 있다. 하지만, 테스트 성숙도 모델은 능력 성숙도 모델을 기반으로 테스트에 초점으로 개발된 모델이기 때문에 테스트 프로세스를 테스트 성숙도 모델만으로 측정하는데 무리가 있다.

본 연구에서는 테스트 성숙도 모델에 테스트 프로세스 개선 모델을 통해 테스트 프로세스 성숙도를 향상시킨 TMM 확장을 제안한다. 확장된 TMM에서는 테스트 프로세스를 이중 V모델을 기반으로 한다. 이중 V모델은 개발 프로세스와 함께 테스트 프로세스를 함께 수행하는 프로세스이다. 테스트 프로세스와 함께 조직의 테스트 수준을 측정할 수 있는 요소를 정의한다. 정의된 각 요소는 매트릭스화 하여 테스트 속성과 성숙도 상호관련 매트릭스(Test Attributes to Maturity Levels Correlation Matrix)로 정의하여 테스트 성숙도를 측정하는데 보다 용이하게 하고자 한다. 또한 테스트 프로세스 개선을 통해 조직의 테스트 프로세스 성숙도를 향상시킬 수 있는 방법을 제안한다.

2장에서는 테스트 성숙도를 보여주는 테스트 성숙도 모델에 대해 설명하고, 테스트 성숙도 모델의 테스트 프로세스 부분을 보충할 테스트 프로세스 개선 모델과 테스트 프로세스 평가모델인 테스트 평가 모델(Test Process Assessment Model: TPAM)에 대해 설명한다. 3장에서는 테스트 프로세스의 성숙도를 평가할 수 있는 확장된 TMM을 제안하고, 4장에서는 확장된 TMM을 실제 적용한 사례를 기술하고 마지막으로 5장에서 본 논문의 결론과 앞으로의 연구 방향에 대해 언급하였다.

2. 관련연구

이번 장에서는 확장된 TMM 모델을 소개하고 테스트 프로세스 개선모델을 통해 조직의 테스트 성숙도를 측정하고 개선할 수 있는 평가모델을 제시한다.

2.1 테스트 프로세스 개선모델 (Test Process Improvement : TPI)

2.1.1 테스트 프로세스 개선 모델 개요

테스트 프로세스 개선 모델은 1997년 Tim Koomen과 Martin Pol에 의해 개발된 테스트 프로세스 개선을 보다 쉽게 수행하기 위해 개발된 모델이다. 테스트 프로세스 개선 모델은 조직의 현재 조직의 테스트 프로세스의 강점과 약점을 파악하여 체크포인트를 통해 프로세스 성숙도를 평가하고, 개

선 사항을 제시하고 있다. 테스트 프로세스 개선 모델은 테스트 프로세스에서 관련 기술, 도구, 보고서등을 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 제안으로 구성된다. 이는 확실한 레벨의 체크포인트는 프로세스 향상을 목표로 한다. 다른 목표는 레벨별 개선 제안이다. 이런 제안들은 조건과 비결처럼 의미되고, 레벨을 도달하기 위한 강제된 방법이 아니다[4].

테스트 프로세스 개선모델은 레벨 평가뿐만 아니라 개선 제안을 통해 현재 레벨보다 향상시킬 수 있는 방법을 제시해준다. 이를 통해 테스트 프로세스 향상을 유도한다. 그러나 테스트 프로세스에 대한 개선 및 제안을 할뿐 테스트 활동에 대한 제안을 하지 않아 테스트 수행을 향상시키지는 못하는 단점을 안고 있다.

2.2 테스트 프로세스 평가 모델(Test Process Assessment

Model: TPAM)

2.2.1 테스트 프로세스 평가 모델 개요

조직의 소프트웨어 프로세스를 평가하고 향상시키기 위한 SEI의 CMM이 성공적인 모델로써 사용되어 왔다. 이후 CMM을 향상시키기 위한 노력들이 여러 방면에서 노력되었다. TPAM에서는 기존의 TPI와 TMM 그리고 소프트웨어 평가와 테스트 핵심 프로세스 영역의 단점을 보완하기 노력하였다[9].

TPAM은 CMM과 동일한 구조를 갖는다. 또한 CMM과 같은 프로세스 특징을 지니고 있다. TPAM은 CMM의 레벨을 2에서 3으로 향상시키는데 도움을 줄 수 있는 모델이다. CMM 내부적으로 3개의 핵심 프로세스 영역을 갖는다. 3개의 영역들은 테스트 전략 계획(Test Strategy Planning), 테스트웨어 설계와 유지보수(Testware Design and Maintenance), 테스트 수행과 보고(Test Execution and Reporting)로 정의된다. 각각의 영역들은 핵심프로세스 영역의 목적과 내용을 보여주는 핵심프로세스 영역과 핵심프로세스 영역 목표 그리고, 위임, 수행능력, 수행된 활동들, 측정과 분석, 구현 검증의 활동으로 구분된다. 활동들 안으로 수준을 각각 정의하고 있다.

TPAM의 CMM기반의 레벨 2와 3에서 능력 평가를 소프트웨어 성숙도로 보여주기에 용이하지만 CMM 레벨 2와 3에 국한되는 모델이라는 한계점을 갖고 있다. CMM의 레벨을 3을 만드는 것도 중요하지만 테스트 프로세스 성숙도를 측정하는데 있어 특정 수준에서만 적용되는 모델은 문제가 있을 것으로 예상된다.

2.3 테스트 성숙도 모델(Test Maturity Model: TMM)

2.3.1 테스트 성숙도 모델 개요

테스팅은 소프트웨어 개발 프로세스의 핵심 요소 중 하나이다. 일반적인 조직에서는 고품질의 소프트웨어 개발에 테

스트의 중요성을 완전히 인식하지 못하고 있다. 테스트 성숙도 모델은 테스트 프로세스 개선과 향상에 중점을 두고, 조직을 가이드로써의 서비스를 제공하는데 있다.

소프트웨어 개발에서 테스트는 매우 중요한 역할임에도 불구하고, 기존의 능력 성숙도 모델들은 제안된 테스트 이슈에 적당하지 않다. 또한, 잘 정의된 테스트 프로세스 성숙도의 본질을 지니고 있지 않다. 이러한 결여된 사항을 보완하기 위해 테스트 성숙도 모델을 개발하게 되었다.

