韓國IT서비스學會誌
第8卷 第3號
2009年 9月

TMMi 기반 자동차 제어 SW 테스팅 프로세스 구축에 관한 연구*

장 진 욱**

A Study on Implementation of Automobile Control Software Testing Process Based on TMMi*

Jin-Wook Jang**

■ Abstract ■

The issue of method of software testing process implementation has recently been in the spotlight in korea and some vendor make every effort to improve the software testing process through the evaluation of TMMi. The level of software process is at initial level, nevertheless, so the high quality of software is not guaranteed. This paper apply to the TMMi Level2 Assessment criteria of Automobile control software testing process. The test policy and strategy, test planning, test monitoring and control, test design and execution, test environment goal.

The result need to emerge development process connection on the test policy and strategy process, test planning process etc. Also, the study analyzed the infra structure to reach the repeated level via key process area. As the result, the process implementation in organization suggests the policy development and the implementation activity. Finally, the practical can refer to this paper in order to implementation the software testing process.

Keyword : Software Testing Process, TMMi Assessment, Risk based, Test Policy and Strategy, Test Planning

논문투고일 : 2009년 06월 16일 논문수정완료일 : 2009년 08월 06일 논문게재확정일 : 2009년 09월 12일
* 2008년 한국소프트웨어진흥원 "소프트웨어 역량구축 사업" 수행 사례 활용.
** (주)STA컨설팅 책임, 건국대학교 첨단기술시스템전공 박사과정 수료

1. 서 론

소프트웨어 규모와 복잡도가 증가함에 따라 최근 소프트웨어 업계는 제품의 신뢰성 및 생산성과 관련하여 품질에 대한 관심이 높아져가고 있다. 품질에 관한 관심은 ISO/IEC 29119, BSI(British Standards Institution)와 IEEE 같은 표준을 중심으로 하는 소프트웨어 테스트 활동으로부터 시작되어, 소프트웨어 테스트 프로세스의 중요성으로 관심이 확대되고 있다.

소프트웨어 테스트 프로세스 진단 및 개선 모델을 TMMi(Test Maturity Model Integration),[1] TPI(Test Process Improvement), TOM(Test Organization Maturity Model), TIM(Test Improvement Model) 등 다양한 평가 및 개선이 정의되고 있다. 이들 개념은 지금 현재도 발전 중에 있으며 국내외 소프트웨어 관련 산업체에서도 관심이 높아지고 있다. 그러나 소프트웨어 테스팅 프로세스 준수가 비용 감소, 일정 준수, 재사용율 증가와 같은 경제적 효과로 즉시 가시화되지 않기 때문에 조직은 보다 적극적인 프로세스 개선 활동을 추구해야 할 필요가 있다[8].

소프트웨어의 품질확보를 위하여 짧은 라이프 사이클, Time to market 등의 이유로 테스트를 생략하거나 형식적으로 진행되고 있는 현실에서 제한된 자원을 최대한 활용하여 소프트웨어 테스팅 프로세스를 구축하고, 개발단계에 도구 및 방법론 등 체계적인 테스팅 기법을 적용함으로써 소프트웨어 테스팅의 성숙도를 높일 수 있다.

따라서 본 논문에서는 자동차 제어 소프트웨어에 테스팅 프로세스를 적용하기 위하여 테스트 프로세스 개선 모델에 대한 일반적인 개념과 표준을 제시 한다. 이러한 성숙도 모델을 바탕으로 자동차 제어 소프트웨어에 테스팅 프로세스 적용에 필요한 핵심영역을 도출하고 체크포인트를 이용하여 각 핵심영역별 테스트 프로세스를 평가 하였다. 테스트 프로세스의 평가 결과에 따라 핵심영역별 테스트 프로세스의 장·단점을 분석하며 이를 통해 각 영역별로 테스팅 프로세스 구축방안을 제안한다.

1) 비영리 단체 "TMMi Foundation"에서 최근 Level 3공개(www.tmmifoundation.org).

2. 관련연구

2.1 테스팅 프로세스 진단 모델

테스트 프로세스 모델 중 진단 지향적인 모델로는 대표적으로 TMMi(Test Maturity Model Integration)과 TPI(Test Process Improvement)를 들 수 있다. TMMi과 TPI는 테스트 수행 조직의 테스트 프로세스 성숙도 평가에 널리 사용되고 있어 그 유용성이 이미 여러 형태로 입증되어 있다.

두 모델은 테스트 프로세스 진단의 여러 가지 측면에서 유사하지만 <표 1>과 같이 몇 가지 차이점이 있어 평가 모델 선정 시 참고하면 테스트 프로세스 평가 및 개선을 보다 효과적으로 수행 할 수 있다[1].

