

■ 사례연구

TDX-10 ISDN 시스템의 시험 및 평가분석 체계화에 따른 품질보증 활동

이해룡 · 정택원

한국전자통신연구소

Systematic Quality Assurance Activity for TDX-10 ISDN Switching System

Haeryong Lee · Taegwon Jeong

Electronics and Telecommunications Research Institute

요약

대용량 ISDN(Integrated Services Digital Network)용 전전자 교환기 TDX-10 연구 개발 사업의 궁극적 목적은 미래 통신망이 요구하는 다양한 음성 및 비음성 정보통신 서비스를 제공하고 정보의 교환, 가공, 축적 및 처리에 고도의 융통성을 부여할 수 있는 국내 표준 전전자 교환기를 개발하여 통신투자의 경제적 통신운영기술 자립화를 달성하는데 있다. 현재 대용량 전전자 교환기인 TDX-10 ISDN의 상용화 단계의 마무리 업무가 진행중이며, 통신서비스의 다양화, 광대역화, 고속화를 위한 광대역통신망의 주축이 되는 ATM교환기가 개발중에 있다. 본 고에서는 TDX-10 ISDN 시스템개발단계에서 구현된 시스템 시험수행 과정 및 평가작업을 토대로한 체계적이며 전략적인 품질보증 활동에 관해 기술하였으며, 신뢰성 있는 시험결과를 도출할 수 있는 다양한 시험관리 기법과 그 결과를 분석하고 차후 시험 및 보완작업을 위한 제반 활동체계가 소개된다.

1. 서론

TDX-10 ISDN 시스템은 32비트 마이크로 프로세서(MC68030)를 사용하는 분산제어 프로세서 시스템으로 상, 하위 2계의 계층구조를 갖는 프로세서와 이 프로세서의 제어하에 교환기능을 수행하는 디바이스로 구성되어 있다. 각각의 프로세서에는 교환기의 제반 기능을 수행하는 소프트웨어들이 실행모듈 형태로 실장되며, 이들 소프트웨어는 대부분 교환기용 언어인 CHILL(CCITT High Level Language)로 구현되어 있고, 단말기와 의 정합 부분과 같은 특정한 부분은 MC68020 어셈블리 및 C 언어를 사용하였다.

1.1 주요 기능

• 운영체제(Operating System)

TDX-10 ISDN OS는 프로세스 관리기, IPC(Inter Processor Communication) 관리기, 시간 관리기, 메모리 관리기, 예외 관리기, 입출력 관리기, 화일 관리기등 교환기능에 필요한 프로그램의 효율적인 수행을 위한 기능을 제공해 주는 운영체제로 상위레벨 CROS(Concurrent Real Time Operating System)와 하위레벨인 PPOS(Peripheral Processor Operating System), IPOS(Inter Processor Operating System), SPOS(Signaling Processor Operating System), 그외에 패킷 디바이스의 PHOS(Packet Handling Module Operating System) 등으로 구성된다.

• DBMS(Data Base Management System)

DBMS는 모든 응용 프로그램들이 데이터를 공유할 수 있도록 여러 응용 프로그램이 사용하는 데이터를 한곳에 모아 종합적으로 유지하고 관리해주는 소프트웨어로 TDX-10 ISDN 시스템의 내부 동작형태로 변경되어 주기억 장치에 상주하는 데이터들을 관리하고, 응용 프로그램이 요구하는 데이터의 검색 및 변경 요구를 처리해 주는 기능을 수행한다.

• 사용자 정합 기능

TDX-10과 운용자의 정합기능은 입출력형태, 대화절차, 상호작용등을 규정하여 운용자와 교환기 시스템 사이의 대화창구 및 수단을 제공해주는 소프트웨어로 본체 및 운용자 단말기(PC)상에 나누어 구현되어 있다.