테스트 성숙도 모델에서는 능력 성숙도 모델이 갖고 있는 아래의 문제점을 해결하기 위해 노력하였다.

- 테스트 성숙도의 개념이 언급되어 있지 않음
- 키 프로세스 영역에 테스트 이슈가 충분하지 않음
- 테스트능력, 테스트가 적당한지에 대한 기준, 테스트 계획, 소프트웨어 검증처럼 품질에 관련된 내용들이 충분히 언급되지 않음[12].

2.3.2 테스트 성숙도 모델 배경

과거 소프트웨어 공학에서 소개하는 소프트웨어 개발 프로세스는 폭포수 모델, 혹은 스파이럴 모델, 프로토타입 모델 등이 알려져 있다. 현재는 조직의 개발 최적화를 위해 개발된 RUP(Rational Unified Process), CBD(Component Based Development), XP(eXtreme Programming) 등이 개발되어 있다.

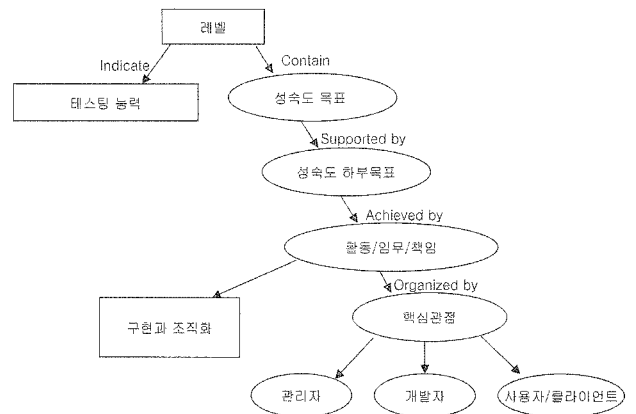
하지만 이들 모두는 소프트웨어 제품의 개발 프로세스이지 테스트 프로세스는 아니다. 물론 일부 개발 프로세스 중에는 테스트 프로세스를 제공하고 있지만 실제 테스트만을 위한 정형화된 프로세스는 아니다.

이중에서 테스트 성숙도 모델은 일리노이 공대의 Burnstein 교수 팀에 의해 개발된 테스트 성숙도를 측정하기 위해 개발된 모델이다. 테스트 성숙도 모델은 1996년 능력 성숙도 모델에서 테스트 활동에 대한 프로세스 개선을 보조하기 위해 개발되었으며, 기존의 심사 모델 중에서도 가장 일관성 있고 완전한 성숙도 모델 구조를 갖고 있으며, 심사 모델 및 절차, 심사 모델 및 절차, 심사 도구 및 질문서, 팀 교육 등에 관한 기준을 제시하고 있다.[5]

2.3.3 테스트 성숙도 모델의 구조

테스트 성숙도 모델 프레임워크에는 테스트 능력을 나타내는 레벨들이 있다. TMM에서 성숙도 레벨은 모두 5레벨로 정의하고 있다. 또한, 레벨은 각각의 레벨에서 수행되어야 하는 성숙도 목표를 포함하고 있다. 성숙도 목표는 하부 목표를 갖고 있다. 각 목표들을 달성하기 위한 각각의 활동/업무/책임들이 존재한다. 각 활동/업무/책임은 관리자/개발자/사용자의 영역으로 분류되어 있다. 아래의 (그림 1)은 테스트 성숙도 모델의 프레임워크를 보여준다.

TMM은 테스트 성숙도를 각 레벨별로 정의하고 있으며, 테스트 조직의 성숙도를 평가하고 이를 통해 테스트의 전반적인 품질을 향상시켜주는 장점을 갖고 있다. 하지만 테스트 프로세스에 대한 제시를 하지 못하여 테스트 프로세스 향



(그림 1) 테스트 성숙도 모델의 프레임워크 [5]

<표 1> 테스트 평가모델 비교

구 분	TPI	TPAM	TMM	E-TMM
평가레벨	14	2	5	5
심사기반	없음	TMM	CMM ISO SPICE	TMM TPI
핵심영역	14	13	13	29
테스트 프로세스	미적용	V모델	V모델	이중 V모델 사용
테스트 프로세스 개선제시	가능	불가능	불가능	가능

상을 위한 방법을 제시하지 못하고 있다.

<표 1>는 위의 테스트 평가모델들에 대한 비교 분석을 보인다.

다음 장에서 우리는 TMM에 테스트 성숙도를 측정할 수 있는 TPI의 접목을 통해 TMM의 개선을 제안한다.

3. TMM모델의 확장

소프트웨어 산업 발전에 따라 개발조직에서는 소프트웨어 품질 향상을 중요한 이슈로 삼고 있다. 조직 안에서 소프트웨어 개발 프로세스(development process), 완벽한 테스트(complete test), 또는 테스트 프로세스(test process)의 적용이 요구되고 있다. 우리는 그 중 테스트 프로세스를 최적화하여 조직에 적용하는 것에 초점을 두고 있다.

관련연구에서 우리는 여러 가지 평가모델을 언급하였다. 그중 능력 성숙도 모델(Capability Maturity Model: CMM)을 기반으로 개발된 테스트 관점의 테스트 성숙도 모델(Test Maturity Model: TMM)은 CMM을 기반으로 만들어진 모델이기 때문에 CMM과 함께 측정하지 않으면 큰 효과를 얻기 힘들다. 또한 개발 프로세스를 중심으로 만들어 졌기 때문에 테스트 전문조직을 위한 테스트 프로세스에 대한 관점이 약하다. TPI의 경우는 테스트 프로세스 향상을 위한 제시를

갖고 있지만, CMM 기반에 대한 적용이 어렵다. 이를 해결하기 위해 본 논문에서는 기존의 개발 프로세스인 V모형을 확장한 이중 V모형과 테스트 성숙도 모델과 테스트 프로세스 개선 모델을 접목한 확장된 TMM을 통해 TMM과 TPI의 문제를 해결하려고 한다.

3.1 확장된 (Enhanced) TMM 프레임워크

이번 장에서는 기존 TMM상의 테스트 프로세스를 보완하기 위해 TPI 모델을 적용한 확장된 TMM을 언급한다. (그림 2)는 확장된 TMM의 개발 프레임워크를 보여준다.