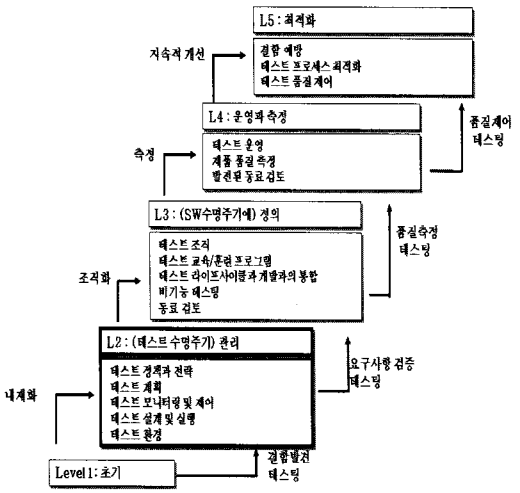
〈표 1〉 TPI 모델과 TMMi 비교

구 분	TPI	TMMi
Type	연속적 모델	단계적 모델
Test Methods	TMap[2]과 연관	테스트 방법과 독립적
SPI	SPI모델과 공식적인 관련 없음	CMMI와 높은 관련
Test Levels	특히 높은 레벨을 지원	모든 테스트 레벨
Focus	20핵심 영역	각 성숙도 레벨마다 프로세스 영역의 제한된 수에 상세한 초점을 맞춤
Approach	테스트 엔지니어링	경영층의 노력에 강하게 집중

TMMi은 개발 능력 성숙도 모델(CMMI)과 관련된 효과적인 테스트 프로세스 핵심요소를 기술하

2) TMap : Test Management Approach.

는 5레벨의 단계적(Staged)평가 프레임워크로 테스트 프로세스 개선을 위해 사용된다[9]. TMMi은 [그림 1]과 같이 시스템 개발 프로세스 진단 모델인 CMMI(Capability Maturity Model Integration)의 단계적 모델(Staged model)과 유사한 구조를 가지고 있다.



[그림 1] TMMi 모델

TMMi 모델은 위의 그림처럼 효과적인 테스트 프로세스의 핵심요소를 5단계(Level)로 구분해 놓았다. 각 단계별 세부 내용은 다음과 같다.

① Level 1 : 초기(Initial)

레벨 1은 테스팅은 정의되지 않거나, 테스팅과 디버깅이 한 부분으로 인식되고, 조직은 일반적으로 프로세스를 지원하기 위한 안정적인 환경을 제공하지 못한다. 이러한 조직의 안정은 조직인력의 능력과 자신감에 의존적이다. 이런 활동은 검증된 프로세스의 활용이 아니다. 테스트는 구현이 완료되고 ad-hoc한 방법으로 이루어지며 테스팅과 디버깅은 시스템에서 버그를 제거하기 위하여 이루어진다.

② Level 2 : (테스트수명주기)관리(Managed)

레벨 2는 테스트와 디버깅이 구분되며 테스트가 소프트웨어 생명주기에서 하나의 독립된 단계로 정의되고, 결함 발견 활동에 집중한다. 상대적으로 중요하고, 의미 있는 레벨인데, 이를 달성하려면 테스트 정책(Policy)을 별도로 문서로 두거나 품질정책이나 개발정책의 일부에 정의 하고 있어야 한다. 또한 테스트 목표를 정확히 하면서 이를 반영하여 테스트 전략 또는 접근법(Test Strategy or Approach)에 근거하여 테스팅하고 있다는 것을 보여줘야 한다. 테스트 계획을 완성도 높게 수립하고 이에 따라 테스팅이 진행되어야 하며, 테스트 규모 산정(Test Estimation)도 현실적으로 설득력 있게 작성 되어야 한다. 그리고 테스트 케이스 작성을 위한 테스트 설계기법도 리스크를 반영하여 리스크 레벨에 따라 차별적으로 적용되고 있어야 한다. 이 밖에도 문서화된 절차에 따라 테스트 환경을 확보 및 관리하고 있어야 한다.

③ Level 3 : (SW수명주기에)정의(Defined)

레벨 3은 테스팅이 개발 생명 주기와 통합되는 단계로, 레벨 2에서 테스팅을 포함하고 개발되었는지 검증하는 테스트 활동을 수행해야 한다. 이 레벨을 달성하기 위해서는 테스트 프로세스가 소프트웨어 개발 수명주기에 통합되어 있어야 한다. 또 별도의 테스팅 조직을 갖추고 있어야 하며, 그렇지 않는 경우 테스팅 조직을 독립적으로 유지하지 않는 명확하고 타당한 이유를 명시하고 있어야 한다. 레벨 2가 프로젝트 레벨에서 내재화하는 수준이라면 레벨 3은 조직차원에서 테스팅을 내재화하는 수준이다.