• 호처리 기능

호처리 소프트웨어는 호의 발생 부터 종료까지 발생하는 모든 상태를 감시, 제어하여 정상적인 통화가 성립되도록 하는 기능을 담당한다. 호의 종류로는 일반 음성호, ISDN 호, Packet 호 등이 있으며 각각에 대한 부가서비스들이 존재한다. 호처리기능은 시스템내의 제한된 자원으로 많은 가입자를 동시에 실시간으로 처리해야되는 특성을 갖고 있다. 이의 효율적인 수행을 위해 호 처리는 단계별로 세분되며, 각 단계별로 호처리 기능들이 실시간으로 수행된다. 또한 TDX-10의 호처리 기능은 계층 구조를 갖는다. 이 구조는 통화로와 신호처리, 가청 신호를 제공하는 장비들을 제어하는 하위 기능과 하위기능에서 전달된 정보를 이용해 서비스를 제어하는 상위 기능으로 이루어 진다.

• 운용 기능

운용 소프트웨어는 시스템의 기본적인 기능인 호처리 기능의 효율을 극대화할 수 있도록 관리 및 제어할 수 있는 정보 및 환경을 제공하여 주며, 운용자 또는 유지보수자의 노력을 최소화할 수 있는 각종 기능들을 제공한다. 제공되는 주요기능으로는 과금, 측정 및 통계, 망관리, 데이터 처리 기능이 있다.

• 유지보수 기능

유지보수 소프트웨어는 시스템에 높은 신뢰도와 가용도 및 지속성을 유지하기 위하여

각종 교환장비 및 프로세서에서 발생하는 모든 장애를 검출하고 장애의 심각성에 따라 적절한 조치를 취하는 프로그램이다.

1.2 ISDN 소프트웨어 관리 체계

TDX-10 ISDN 소프트웨어 형상 체계는 개발 흐름상 자연스럽게 계층적 구조로 형성되는 개발단계의 개발형상 체계, 개발된 각종 형상물들을 관리하기 위한 관리 형상 체계, 그리고, 최종 형상물들을 어떻게 조합할 것인가에 대한 생산 형상 체계로 나눌수 있다.

1.2.1 개발 형상 체계

시스템이 가져야 할 기능은 사용자로부터 요구되어 개발자가 사용자 관점으로 정리를 함으로서 결정된다. 이렇게 결정된 기능을 각각 동작으로 정리하여 필요한 동작(action)을 시스템의 기본 요소인 블럭으로 변환시킨다. 즉, 이 과정은 기능을 정립함으로서 개발되어야 할 대상(object)들을 추출하고 그것을 실현단위에 구현하는 것이다. 여기에서 블럭은 시스템 설계단계에서 설계된 기능들을 구현시키기 위한 단위로 최종산물(product)인 화일들로 구성된다.

1.2.2 관리 형상 체계

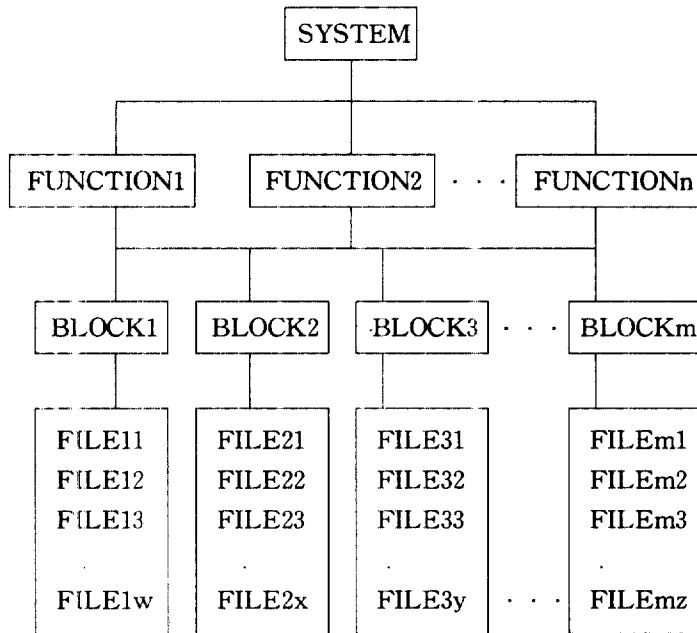
블럭은 개발시에 관리하기 편리한 적절한 단위를 고려하여 구성하고, 이러한 블럭들의 Group을 관리가 용이한 Subsystem(Functional Subsystem)으로 분류한다. 이러한 관점에서 시스템 체계를 정리할 경우 System, Subsystem, 블럭의 체계로 분류된다. 이 형상체계는 시스템 개발시 주로 이용되는 것으로서 한 Subsystem은 다른 Subsystem과 기능적으로 중복되지 않으며, 모든 블럭은 각각 한 Subsystem에만 속한다.