확장된 TMM은 기존의 TMM이 갖고 있는 테스트 인프라와 테스트 프로세스 개선에 대해 언급이 부족한 부분을 해결하기 위해 TPI의 모델을 이용하였다[6].

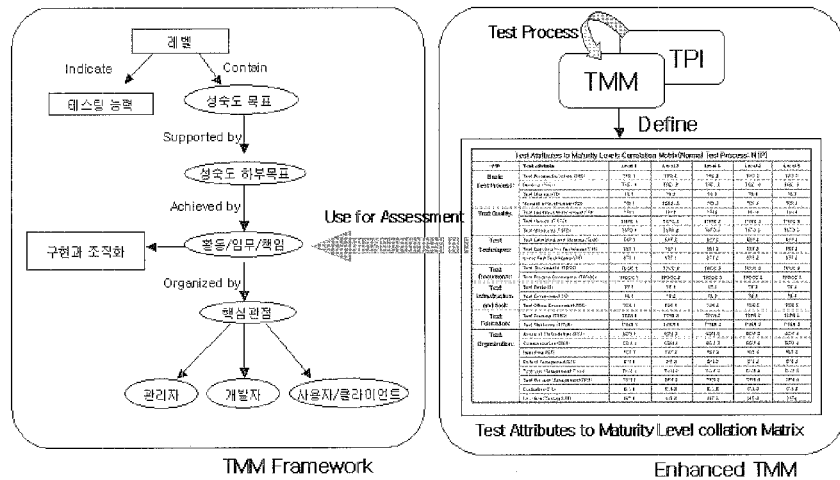
TPI 모델은 테스트 프로세스와 함께 테스트를 수행할 요소 및 활동 내용들을 제안하고 있다. 또한 각각의 요소들 중에서 테스트 인프라에 대해서도 언급하고 있다. TPI에서는 제안하는 테스트 프로세스 개선모델을 추가하여 현재의

테스트 프로세스의 장단점을 파악할 수 있을 뿐만 아니라 테스트 프로세스 개선을 위한 제시가 가능하다. 그리고 테스트 속성과 성숙도 관련 매트릭스를 통해 테스트 전문조직을 위한 성숙도를 제공할 수 있다. 마지막으로 확장된 TMM은 기존의 V모형을 확장한 이중 V모형을 이용하기 때문에 CMM의 개발 프로세스에 영향을 받지 않고 테스트 프로세스를 수행할 수 있다.

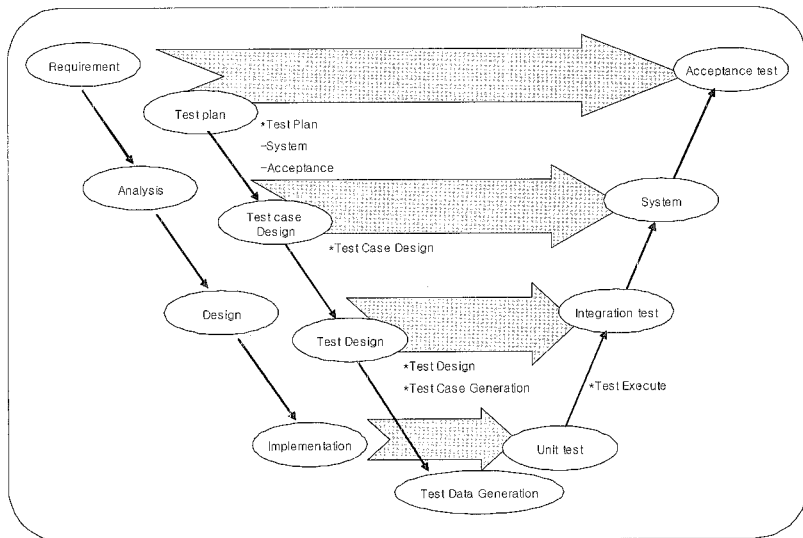
3.2 이중 V모형

기존의 개발 프로세스인 V모형은 폭포수 모델의 단점을 보완하고자 각 단계마다 정확히 수행하고 verification과 validation 수행을 강조한 모델이다[9, 10]. 본 연구에서는 확장된 TMM의 테스트 활동 요소를 심사하기 위해 기존의 V모형을 향상시킨 이중 V모형을 제안한다[6-8].

(그림 3)은 V모형을 확장한 이중 V모형으로 소프트웨어 개발 프로세스 (요구사항, 분석, 설계, 구현, 테스트)의 각 단계에서 테스트 프로세스를 동시에 수행하기 위한 활동들을 제



(그림 2) 확장된 TMM Framework



(그림 3) 이중 V모형

시 하고 있다.

이중 V모델에서는 기존의 V모델과 같이 개발 프로세스 단계에서의 validation과 테스트 인스펙션 활동들 그리고 그와 함께 테스트 조직이 따로 수행할 수 있는 테스트 프로세스를 함께 정의하고 있다. (그림 3)에서처럼 이중 V모델의 오른쪽 타원형부분이 실제 테스트를 수행하는 부분이 되고, 왼쪽의 두 부분이 개발 프로세스부분과 테스트 프로세스가 함께 각 단계에서 이루어지는 활동들이다.

3.3 테스트 속성과 성숙도 상호관련 매트릭스

대표적인 테스트 프로세스 모델인 TMM에서는 다양한 조직의 서로 다른 테스트 프로세스를 향상 시킬 수 있는 방법을 제시하지 못하고 있다. 본 연구에서는 TMM에 TPI를 적용한 테스트 속성과 성숙도 상호관련 매트릭스를 통하여 테스트 성숙도 프로세스를 향상시키려 한다.

본 연구에서는 앞 장에서 언급한 TMM Framework(그림 1)

의 활동/임무/책임 단계에서 조직을 정량적으로 평가하기 위해 테스트 속성과 성숙도 상호관련 매트릭스를 제안하였다.

<표 2>는 TMM의 테스트 프로세스의 측정을 매트릭스로 각 성숙도 레벨에서의 테스트 프로세스에서 도달해야 하는 요소들을 보여준다. 각 테스트 요소들은 TMM을 기반으로 성숙도 목표와 TPI의 테스트 프로세스를 통해 구현되었다. 각 요소들에 대한 정의를 기술하였고, 지면상의 문제로 1~5까지 레벨에 도달하기 위한 요구들에 대한 설명은 참조 문헌[6]에 기술하였다.