④ Level 4 : 운영과 측정(Management and Measurement)

레벨 4 달성을 위해서는 '발전된 동료검토' 활동이 수행되고 있어야 하는데, 이는 테스팅 시각에서 볼 때 테스트케이스를 요구사항 분석 단계부터 설계하고 작성하는 것을 통해 개발 중간 산출물의 결함을 조기에 발견하는 예방적인 테스팅을 의미한다. 테스트를 관리하고 측정하는 단계로 소프트

웨어 품질 평가와 메트릭(Metric)을 이용한 테스트 측정을 통해 테스트를 수치화(측정)하고, 이를 기반으로 정량적으로 테스트를 관리하고 있어야 달성이 가능하다. 개발 제품뿐만 아니라 테스트 관련 문서 등도 검토의 대상이며, 검토 활동을 측정의 필수요소로 인식한다.

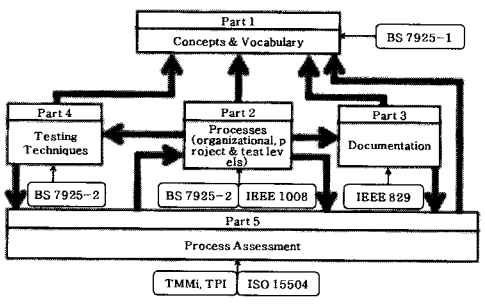
⑤ Level 5 : 최적화(Optimization)

결함예방과 품질제어 활동에 초점, 테스트 프로세스가 정의되고 관리되며, 비용과 효과가 추적되고 감시된다. 테스트 프로세스가 지속적으로 개선되고 조정되며, 결함예방과 품질제어 활동이 수행된다. 이들 활동들이 통계적 방법과 다양한 평가 기준에 의해 측정되고, 관리자는 지속적인 개선을 유도하기 위해 인프라를 지원하고 동기를 부여한다.

이와 같이, TMMi 모델은 완성도가 높은 모델로서 높은 일관성, 완전한 성숙도 모델 구조의 장점을 지닌다.

2.2 ISO/IEC 29119 테스팅 표준 체계

ISO/IEC 29119 소프트웨어 테스팅 표준이 현재 국제 표준화 작업 그룹에서 만들어지고 있는데 그 내용은 크게 [그림 2]와 같이 구성 된다.



[그림 2] ISO/IEC 29119 범위

Part1, Part2, Part3, Part4의 세부내용은 <표 2>와 같다, Part5는 신규 제안을 통한 표준화가 필요한 부분이다.

〈표 2〉 소프트웨어 테스팅 표준 구성

구 분	내 용
Part 1 : 개요와 용어	◦ 소프트웨어 테스팅 정의 및 소개 ◦ 테스팅, 개발, 유지보수 사이의 관계 정립
Part 2 : 프로세스	◦ 조직 차원의 테스트 정책 프로세스 ◦ 조직 차원의 테스트 전략 프로세스 ◦ 프로젝트 테스트 관리 프로세스 ◦ 테스트 레벨 프로세스
Part 3 : 문서화	◦ 테스트 관리 문서화(테스트 정책, 전략, 프로젝트 완료 보고) ◦ 테스트 문서(테스트 계획, 명세, 결과, 결함보고, 테스트 레벨 완료 보고) ◦ 테스트 상태 리포팅 (테스트 레벨 상태 리포트, 테스트 환경 리포트)
Part 4 : 테스팅 기법	◦ 테스트 케이스 설계 기법 (정적, 동적, 비기능 테스트 기법) ◦ 테스트 측정(커버리지)