1.2.3 생산 형상 체계

이 형상체계는 실현시킨 생산물(product)들을 어떻게 조합하고 building해 나가느냐 하는 것을 나타내는 것이 형상체계이다. 소프트웨어 생산의 기본단위는 컴파일, 보관등의 업무를 처리하는 단위로서 화일이 된다. 소프트웨어 블럭은 한개이상의 화일들로 구성되며 이들은 실행단위인 실행모들로 분류가 되고 실행모들은 로딩되는 프로세서에 속하게 된다. 각 단계별로 공통화일이 존재 가능하다.

1.3 연구의 필요성

개발중인 시스템에 대한 품질 및 신뢰성을 검증하여, 발생한 문제점을 개발자에게 feedback할 수 있는 시발점이 되는 시험 및 평가작업은, 종종 시스템의 외적인 수행능력만을 정량적으로 측정하는데 편중되어 있는 것이 현실이다. 현재 개발중인 ATM교환기의 경우, 음성신호는 물론 데이터, 고품위영상등의 복합적인 서비스를 동시에 교환하는 광대역교환기이므로, 시스템의 신뢰성이 통신망의 운용자와 고객에게 미치는 영향은 지대하다. 개발된 시스템은 개발자의 의도대로, 주어진 기능을 수행함과 동시에 사용자의 요구를 충족할 수 있도록 주어진 시간동안 고장없이 계속적으로 동작할 수 있어야한다. 광대역 통신망에서 교환기가 높은 신뢰성을 유지하기 위해서는 다양한 변수, 즉 하드웨어 및



〈 그림 1 〉 ISDN 소프트웨어 관리 체계

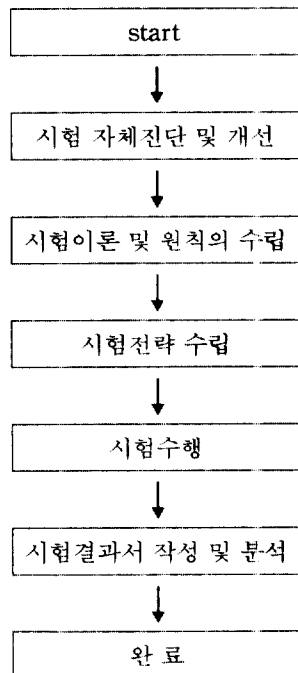
소프트웨어의 결합, 시스템과 시스템 운영자와의 interface, 시스템동작환경, 시스템작동 및 관리등 여러 가지 인자가 고려되어야한다. 교환기의 life cycle은 사용자 요구사항, 구조, 설계, coding, 시스템 시험, 운용관리, 그리고 개발자와 사용자간의 effort인데, 교환기의 신뢰성보증을 위해 다음과 같이 일반통신시스템과는 구별되는 다음의 몇 개의 특성에 대하여 이루어져야 한다.

- 교환기는 다수의 사용자에게 실시간의 서비스를 제공한다.
- 교환기는 대량의 소프트웨어를 갖고 있다.
- 교환기는 망 노드와 사용자간의 시스템의 기능과 수행을 위한 모든 환경이 일치 할 때 동작한다.
- 교환기는 평상운용모드뿐만 아니라, 프로그램의 data update시에도 중단을 용인치 않는 설계를 목표로 한다.
- 교환기는 다양한 market demands, 형상 enhancement, 규칙적인 변동, 기술변화 망 노드의 고유특성의 다양한 변화에 신속히 대응할 수 있어야한다.
- 교환기는 망 사용자의 서비스중단을 최소화하기 위해, upgrade나 patch시, 충분한 시험이 필요하다
- 교환기는 예외의 경우, 즉 abnormal한 case를 예기할 수 있어야한다.