3.3.1 요소별 활동

1) Test Process Definition(TPD)

테스트 프로세스가 정형화된 정의가 존재여부를 확인한다. TPD에서는 테스트 프로세스의 각 단계(계획, 준비, 명세, 실행 등)가 각각 구별되어 있으며, 각각의 단계는 하부에 여러 개의 활동 요소들이 정형화 여부를 확인한다.

<표 2> 테스트 속성과 성숙도 상호관련 매트릭스

Test Attributes to Maturity Levels Correlation Matrix						
구분	Test Attribute	Level 1	Level 2	Level 3	Level 4	Level 5
Test Process	Test Methodology (TMETH)	TMETH1	TMETH2	TMETH3	TMETH4	TMETH4
	Test Process Definition (TPD)	TPD1	TPD2	TPD2	TPD2	TPD2
	Test Case Generation(TCG)	TCG1	TCG2	TCG3	TCG4	TCG4
	Reviews (TREV)	TREV1	TREV2	TREV3	TREV4	TREV4
	Tool Support (TTLS)	TTLS1	TTLS2	TTLS3	TTLS4	TTLS4
	Test Strategy(TS)	TS1	TS2	TS3	TS4	TS4
	Moment of Involvement(MOI)	MOI1	MOI2	MOI3	MOI4	MOI4
Test Quality	Feedback Improvement (TFI)	TFI1	TFI2	TFI2	TFI2	TFI2
	Management Control (TMC)	TMC1	TMC2	TMC3	TMC4	TMC4
	Test Metrics (TMTC)	TMTC1	TMTC2	TMTC3	TMTC4	TMTC5
	Test Standards (TSTD)	TSTD1	TSTD2	TSTD3	TSTD3	TSTD3
Test Techniques	Estimating and planning(EAP)	EAP1	EAP2	EAP2	EAP2	EAP2
	Test Specification Techniques(TST)	TST1	TST2	TST2	TST2	TST2
	Static Test Techniques(STT)	STT1	STT2	STT2	STT2	STT2
Test Documents	Test Documents (TDOC)	TDOC1	TDOC2	TDOC2	TDOC2	TDOC2
	Test Process Documents (TPDOC)	TPDOC1	TPDOC2	TPDOC3	TPDOC4	TPDOC5
Test Infrastructure and tool	Test Tools(TT)	TT1	TT2	TT3	TT3	TT3
	Test Environment(TE)	TE1	TE2	TE3	TE3	TE3
	Test Office Environment(TOE)	TOE1	TOE1	TOE1	TOE1	TOE1
Test Education	Test Training (TTRN)	TTRN1	TTRN2	TTRN3	TTRN3	TTRN3
	Test Mentoring (TMEN)	TMEN1	TMEN2	TMEN3	TMEN3	TMEN3
Test Organization	Commitment And Motivation(CAM)	CAM1	CAM2	CAM3	CAM3	CAM3
	Scope of Methodology(SOM)	SOM1	SOM2	SOM3	SOM3	SOM3
	Communication(COM)	COM1	COM2	COM3	COM3	COM3
	Reporting(RET)	RET1	RET2	RET3	RET3	RET3
	Defect Management(DM)	DM1	DM2	DM3	DM3	DM3
	Testware Management(TWM)	TWM1	TWM2	TWM3	TWM4	TWM4
	Test Process Management(TPM)	TPM1	TPM2	TPM3	TPM3	TPM3
	Evaluation(EVL)	EVL1	EVL2	EVL2	EVL2	EVL2
Low-level Testing(LLT)	LLT1	LLT2	LLT3	LLT3	LLT3	

2) Test Review(TREV)

테스트를 수행하는데 소프트웨어 프로젝트의 소스를 통해서만이 아닌 중간 산출물, 요구사항, 명세 등의 문서를 리뷰하는 것으로 버그를 줄일 수 있다.

3) Test Strategy(TS)

Test Strategy에서의 목표는 테스트가 결함(defect)을 얼마만큼 초기에 그리고 적은 비용을 사용하여 발견하는 것이 가장 중요하다. Test Strategy에서는 무엇에 의해 테스트 할 것인가에 대해 요구사항과 위험을 정의하고 있다.

4) Test Moment of Involvement(TMOI)

이전 과거의 테스트는 구현단계 이후의 작업으로, 테스트의 적용 시간이 후기 작업으로 인식되어왔었다. 하지만 차후 테스트의 적용 시점이 앞당기게 되었다. 실제 좀더 빠른 시점의 테스트는 가능한 빨리 그리고 가능한 쉽게 결함을 찾는 데 도움을 준다. 심지어 결함을 예방하는 효과를 주기도 한다.

5) Test Feedback Improvement(TFI)

테스트 수행은 단 한번의 수행으로 작업을 끝낼 수 없다. 다른 개발 프로세스와 같이 반복적으로 작업을 수행해야 한다. 특히 테스트 작업은 수행한 작업에 대해 정확한 결과를 전달해 주어야 한다.

6) Test Metrics(TMTC)

테스트 매트릭스는 각 단계의 결과를 다른 테스트 결과들과 비교하는데 유용하다. 테스트 프로세스에 대해 테스트된 시스템의 품질과 프로세스의 진보는 매우 중요하다. 매트릭스는 명세적으로 테스트 프로세스의 진보를 평가하는데 매우 중요한 역할을 한다.

7) Test Standard(TSTD)

조직 내에 테스트를 위한 표준이 존재하고 이를 준수하는 것만으로도 그 조직은 높은 수준의 테스트 단계를 갖고 있다고 할 수 있다. 테스트 표준의 의미는 조직마다 조금씩 다를 수 있으나 각각의 항목에 대해서는 범용적으로 같은 의미를 갖고 있어야 한다. 테스트 표준은 각 단계별로 테스트의 기준을 제시하고 있어야 한다.

8) Test Estimating and Planning(EAP)

테스트 계획과 평가는 언제 그리고 얼마나 많은 자원들을 필요로 수행되는지에 대한 활동들을 나타낸다.