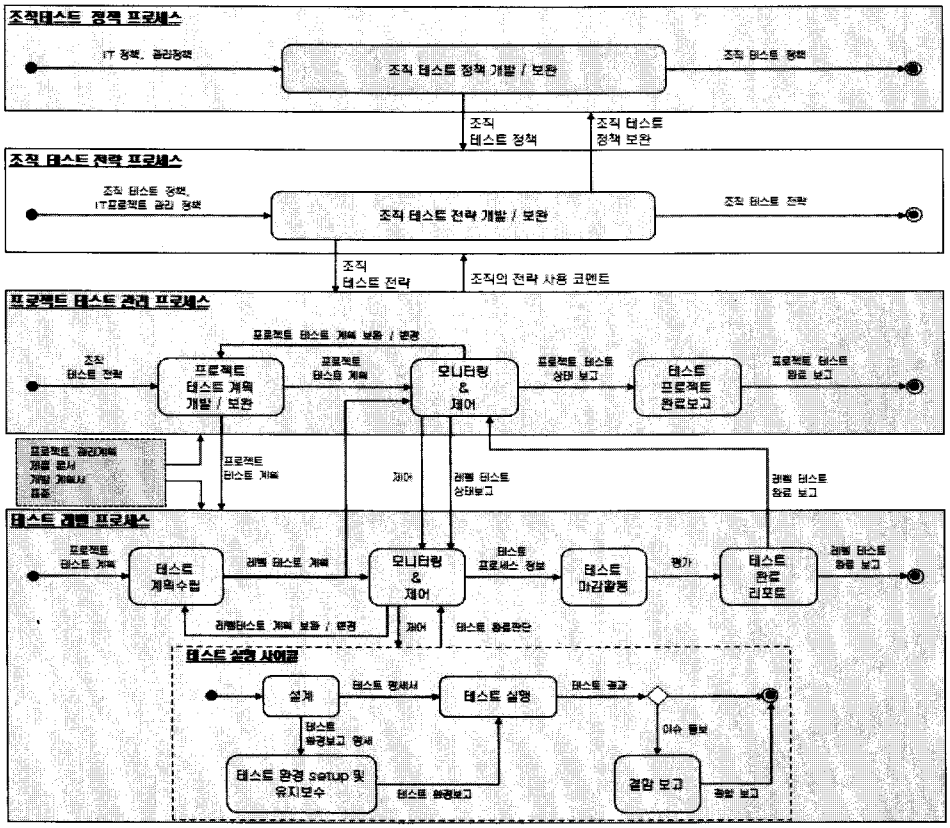
[그림 3]은 위에서 설명한 각각의 프로세스들 사이의 관계를 4계층 모델(4-Layer Model)로 구조적으로 도식화하여 보여주고 있는데, 조직 차원에서의 테스트 정책 프로세스와 테스트 전략 프로세스, 프로젝트 차원에서의 프로젝트 테스트 관리 프로세스, 테스트 레벨 차원에서의 테스트 레벨 프로세스로 구성되어 있다. 그림에서 볼 수 있듯이 각 프로세스들은 서로 유기적으로 상호 연관 관계를 가지고 있다.

프로젝트 테스트 계획의 전략 수립 시에는 전사적 테스트 전략이 반드시 참고 반영되어야 한다. 또한 각 레벨 테스트는 상위의 프로젝트 테스트 관리 활동과 유기적으로 연계 되어 피드백을 주고받아야 한다[2].

3. 연구방법

3.1 모델 및 방법

본 연구의 목적은 소프트웨어 테스팅 프로세스 개선활동에 대한 인식 전환과 체계적인 소프트웨어 테스팅 프로세스 개선방안을 제시하기 위한 것이다. 따라서 규모와 유형에 따른 소프트웨어



[그림 3] ISO/IEC 29119 4-Layer Model

기업들의 소프트웨어 테스팅 프로세스 능력수준을 파악하고 분석하는 것이 필요하며, 이를 목적으로 TMMi 기반 체크리스트를 사용 하였다.

체크리스트는 TMMi Level2 달성을 목표로 하는 구조화된 질문서로서, 각 KA (Key Area) 또는 목표(Goal)별로 5가지 핵심영역을(테스트 정책 및 전략, 테스트 계획, 테스트모니터링 및 통제, 테스트 설계 및 실행, 테스트환경)을 반영하고 있다.

각 질문에 대한 응답 결과는 현장 심사에서의 인터뷰를 위한 중요한 정보가 되므로, 성숙도 질의 응답자를 선정할 때는 4~10명의 응답자를 참가시키되, 프로젝트 리더는 전원 참가할 수 있도록 한다. 설문서에 대한 응답은 예/아니요 외에도 설문자의 코멘트를 기록 하도록 하고, 응답자가 설문서의 질문을 충분히 이해할 수 있도록 일반적인 어휘를 사용하고 필요한 경우 용어집을 제공한다.

<표 3>은 질문조사 핵심영역이다.

〈표 3〉 질문조사 영역 및 항목

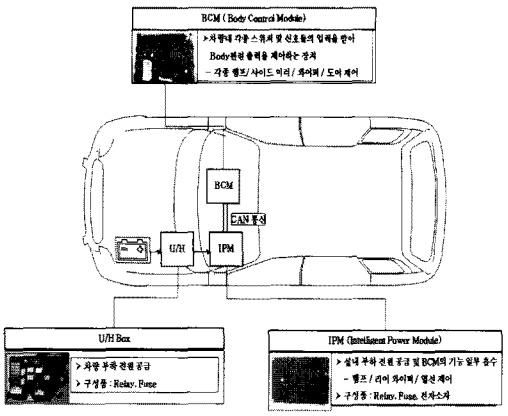
핵심 영역	체크 포인트
테스트 정책 및 전략 수립 프로세스(KA 1)	테스트 목적에 따른 정책, 전략 수립, 능력지표 등 21개 항목
테스트 계획 프로세스 (KA 2)	테스트 대상목록, 접근방식, 추정, 계획수립검토 등 19개 항목
테스트 모니터링 및 통제 프로세스(KA 3)	테스트 진척도 모니터링, 계획 대비 product Quality 모니터 링, 수행관리 등 11개 항목
테스트 설계 및 실행 (KA 4)	테스트 분석설계, 개발, 실행, 인시던트3) 등 18개 항목
테스트 환경 프로세스 (KA 5)	테스트 환경 요구사항 분석, 환경구축, 환경관리 등 10개 항목

3) 인시던트(Incident) : 기대결과와의 차이.