이에 따라, 개발된 시스템이 개발자와 사용자의 의도대로, 주어진 시간동안 주어진 기능을 성공적으로 수행하는가를 측정하기 위해 시험이 진행되며, 도출된 시험결과에 대한 체

계적인 문서화와 정확한 분석기술을 통하여 문제점의 보완, 수정이 용이해질 수 있으므로 시험평가가 중요한 역할을 한다. 교환기가 높은 신뢰성을 유지하기 위해서는 다양한 변수, 즉 하드웨어 및 소프트웨어의 결합, 시스템과 시스템 운용자와의 인터페이싱, 시스템 운용환경, 시스템 작동 및 관리등 여러 가지 인자가 고려되어야 한다. 위에 언급한 여러 인자들의 측정을 위한 시험작업은 단지 시험자의 한정된 시스템 이해를 바탕으로 할 때, 최종개발목표 달성에 많은 문제점을 야기시킬 수 있다. 시험팀과 개발팀은 시, 공간적으로 어떠한 기술적인 격차도 있어서는 안되며, 항상 평행선적인 공조체제를 유지하여야 한다. 시험팀은 시험하여야 할 시스템의 정확한 기술적인 지식습득이 필요하며, 개발팀은 시험팀이 어떤 장비와 어떤 방법으로 시스템의 시험수행을 하는지를 인지하여야 하고, 어느 한팀도 다른 팀을 감사하는 입장이 아닌, 하나의 공통된 목표를 향한 체제가 필수적이다. 본 고에서는 개발초기 단계를 포함한 개발 전 순기를 통해 진행되는 시험수행 및 평가작업의 방향을 모색하고, 통합된 시스템 시험환경 조성을 위해 쓰여지는 다양한 도구들을 소개한다. 시험 및 평가작업을 위한 제반활동은 아래의 5개의 단계로 나뉘어 진다.

- 현재 시스템 시험수행과정에 대한 자체진단 및 개선
- 시험이론 및 원칙의 수립
- 시험전략 수립
- 시험수행
- 시험결과서 작성 및 분석



〈 그림 2 〉 시스템시험 및 평가 절차도

2. 현재 시스템 시험수행과정에 대한 자체진단 및 개선

현재 시스템 시험수행과정에 대한 자체진단작업은 과거, 현재를 포함한 시스템 수행과정의 분석, 도출된 문제점, 개선되어야 할 부분 등을, 시험팀 스스로 진단하고, 문제를 제기하여 차기 시험 수행시 개선된 프레임을 구성하기 위한 일련의 작업이다. 자체진단을 위해 포함되어야 할 사항은 아래와 같다.

2.1 시험용어 및 시험과정의 통일화

- 시험팀은 표준화된 시험용어를 사용
- 시험목적내의 시험되어질 형상의 정의
- 시험목적내의 시험의 범주정의
- 시험목적의 설정을 위한 표준정의
- 시험목적의 구현성을 검증하는 과정
- 시험팀의 시험과정 분석수행
- 시험팀 자체의 시험용어 및 개념을 위한 교육

2.2 시험관리

- 시험과정의 관리를 위한 책임분담
- 효과적인 시험환경구성을 위한 개발자에 대한 의견제시
- 적절한 시험도구(tool)의 선택
- 자동화된 시험도구(tool)의 사용제안
- 시험표준의 작성관리
- 시험표준의 수행여부 관리
- 통계화된 시험결과에 대한 검증
- 시스템의 오류와 시험과정의 통계적 분석 실행
- 효과적인 시험과정을 위한 계속적 평가, 제안 수행

2.3 불량항목의 파악, 기록, 분석

- 프로그램의 오류를 기록하는 작업표(worksheet) 사용
- 시험자는 오류에 대한 정확한 기록을 위한 훈련
- 오류에 의한 시스템의 영향의 기록
- 시험의 목적인 오류검출, 성능 및 기능동작의 예측, 오류의 형태 및 빈도를 포함
- 사용자요구사항의 구현단계에서 오류의 특별한 분류
- 발생된 오류에 대한 정확한 분류
- 규칙적인 시간 간격을 기초한 오류기록
- 오류의 경향(trend)을 위한 통계학적 분석
- 오류 발생률에 대한 기대치