9) Test Specification Techniques(TST)

테스트 명세 기술은 소스 정보로부터 테스트 케이스를 이끌어내는 표준화된 방법이다.

- 테스트 전략(strategy)의 입증된 내용을 통해 올바른 장소에서 올바른 커버리지를 만들 수 있다.
- 결함들은 테스트 케이스의 무작위 식별보다 효과적인 방법에서 발견되어질 수 있다.

- 테스트는 재사용가능하다.

10) Static Test Techniques(STT)

동적인 테스트는 수행중인 테스트에서 테스트 케이스의 수행이다. 이것은 항상 가능하거나 요구되지는 않는다. 즉 정적인 테스트는 작업 중인 소프트웨어를 갖고 테스트가 수행되지 않을 때 이다.

11) Test Document(TDOC)

소프트웨어 테스트 문서는 각각의 테스트 수행 후 작성될 수 있는 문서를 의미한다. 일반적으로 테스트를 수행한 결과를 문서화한 것을 말한다. Test Process Documents(TPDOC)와는 달리 Test Documents는 하나의 테스트 기법을 수행하고 나온 결과를 문서화 한 것을 말한다.

12) Test Process Documents(TPDOC)

테스트 프로세스 문서는 하나의 소프트웨어를 테스트 하기 위한 일련의 프로세스에 관련된 문서이다. 테스트 계획, 테스트케이스 생성, 테스트 수행들의 프로세스를 수행하면서 나오는 문서이다. 즉 테스트 프로세스 문서는 테스트 프로세스 품질에 영향을 끼친다.

13) Test Tools(TT)

테스트 도구는 하나 혹은 그 이상의 계획, 제어, 명세, 수행, 분석 등의 테스트 활동을 지원하는 자동화된 수단이다. 테스트를 수행하는데 오직 사람의 힘으로만 하는 것은 비용이나 시간의 측면으로 봤을 때 많이 소요된다. 테스트를 수행하는데 인간이 하기 어려운 작업을 도구를 통해 수행하면 테스트 작업을 시간과 비용 면에서 효과를 얻을 수 있다.

14) Test Environment(TE)

테스트 수행에 있어 테스트 환경 또한 중요한 의미를 갖는다. 왜냐하면 테스트를 수행하는 장소가 바로 테스트 환경이기 때문이다. 테스트환경의 중요 요소들은 하드웨어, 소프트웨어, 통신수단, 테스트파일, 데이터베이스를 만들고 사용하기 위한 설비, 절차가 있다. 테스트 환경은 품질, 인도시간, 비용에 많은 영향을 준다.

15) Test Office Environment(TOE)

테스트를 수행하는데 있어 테스트의 사무적 환경도 테스트 수행에 영향을 미칠 수 있다. 예를 들어 테스트를 위한 개인적인 방, 책상, 의자, PC, 전화 등이 있을 수 있다. 사무적 배치가 좋고 시간 효율적으로 되어 있다면 작업에 소요되는 시간, 비용이 보다 효율적인 것이다. 또한 좋은 테스트 사무 환경은 테스트 프로세스의 품질에 긍정적인 영향을 미칠 것이다.

16) Test Training(TTRN)

테스트 프로세스에서 테스트 팀의 올바른 구성은 매우 중요하다. 이들에게는 다양한 분야, 기능, 지식, 기술들이 복합적으로 필요하다. 테스트 전문적인 의견, 테스트 되어야 할

문제의 주제에 대한 지식, 조직에 대한 지식, 일반적인 IT에 대한 지식 등이 요구된다. 사회적인 기술 또한 매우 중요하다. 이런 복합적인 것을 얻기 위해서는 훈련이 필요하다.

17) Test Mentoring(TMEN)

테스트 조직 혹은 팀을 향상시키기 위해서는 그 조직을 이끌 수 있는 지도자의 힘이 중요하다. 전체 조직이 올바르게 테스트 프로세스를 수행하는데 테스트 팀을 올바른 방향으로 인도하는 역할을 수행하게 된다. 테스트 프로세스를 올바르게 이끄는 것뿐만 아니라 에러를 발견하여 수정하거나 잘못된 테스트 수행을 하였을 때 다시 정상적인 궤도로 돌아오게 하는 역할을 갖는다.

18) Scope of Methodology(SOM)

조직 내에서 각각의 테스트 프로세스를 위해 활동들, 절차, 훈련, 기술들 이외에 기타 등의 확실한 방법론이나 작업 매소드가 쓰여 진다. 이러한 방법론들은 작업의 시간이나 효율성이 모두 다르다. 조직이 방법론을 사용하는 목적은 효율적인 일반성을 어디서라도 적용할 수 있는 것이다.

19) Communication(COM)

테스트 프로세스에서 Communication은 다양한 사람들을 포함하는 모든 종류의 방법에서 발생할 수 있다. 이러한 Communication의 형태는 좋은 상태를 생성하고 테스트 전략을 조종하는 사이에서 테스트 프로세스를 부드럽게 진행하는데 있어서 매우 중요하다.

20) Reporting(RET)

테스팅은 제품의 품질 수준에 대한 통찰력을 제공하듯이 '결함 발견'만을 말하는 것은 아니다. 그래서 Reporting은 테스트 프로세스의 가장 중요한 산물로 고려되어 진다.

21) Defect management(DM)

비록 결함의 관리는 사실 프로젝트 문제이지만 테스터들에게 특별히 제한되지는 않는다. 좋은 관리는 결함의 생명 주기를 모니터링하고 다양한 개략(overview)을 줄 수 있어야 한다. 예를 들어 이런 개략은 품질 설명서를 입증하는데 사용된다.

22) Testware management(TWM)

테스팅하는 제품은 유지보수와 재사용 가능해야하고, 따라서 관리되어야한다. 테스트 계획, 테스트 명세, 테스트 데이터베이스와 같은 테스트 그 자체는 이전 프로세스의 디자인 자체처럼 실현이 관리되어진다.

23) Test process management(TPM)

테스트 프로세스와 각각의 활동들을 관리하는 것은 의미 있는 일이다. 특히 거친 테스트 프로세스에서 제어된 테스트 프로세스는 테스트를 수행하는데 매우 중요하다.