질의조사 영역 및 항목 등을 적용함으로서 테스팅 조직의 성숙도와 업무역량 향상을 측정하고 개선 포인터를 도출한다.

4. 적용 및 평가

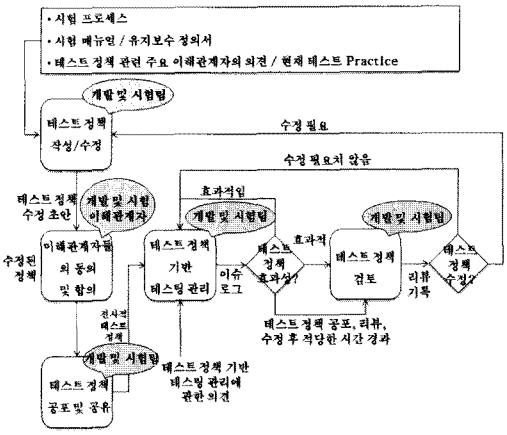
본 논문에서는 소프트웨어 테스팅 프로세스 와 테스팅 기법 등을 적용할 분야는 자동차 제어 소프트웨어 중 [그림 4] IPM[4]으로 차량 내부 전원을 필요로 하는 서브시스템 및 모듈에 대해 부하 제어 및 전원 분배 공급 기능을 하는 U/H(Under Hood) Box와 차량용 제어기 통신 모듈인 BCM (Body Control Module)이 CAN(Controller Area Network)방식 통신을 이용하여 고장 진단 및 데이터 전송을 하는 제어 소프트웨어 이다.



[그림 4] IPM 제품 장착 위치

4.1 테스트 정책 수립

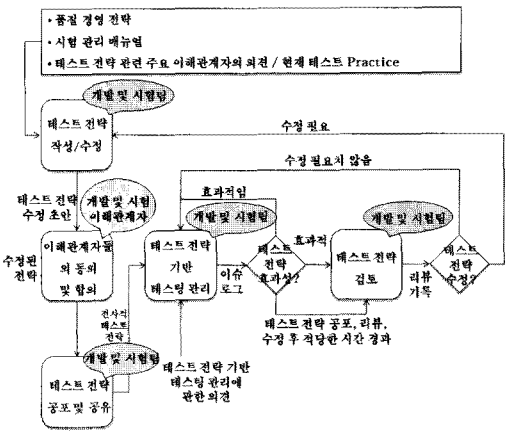
ISO/IEC 29119 4-Layer Model 중 테스트 정책은 조직 내 소프트웨어 테스팅의 범위를 정의하고 따라야 할 테스팅 관련 프랙티스(Practice)와 규칙을 확립하기 위함이다. 조직 내 이해관계자가 테스트 정책기반의 테스팅 관리를 할 수 있도록 기본 틀을 제시함으로서 지속적인 개선이 이루어 질 수 있다. [그림 5]와 같이 주요 관계자들의 테스트 정책 리뷰 후 피드백을 반영하여 합의를 도출한다.



[그림 5] 테스트 정책 프로세스

4.2 테스트 전략 수립

ISO/IEC 29119 4-Layer Model 중 테스트 전략 수립을 통하여 조직 내 이해관계자가 테스트 전략 기반의 테스팅 관리를 할 수 있도록 기본 틀을 제시한다. [그림 6]과 같이 주요관계자들의 테스트 전략 리뷰 후 피드백을 반영하여 합의를 도출 한다. 도출된 테스트 전략은 조직 내의 모든 테스트 업무에 적용되는 기본 근거 이다.

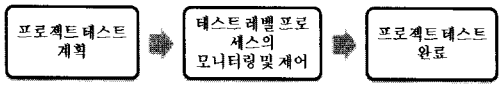


[그림 6] 테스트 전략 프로세스

4) IPM(Intelligent Power Module) : 차량의 전기장치를 제어할 수 있도록 마이크로 컨트롤러를 내장한 퓨즈박스.