2.4 시험수행 및 시험결과

- 표준화된 시험계획
- 기능시험에 대한 표준
- 시스템시험에 관한 표준
- 시험계획내의 시험 도구(tools), 시험방법, 시험전략 포함
- 분석의 용이성을 위한 시험결과서의 작성
- 보완되어야 할 필수항목에 대한 시험결과서 기재
- 개량(update)된 시험환경으로 인해 변경될 시험결과 예측
- 발견된 오류에 대한 권고(Recommendation)
- 통계화된 시험결과

위의 사항들을 고려하여 작성된 자체진단서는 시험팀을 관리하는 시험팀장을 포함한 시험부서의 시험자 전체가 아래와 같은 단계로 진행한다.

1. 자체진단팀 구성
2. 자체진단서의 용어해설
3. 자체진단실시
4. 자체진단 응답 결과의 분석
5. 긍정적인 응답의 합산과 차트표기
6. 최종결과의 분석
7. 자체진단결과를 통한 시험팀의 시험수행전략 개선

3. 시험이론 및 원칙의 수립

시스템시험을 구현하기 전에 시험팀이 시험전반에 관한 이론 및 원칙을 숙지하고, 가장 빈번하게 발생하는 오류의 형태와 발생원인을 파악할 수 있는 기술습득이 선행되어야 한다. 일반적으로 시스템을 시험하는 기본적인 목표는 아래와 같다.

- 개발자가 기대하는 기능과 실제 시스템의 수행기능과의 차이를 파악; 사용자 요구 사항 구현여부 및 손상된 기능 파악
- 시스템 오류시 기대되는 위험효과의 예측
- 시험중이나 시험 후에 발생된 시스템 오류의 분류 및 요약

프로그램의 오류는 시스템기능의 손상을 가져오며, 그 오류는 몇 개의 faults를 유발시킬 수 있다. 일반적인 시험팀의 목표는 “bugs”를 찾는 것으로 인식되기 쉬우나, 실제 시험자의 역할은 “bugs”를 찾을 뿐만 아니라 개발된 프로그램의 동작을 예측할 수 있어야 한다. 교환기의 경우 기존의 불량이라는 단일 측정계수로는 시스템의 오류를 정확히 표현하기에는 불충분하며, 근원적인 문제점의 파악을 위해 시스템의 오류는 그 미치는 영향에

따라 크게 2개의 범주로 나뉘어진 후 세분되어 진다.

- 주 오류 : 발생된 오류가 교환기의 동작, 운용, 관리, 과금등의 주 기능을 마비시켜 사용자에게 서비스를 공급할 수 없는 원인을 제공하는 오류
- 부 오류 : 사용자에게 서비스를 제공하는데 큰 영향을 미치지 않는 오류

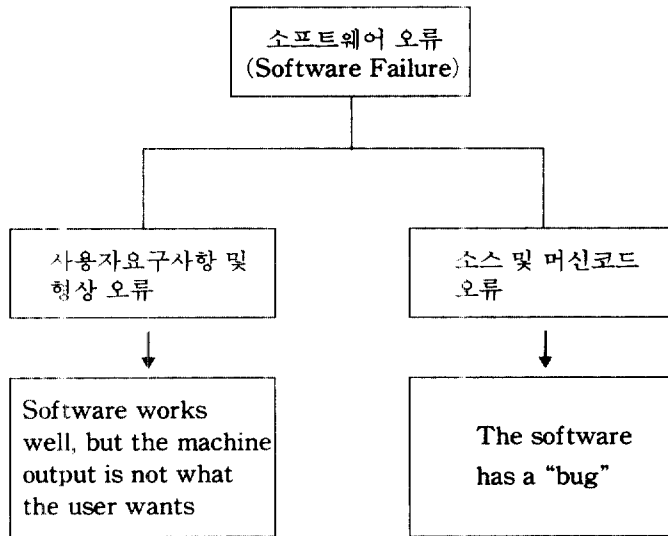
오류의 형태에 따라 기술적인 오류, 구조상의 오류, 개인오류, 기타오류로 분류할 수 있다.