24) Evaluation(EVL)

용어 적으로 Evaluation은 기능적 디자인처럼 중간단계의 산출물인 인스펙션에 속하는 것으로 한다. Evaluation의 가장 중요한 점은 테스트와 함께 개발 프로세스에서 매우 빨리 결함을 발견 할 수 있어야 한다. 이것은 제작업의 비용을 훨씬 적게 사용하게 해준다.

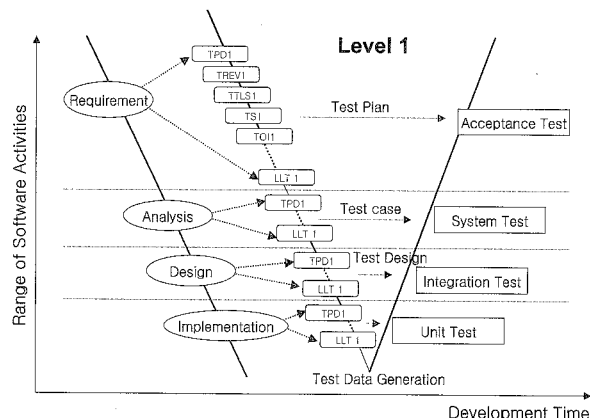
25) Low-level testing(LLT)

Low-level 테스트는 개발자에 의해 수행된다. 잘 알려진 Low-level 테스트는 단위 테스트와 통합 테스트이다. 평가와 같이, 이 테스트는 High-level 테스트가 수행하는 것 보다 시스템 개발 경로의 초기 단계에서 결함을 찾을 수 있다. Low-level 테스트는 사람이 야기하는 결함과 그것을 수정하는 검사원이 요구하는 환경이 적기 때문에 효율적이다.

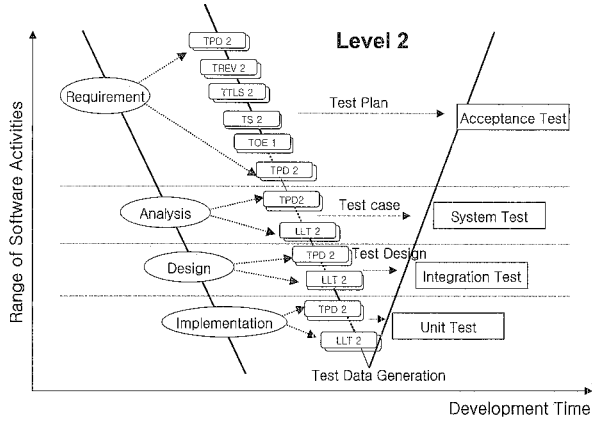
3.4 평가방법

이번 장에서는 이중 V모델과 확장된 TMM을 통해 Level 1~Level 5까지 평가하는 방법에 대해 언급한다. 이중 V모델의 각 개발단계에서 평가 요소들의 측정한다. 요소별 활동에서 정의하는 레벨을 기준으로 평가결과를 만족하는 공통 레벨이 해당 조직의 성숙도 수준을 나타낸다. 각 레벨에서 평가할 요소는 테스트 속성과 상호관련 매트릭스에서 정의한 레벨별 요소들을 대상으로 한다. 각 레벨별로 만족해야하는 수준은 테스트 속성별로 다르게 정의되어 있다. 예를 들어 레벨 3에서 TOE의 항목의 경우 정의한 수준이 TOE 2를 달성하게 되면 레벨 3을 만족하게 된다.

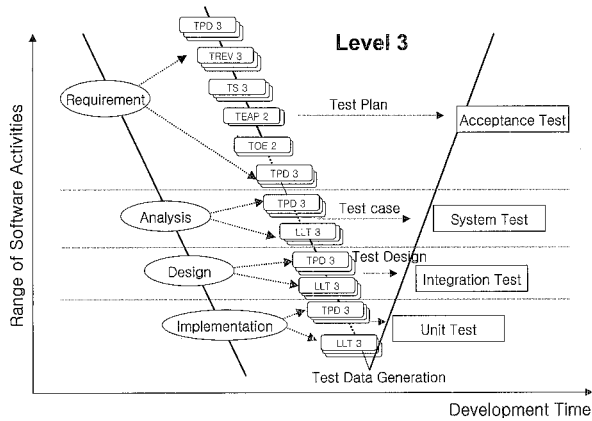
(그림 4)과 같이 하나의 테스트 프로세스가 레벨 1을 얻기 위해서는 다른 단계, 즉 Analysis, Design, Implement 단계에의 심사요소들까지 모두 레벨 1의 요구를 만족해야한다. 예를 들어 레벨 1일 때 Requirement 단계에서 심사한 TMETH, TPD, TREV, TTLS, TS, TOI 등의 레벨이 최소한 레벨 1을 모두 만족해야한다. Analysis의 Test Case, Design 단계의 Test Design, Implementation단계의 Test Data Generation에서도 모두 최소한의 수준이 레벨 1을 만족해야만 심사 조직의 수준이 레벨 1이라 할 수 있다. 이 모델의 중요한 점은 각각의 개발 프로세스 단계에서 테스트 활동들이 병렬적으로 이루어지는 점이다. 기존의 테스트 활



(그림 4) 이중 V모델을 통한 레벨측정(Level 1)



(그림 5) 이중 V모델을 통한 레벨측정(Level 2)



(그림 6) 이중 V모델을 통한 레벨측정(Level 3)

동들은 코딩이 끝난 이후의 수행되어 왔다. 그러나 이것은 증식된 테스트 버그로 인해 소프트웨어 개발비용이 테스트와 유지보수에 집중되는 문제점을 야기한다[6].

확장된 TMM의 이중 V모델에서는 각 개발단계에서 이루어질 수 있는 테스트 활동들을 동시에 수행하기 때문에 결합의 증식을 막을 수 있다.

(그림 5)은 평가 모델이 레벨 2의 수준으로 측정됨을 보여준다. 레벨 2에서는 해당 속성이 레벨 2의 요구를 만족할 뿐만 아니라 레벨 1의 정의도 함께 만족하고 있어야 한다.