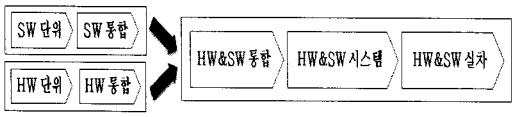
4.3 프로젝트 테스트 관리 프로세스 수립

조직 내에 레벨 테스트가 하나만 있는 경우도 있겠지만 일반적으로 두 개 이상일 경우가 많을 것이다. 이러한 레벨 테스트들 사이의 관계를 프로젝트 레벨에서 보다 큰 시각으로 정의하고 관리하기 위한 목적으로 프로젝트 테스트 프로세스를 수립한다. 프로젝트 테스트 관리 프로세스는 [그림 7]에서와 같이 프로젝트 테스트 계획부터 테스트 레벨 전반에 대한 모니터링 및 제어 활동을 포함하여 프로젝트 테스트 완료 보고까지 전체 프로젝트 테스트를 총괄하는 프로세스이다.



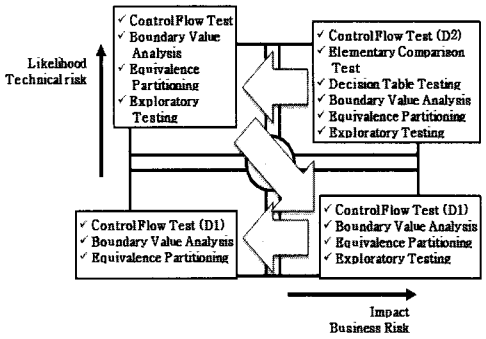
[그림 7] 프로젝트 테스트 관리 프로세스

본 논문에서는 자동차 제어 소프트웨어 프로젝트 테스트의 테스트 레벨을 [그림 8]을 각각 HW 단위, HW 통합, SW 단위, SW 통합, HW&SW 통합, HW&SW 시스템, HW&SW 실차 테스트로 정의 하였다.



[그림 8] IPM 프로젝트 테스트 레벨

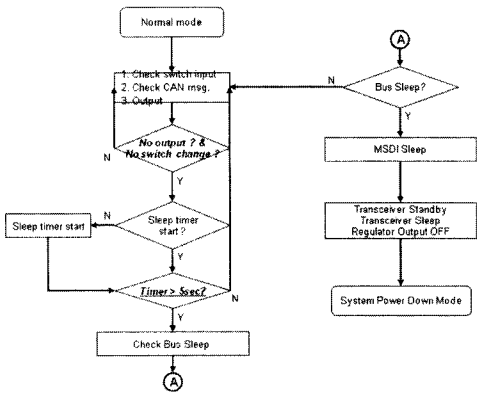
테스팅의 리스크 기반 접근을 위하여 프로젝트 관련자들 간의 합의 도출을 위하여 리스크 팩터와 리스크 아이템을 식별하고 이후 각 관련자들의 리스크 점수화 및 취합 과정을 적용 하였다. 리스크 우선순위에 따라 [그림 9]와 같이 리스크 영역별로 각기 다른 테스트 설계 기법 및 완료 조건을 적용하는 과정을 통해 프로젝트에 리스크 기반 전략을 수립 한다. 적용하는 설계 기법은 리스크에 따라 차별적으로 기법을 적용하고 리스크가 높은 영역은 좀 더 강력한 기법의 적용이 필요하다[6].



[그림 9] 리스크 기반 테스트 설계 기법 적용

4.4 테스트케이스 설계

IPM 프로젝트의 Power Down Mode 모듈 중에서 배터리의 방전을 방지하기 위해 가장 중요한 기능인 Sleep Mode 진입 Flow 부분을 대상으로 적용 하였다.



[그림 10] Sleep Mode 진입 Flow

MCU에서 Sleep Mode 진입은 MCU와 4개의 모듈(Switch, CAN, Output, Timer로 간단히 표기함)과의 통신 결과에 따라 결정된다. 이를 효과적으로 테스트하기 위해 MC/DC(Modified Condition/Decision Coverage)를 적용하였다[5]. 이를 기반으로 <표 4>와 같이 상위 설계의 논리적 테스트케이스를 도출 하였다. MC/DC 커버리지 특성상 두 개 이상의 Switch 값이 동시에 변환이 이루어지는 경우를 고려하지 못하므로 조합(Pairwise) 테

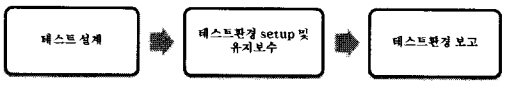
스팅을 통하여 추가 보완 하였다[3].

〈표 4〉 MC/DC 커버리지 적용 및 보완

TS[5]	Switch	CAN	Out put	Timer	기대 결과	비 고
1	1	1	1	1	Sleep	
2	0	1	1	1	Normal	
3	1	0	1	1	Normal	
4	1	1	0	1	Normal	
5	1	1	1	0	Normal	
6	1	1	1	0	Normal	기법 보완
7	1	1	1	1	Normal	

4.5 테스트 환경

테스트 설계를 수행하기 위해 필요한 하드웨어, 계측기(Instrumenter), 시뮬레이터, 소프트웨어 툴, 그 밖의 지원 요소를 구성하는 테스트환경 프로세스를 [그림 11]과 같이 구축 하였다.