- 기술적인 오류
 - SW errors
 - Procedural errors
 - Mishandling tool errors
- 구조상의 오류
 - H/W errors
 - Test tool errors
- 개인 오류
 - Operational errors
 - Testing method errors
 - Unskilled tester errors
- 기타 오류
 - Environmental errors

분류된 오류 데이터는 근원지와 무엇이 오류를 발생시켰으며, 어떤 형태로 발생되었는지에 대한 원인분석이 필요하다. 그러한 작업은 사용자의 관점에서 본 문제의 묘사와 개발자의 관점에서 본 문제의 묘사, 그리고 보완, 수정되어져야 할 내용으로 구성된다. 조사에 의하면 실제로 발생하는 시스템 오류중 46-60%가 개발초기단계의 사용자요구사항의 구현단계에서 발생한다. <그림 2>는 대부분의 소프트웨어가 외형적으로 완전하게 동작하는 것처럼 보이나, 실제의 시스템 output은 본래의 사용자 요구사항에 정의된 내용을 충분히 구현하지 못함을 보여준다. 이는 상위레벨에서의 개발형상의 정의과정에서, 사용자가 원하는 실제기능을 구현함에 있어 정확하지 않거나, 불충분함을 의미한다. 혹은 사용자의 요구사항이 명백치 않아 이를 구현함에 무리가 있을 경우로 분류된다.

사용자 요구사항의 시스템 구현 단계에서 종종 발생하는 오류의 근원은 아래와 같다.

- 기능의 동작형식, 동작정보의 미비
- 성능에 관한 정보 미비
- 환경인자의 미고려
- 각 사용자 요구사항간의 비호환성
- 시험불가능한 요구사항
- 부정확한 요구사항
- 실현불가능한 기능정확도 요구



〈 그림 3 〉 사용자요구사항의 구현오류

4. 시험전략 수립

시험자와 개발자가 예상하는 고품질의 시험을 수행한다는 것은, 시험수행전 계획된 시험표준을 만족하는 것을 의미한다. 시험수행에 관한 일관성 있는 시험표준이 없이는 품질을 결정할 수 있는 기본요소가 제공되지 않으므로 체계화되고, 정제된 시험수행 전략이 수립되어야 한다. 시험수행에서의 품질관리는 시험작업이 시험표준과 일치하는지를 확인하는 시험방법, 시험과정, 시험도구들을 의미하며, 시험수행에서 얻어진 시험과정의 효과성을 평가하는 품질보증활동의 하나로서 올바른 시험전략수립은 필요불가결하다. 그러므로, 시스템의 품질수준을 측정하는 시험수행을 위한 표준화된 시험전략수립은 시스템의 품질보증활동의 성패를 좌우하는 중요한 역할로서 인식되어야 한다. 일반적으로 아래와 같은 8단계의 고려사항으로 시험전략은 구성된다.

4.1 시험목적 설정

시험목적의 설정은 시험전략의 가장 첫번째 단계로서 무슨 시험을 할 것인가를 파악하는 과정으로서 차후 진행되는 일련의 과정의 근간이 되는 시험을 위한 요구사항이다. 시험목적의 작성은 시스템 단위, 기능 단위로 목록을 작성한다.

4.2 시험팀 구성

시험팀은 기능을 구현한 개발자를 포함하는 것이 이상적이나, 시험을 전담하는 부서 자체가 시험을 담당할 때는, 시험절차서의 기능을 숙지할 뿐만 아니라 기능동작을 예측할 수 있는 시험자를 선정한다.

4.3 시험계획 작성

시험계획은 설정된 시험목적은 어떻게 구현하는가를 구상하는 단계로서 전반적인 시험 절차를 포함한다. 시험계획은 시험수행에 대한 전반적인 윤곽을 나타내고, 시험절차는 각 시험 별로 세부절차를 나타낸다. 시험계획은 시험의 전략, 시간계획, 시험시스템의 구성, 시험도구(tool)의 선정, 인원, 절차 등을 포함하고, 시험절차는 각 시험의 목표, 완료기준(completion criteria), 시험사례, 평가를 포함한다. TDX-10 ISDN 교환기는 모든 작업을 실시간으로 처리해야 하므로 높은 신뢰도가 요구되는 시스템으로 시험수행시 여러 가지 사항에 대해 고려하여야 한다. 예를들면 시스템의 기본이 되는 호 처리 기능과 이에 수반되는 운용, 보전기능은 교환기 운용조건(운용프로파일)에 유사하게 적절히 혼합하여 시험을 수행해 나간다.