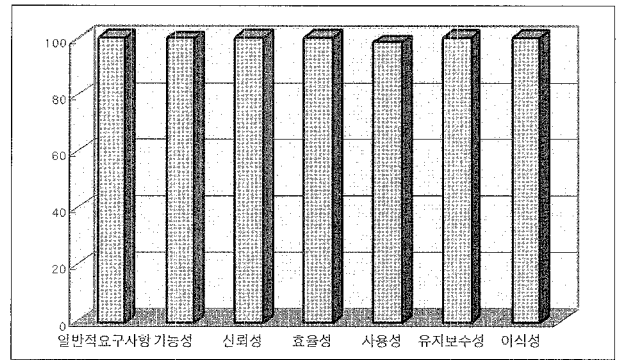
(그림 6)은 조직의 테스트 프로세스 성숙도를 측정된 결과 레벨 3을 획득한 모습이다. 레벨 3에서는 각 속성별 요구 수준이 레벨 3을 만족하는 수준이 되어야한다. 레벨 3은 레벨 2에서와 같이 요구되는 수준이 레벨 1~3까지의 요구를 모두 만족해야한다. 하지만 몇몇의 속성들은 예를 들어 TDP와 같은 경우에는 레벨 3을 얻기 위해서는 레벨 2이상만 획득하면 된다.

레벨 4와 5도 역시 이전 레벨과 같은 방법으로 측정된다.

4. 적용사례

4.1 평가 대상

본 연구의 적용대상을 TTA에서 시험한 소프트웨어 제품



(그림 7) GS평가결과

<표 3> 확장된 TMM 평가결과

		레벨 3									
		Test Attributes to Maturity Levels Correlation Matrix									
구분	Test Attribute	Level 1	Level 2	Level 3	Level 4	Level 5					
Test Process	Test Methodology (TMETH)	TMETH1	TMETH2	TMETH3	TMETH4	TMETH5					
	Test Process Definition (TPD)	TPD1	TPD2	TPD3	TPD4	TPD5					
	Test Case Generation (TCG)	TCG1	TCG2	TCG3	TCG4	TCG5					
	Reviews (TRV)	TRV1	TRV2	TRV3	TRV4	TRV5					
	Test Support (TTL)	TTL1	TTL2	TTL3	TTL4	TTL5					
	Test Strategy (TS)	TS1	TS2	TS3	TS4	TS5					
Test Quality	Moment of Deviation (MOD)	MOD1	MOD2	MOD3	MOD4	MOD5					
	Feedback Improvement (FI)	FI1	FI2	FI3	FI4	FI5					
	Management Control (TMC)	TMC1	TMC2	TMC3	TMC4	TMC5					
	Test Metrics (TMC)	TMTC1	TMTC2	TMTC3	TMTC4	TMTC5					
	Test Standards (TSTD)	TSTD1	TSTD2	TSTD3	TSTD4	TSTD5					
	Test Techniques	Estimating and planning (EAP)	EAP1	EAP2	EAP3	EAP4	EAP5				
Test Specification Techniques (TSF)		TSF1	TSF2	TSF3	TSF4	TSF5					
Static Test Techniques (STT)		STT1	STT2	STT3	STT4	STT5					
Test Documents	Test Documents (TDOC)	TDOC1	TDOC2	TDOC3	TDOC4	TDOC5					
	Test Process Documents (TPDOC)	TPDOC1	TPDOC2	TPDOC3	TPDOC4	TPDOC5					
Test Infrastructure and tool	Test Tools (TT)	TT1	TT2	TT3	TT4	TT5					
	Test Environment (TE)	TE1	TE2	TE3	TE4	TE5					
	Test Office Environment (TOE)	TOE1	TOE2	TOE3	TOE4	TOE5					
Test Education	Test Training (TRN)	TRN1	TRN2	TRN3	TRN4	TRN5					
	Test Mentoring (TMEN)	TMEN1	TMEN2	TMEN3	TMEN4	TMEN5					
Test Organization	Commitment And Motivation (CAM)	CAM1	CAM2	CAM3	CAM4	CAM5					
	Scope of Methodology (SOM)	SOM1	SOM2	SOM3	SOM4	SOM5					
	Communication (COM)	COM1	COM2	COM3	COM4	COM5					
	Reporting (REP)	REP1	REP2	REP3	REP4	REP5					
	Defect Management (DM)	DM1	DM2	DM3	DM4	DM5					
	Resource 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					

으로 하였다. 대상제품은 GS 인증시험 평가에서 아래와 같은 결과를 얻었다.

GS시험에는 7개의 품질특성에 대한 평가 항목 수가 있으며, 실제평가를 통해 측정된 결과가 총점이 된다[11]. (그림 7)와 같이 대상 제품은 평가결과 일반적 요구사항(100%), 기능성 (99.95%), 신뢰성(99.95%), 효율성(100%), 사용성(98.37%), 유지보수성(100%), 이식성(100%)로 평가항목을 만족하였다.

각 항목의 기준은 GS시험평가 기준을 따르고 있으며 GS 인증시험에 Pass된 SW제품을 평가 대상으로 선택하였다. 실제 시험대상에 대한 설명은 사내대외비로 언급하지 않았다.

4.2 평가에 대한 개선안 제시

본 연구에서 제시한 이중 V모델에 따라 테스트 속성과 성숙도 상호관련 매트릭스를 통해 평가하여 대상을 평가하여

<표 3>과 같이 각각의 항목들에 대해 3레벨을 만족하였다.
 확장된 TMM을 통해 조직을 테스트 프로세스를 분석하고, 분석 결과를 통해 테스트 프로세스에서 개선되어야 할 사항들을 제시할 수 있었다.