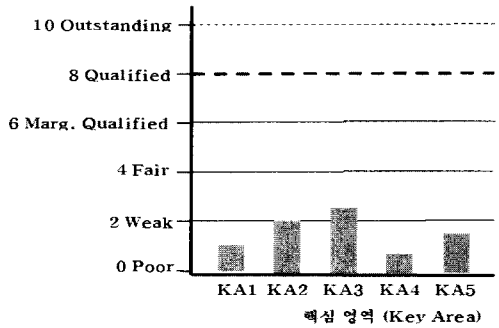


[그림 11] 테스트 환경 프로세스

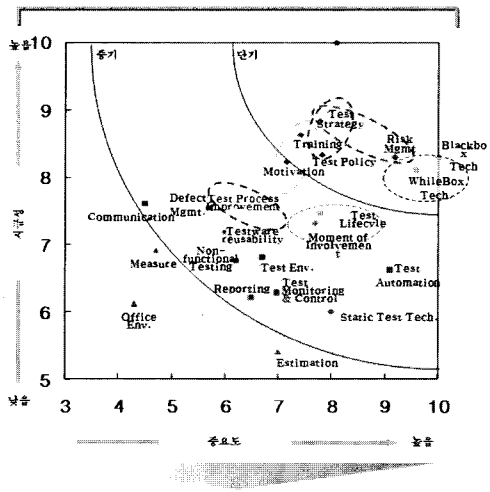
4.6 TMMi 진단을 통한 개선

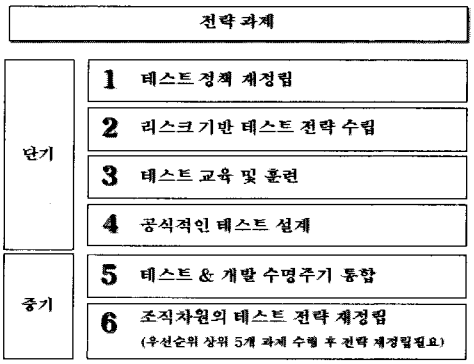
TMMi Level 2를 기준으로 진단결과 [그림 12]와 같이 핵심영역별 개선방향을 도출 하였다. 테스트 정책 및 전략 수립 프로세스(KA 1), 테스트 계획 프로세스(KA 2) 등, 전체적인 테스트 프로세스에 대한 정립과 소프트웨어 개발 프로세스와의 연계가 필요하다.

TMMi 진단 결과를 바탕으로 테스팅 관련 과제별 시급성과 중요성 또는 파급효과를 분석하여 개선 우선순위를 정의 하였다. [그림 13]과 같이 단기 우선 과제로는 자동차 제어 소프트웨어 테스팅 정책 및 리스크 기반 테스팅 전략을 수립하여 테스트 목적을 명확히 하고, 테스트 교육 및 훈련을 통한 역량향상과 공식적 테스트 설계 기법 적용을 도출하였다.



[그림 12] TMMi 진단 결과





[그림 13] 개선과제 도출

도출된 개선과제를 기반으로 조직에 내재화를 위하여 다음과 같은 영역별 개선이 필요하다.

5) TS(Test Suite) : 테스트케이스의 집합.

① 테스트 정책 및 전략 수립 프로세스

테스트 정책 및 전략 수립에 대한 기본적인 틀이 마련되었다. 그러나 명확한 이해와 상호관계에 대한 이해가 부족하고 정책과 전략 이후에 계획 및 수행단계까지 연계가 필요하다.

② 테스트 계획 프로세스

위험분석이 프로덕트의 기능 위주뿐만 아니라 프로젝트 위험도를 분석하여 계획에 반영되어야 하며 우선순위 화된 위험에 기반을 두어 테스트 항목을 선정하고 테스트 추정에 어떻게 적용되는지 정의되어야 한다. 전체 테스트 프로세스의 라이프 사이클이 정의되어 있지 않아 라이프 사이클에 맞추어 계획되었다고 보이지 않고, WBS(Work Break down Structure)상에서 Test Execution Phase가 빠져 있다, 각각의 단계(Phase)에 대한 자세한 활동이 정의 되어 있지 않고 그 활동 대한 추정이 이루어 지지 않았다. 추가적으로 테스트계획이 모든 이해당사자에 의해 리뷰 한 로그의 문서화 필요가 있다[4].

③ 테스트 모니터링 및 통제 프로세스

테스트 일정 및 의사소통 관리를 위한 테스트 진척 보고의 실행이 필요하다. 예컨대 프로젝트 계획서, 테스트전략, 상태리포트, 도구적용개선 보고서 등의 최종 보고, 그리고 테스트 활동에 대한 리뷰를 통하여 전체 테스팅 활동에 대한 모니터링 및 통제 활동이 주기적으로 필요하다.

④ 테스트 설계 및 실행

레벨 별로 테스트 설계 및 실행이 되지 않고, 특정 레벨테스트 부분에만 한정되어서 수행되었다. 요구사항과 테스트 케이스 간의 추적성에 관한 내용이 구체적인 요구사항과의 매핑이 필요하다.

⑤ 테스트 환경 프로세스

테스트 환경에 대한 요구사항을 반영 하여 환경을 구축하는 부분은 통합테스트 한정되어 문서화 되어 있으나 이 부분도 구체적화가 필요하다.

TMMi 기반 심사의 한계점으로는 프로세스별로 성숙도가 나타나지 않고 조직 전체의 성숙도만 표현 된다는 점이다. 실제 테스팅 조직에서는 각각의 테스트 프로세스 별로 성숙도가 다를 수 있다. 또한 TMMi 심사를 위한 세부 항목(체크리스트)은 공개되어 있는 것이 제한적이고 전문가마다 각자의 체크리스트 및 방법으로 평가가 이루어지고 있어 표준화가 요구된다. 현재 이러한 문제점을 해결하고자 진단(심사) 방법 및 평가항목, 평가 절차의 표준화 및 선임 심사원의 양성 및 자격관리가 비영리 단체인 TMMi Foundation을 중심으로 이뤄지고 있다[7].

5. 결 론

본 논문에서 소프트웨어 테스팅 프로세스는 개발 과정과 밀접한 연관을 맺고 생산성에 직접 영향을 미치기 때문에 다른 어떤 프로세스보다도 개발 과정에 바로 쉽게 적용할 수 있는 실용적인 프로세스이어야 한다. 소프트웨어 테스팅 현황이 진단되어 실무에 적합한 테스팅 프로세스 개선안을 도출하여 우선순위에 따라 단기, 중기 계획을 제시하고 이를 수행함으로써, 소프트웨어 테스팅 프로세스를 구축하고, 소프트웨어 테스팅 관련자들이 국제 표준 기반의 전문적인 소프트웨어 테스팅 지식 및 기술을 실무에 적용함으로서 소프트웨어 품질에 대한 정량적 평가 기반을 확보하여 제품의 품질과 신뢰성 향상에 기여 할 수 있다.

조직은 테스트 프로세스 진단 결과를 바탕으로 우선순위를 고려하여 단기 과제를 중심으로 우선 내재화 노력과 중기 과제를 기본 프로세스 발전 방향으로 채택하여 차기 프로세스 개선 계획에 활용한다면 발전된 조직으로 향상 될 수 있을 것이다. 본 논문에서는 품질개선을 위한 실용 프로세스를 실무에 적용 및 구축방안을 제시하였다.

참 고 문 헌

[1] Erik Van Veenendaal, "Test Maturity Model Integration(TMMi)", Version 1.0 Produced by The TMMi Foundation, 2008.

[2] ISO/IEC 29119 Test Process, 2008.

[3] ISTQB, Certified Tester Advenced Level Syllabus, 2007.

[4] ISTQB, Certified Tester Foundation Level Syllabus, 2007.

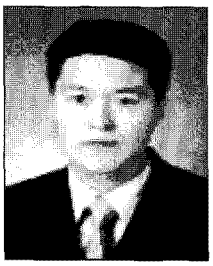
[5] Arnaud Dupuy, Alcatel, France, Nancy Leveson, MIT, USA, An Empirical Evaluation of MC/DC Coverage Criterion on the HET E-2 Satellite Software, 2001.

[6] James Bach, "James Bach on Risk-Based Testing", 1999.

[7] 권원일, 이준섭, 「개발자도 알아야할 소프트웨어 테스팅」, 2007.

[8] 오기성, 이남용, 류성열 "Software Testing", 글로벌, 2003.

◈ 저 자 소 개 ◈



장 진 욱 (jwjang@sta.co.kr)

육군전산장교를 거쳐 현재 소프트웨어 테스팅 컨설팅사인 (주)STA컨설팅 책임 컨설턴트로 재직 중이며, 육군 3사관학교 전산학 학사, 숭실대학교 컴퓨터 공학석사를 마치고 건국대학교 첨단기술시스템전공 박사과정을 수료 하였다, 한국IT서비스학회, 인터넷정보학회, 임베디드월드 등의 학술지 및 저널에 논문 및 기사를 게재하였으며 주요 관심분야로 국방정보화 정책, SW 테스팅 기법, SW 테스팅 프로세스, SW 자동화 테스팅, SW 프로세스 심사모델, SW 테스팅 컨설팅, 테스팅 조직관리 등이다.