<표 4> Test Process에서 적용결과

Test Process	문서번호	No.1
	단계	TPD, TREV, TS, TMOI
분석	- 조직 내에 체계적인 테스트 프로세스가 정의되어 있으며 정립된 프로세스를 프로젝트에 운영함 - 테스트 단계별로 테스트 문서를 통해 결함 추적 이 이루어짐 - 테스트 팀내에서 리뷰를 수행하며 정형화된 문서 포맷이 존재함	
개선	- 리뷰를 위한 도구가 필요함 - 테스트 단계, 리뷰단계가 개발 단계와 함께 수행 할 필요가 있으며 개발 초기단계에 정의되어야함	

<표 5> Test Quality에서 적용결과

Test Quality	문서번호	No.2
	단계	TFI, TMTC, 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 활동들이 표준 문서를 기준으로 수행되고 있음	
개선	- 조직 내 다른 개발팀과 테스트 결과를 공유 할 필요가 있음 - 조직 전체에서 사용할수 있는 매트릭스가 필요함	

<표 6> Test Techniques에서 적용결과

Test Techniques	문서번호	No.3
	단계	TEAP, TST, STT
분석	- 테스트 평가와 계획이 조직의 결과 보고서를 통해 수행되며, 이는 표준 문서에 명시 되어 있음 - 테스트 체크리스트를 사용 하여 테스트를 보다 용이하게 수행하고 있음	

<표 7> Test Document에서 적용결과

Test Document	문서번호	No.4
	단계	TDOC, TPOC
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	

<표 8> Test Infrastructure and Tool에서 적용결과

Test Infrastructure and Tool	문서번호	No.5
	단계	TT, TE, TOE,
분석	- 테스트 도구에 대한 지원이 충분하게 이루어 지고 있으며 독립적인 테스트 팀에 의해 관리되고 있음 - 여러 부문에 걸쳐 테스트 도구를 사용하고 있음 - 테스트 환경이 개발 초기 시점에 정의되며 최적화된 구성 - 사무적 환경도 프로젝트에 맞게 구성되어 있음	
개선	- 테스트프로세스 자동화가 필요함 - 요구조건에 신속히 환경을 변경 가능할 필요가 있음	

<표 9> Test Education에서 적용결과

Test Education	문서번호	No.6
	단계	TTRN, TMEN
분석	- 테스트에 대해 전체적인 교육이 수행되고 있음 - 업무에 대해 기능과 기술을 분류하고 있음	
개선	- 품질보증을 위한 정형적인 훈련이 필요함 - 지속적인 향상을 위해 테스트 프로세스에서 적절히 운영해야함	

<표 10> Test Organization 에서의 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	- 응용개발이 신속히 이루어져야함 - 테스트 조직 내에서 제품 품질을 위한 의사 전달이 이루어지지 않음 - 매트릭스를 통해 고객에게 위협과 권고를 전달해야함 - 테스트 케이스를 통해 결함을 추적할 수 있는 체제가 필요함 - 프로세스의 진행에 벗어나는 행위에 대한 문서화 작업이 필요함	

5. 결론

높은 품질의 소프트웨어를 완성하는데 있어 개발 모델의 적용만큼 소프트웨어 평가모델 적용 또한 많은 비중을 차지하고 있다. 우리는 소프트웨어 평가모델을 각각 비교하

였다. 그 중 테스트 조직의 성숙도 수준을 평가할 수 있는 테스트 성숙도 모델이 갖고 있는 문제점을 극복하기 위해 테스트 프로세스 관점의 평가모델인 테스트 프로세스 개선 모델을 접목하였다. 또한 테스트 속성과 성숙도 상호관련 매트릭스를 만들고 각각의 속성들을 제안하였습니다. 이를 통해 정량적인 조직의 테스트 성숙도 수준을 평가할 뿐만 아니라 테스트 프로세스도 함께 고려하여 평가할 수 있었다. 또한 테스트 프로세스 개선을 통해 개발 혹은 테스트 조직이 높은 테스트 수준으로 향상시키기 위한 개선 제안을 제시할 수 있었다.

향후 테스트 인증 모델인 GS 품질평가 모델과 확장된 TMM과의 접목을 통해 테스트 인증뿐만 아니라 테스트 조직의 향상을 위한 개선 가이드를 제안하려고 한다.

참 고 문 헌

[1] I. Burnstein, A. Horny, T. Suwannasart, G. Saxena, and R. Grom, "A Testing Maturity Model for Software Test Process Assessment and Improvement," Software Quality Professional, 1999.

[2] Ilene Burnstein, Taratip Suwannasart, and C.R. Carlson, Developing a Testing Maturity Model: Part I.

[3] Ilene Burnstein, Taratip Suwannasart, and C.R. Carlson, Developing a Testing Maturity Model: Part II.

[4] Tim Koomen, Martin pol, "Test Process Improvement," Addison-Wesley 1999 pp 2~53.

[5] I. Burnstein, "Practical Software Testing," Springer, 2003.

[6] 김기두, 김영철, "테스트 프로세스 개선(TPI)을 통한 테스트 성숙도 모델(TMM) 개선에 관한 연구, 홍익대, 2004.

[7] 김기두, 김영철, "테스트 프로세스 향상을 통한 테스트 성숙도 모델 개선에 관한 연구," 한국소프트웨어 공학 학술대회 논문집, 제1권, 제1호, 2005년 2월.

[8] 김기두, 류동국, 김영철, "TMM 향상을 위한 테스트 프로세스 성숙도 체크리스트 연구," 정보과학회 추계학술대회 논문집, 2004.

[9] Yuri Chernak, Ph.D, "Introducing TPAM: Test Process for Software Test Documentation)," Vally Forge Consulting Inc, 1988.

[10] "IEEE Standard for Software Test Documentation," IEEE std.829-1988, 1988.

[11] "IEEE Standard for Software Verification and Validation," IEEE std 1012-1988, 1998.

[12] "소프트웨어 테스트 전문기술", 한국정보통신기술협회, 2004

김 기 두



e-mail : kdkim@tta.or.kr

2003년 홍익대학교 컴퓨터정보통신과 (공학사)

2005년 홍익대학교 일반대학원 소프트웨어공학전공 (석사)

2005년 홍익대학교 일반대학원 소프트웨어공학전공 박사과정

2004년~현재 한국정보통신기술협회 SW시험인증연구소 연구원
 관심분야: 테스트 성숙도 모델(TMM), 테스트 프로세스, 임베디드 소프트웨어 테스트, 소프트

김 영 철



e-mail : bob@hongik.ac.kr

2000년 Illinois Institute of Technology (박사)

2000년~2001년 LG 산전 중앙연구소 Embedded system 부장

2001년~현재 홍익대학교 컴퓨터정보통신 교수

관심분야: 소프트웨어 성숙도 모델, Use Case 방법론 및 도구 개발, 테스트 성숙도 모델(TMM), 컴포넌트 시험 및 평가, 임베디드 소프트웨어 공학