4.4 시험환경 구성

시험환경 구성을 위한 제반활동은 시험팀과 기능개발 담당자와의 의견의 합치 하에서 이루어진다.

4.5 시험 도구(tool) 제작

시스템 시험시 사용될 도구(tool)의 제작단계로서, host computer상에서 시스템 하드웨어의 도움없이 시스템 프로그램을 작동시키고 기능시험을 가능케 하는 자동화된 도구(tool)의 선정, 제작이 진행된다. 이러한 자동화된 시스템으로 여러 개발자가 동시에 독립적인 접근을 할 수 있으며, 시스템 소프트웨어가 target에 설치되기 전에 발생하는 많은 논리적인 문제점을 해결해 주고, 시험시간과 debugging시간을 줄일 수 있다. 또한 시험과정의 표준화를 꾀할 수 있으며, Regression testing의 용이성을 주며, 시험결과의 신뢰성을 증가시킨다. 또한 개발된 기능에 대한 정상여부를 판단하는데 기준이 되는 지침서로서 시험절차서의 작성이 필요하다. 시험절차서는 시험 수행 방법 및 판정 기준, 기타 시험 수행시 참고될 사항을 기술하고 있다. 특히 시험절차서는 그 기능에 대한 정확한 시험방법의 모든경우(test case)를 기술하는데 어려움이 있기 때문에, 가장 기본이 되는 시험 방법 중심으로 기술되어 있다.

4.6 시험수행

시험시간이 긴 기능을 시험할 때, 새로운 단계의 시험실시전 이미 진행되고 있는 시험의 종료를 검증할 수 있어야 한다. 또한 시험시간이 길고, 각기 다른 시험절차가 요구될 때 경로표(route sheet)의 사용으로 시험과정을 체계화할 수 있다. 시험장비의 운용시, 장비의 정확성을 검증하는 작업이 필요하다. 이는 시험의 품질을 객관화시키는 수단으로서 시험수행전 선행되어야 한다.

4.7 시험결과서 작성

- 표준 작업표(worksheet)의 사용
- 시험판정을 위한 범주 설정(합격, 불합격, 미판정, 유보등)

문서번호 : 2314

항목명 : 시스템 가상오류 내역 출력

대항목명 : 시스템 정보처리 및 제어

목적

시스템내의 가상오류 내역 출력 기능 시험

시험 조건

제약 조건

원격 교환장치가 OFF되어 있어야 한다.

타 기능과의 연계 조건

시험 방법 및 결과

1. 대상 데이터에 대하여 가상오류를 유발시킨다.

C2982 INOT-MSG-DTR

다음과 같은 메시지가 출력되어야 한다. (M2982)

2. 오류를 유발시킨 Relation에 해당되는 감사 상태 출력 명령을 입력한다.

C4110 LIST-INOT-MSG

1항에서 유발시킨 시스템 메시지가 출력되어야 한다. (M4110)

3. 1항의 오류가 검출되는지 확인한다. (F6701)

〈 그림 4 〉 시험절차서 예

4.8 시험결과의 오류분류 및 분석

- 오류의 형태
- 오류의 빈도수
- 오류의 발생원인
- 오류의 시스템에 대한 영향
- 오류의 연계성
- 통계학적 자료분석을 정확한 오류의 분류

5. 시험수행

시험은 시험목적과 개발순기에 따라 다음의 네 가지 방법으로 수행한다. 각 기능의 완전한 동작을 증명해 보이는 기능적 시험(Functional test)과, 각 프로그램 및 하드웨어 결과물의 내부명세에 따라 내부동작을 완전하게 증명해 보이는 구조적 시험(Structural test), 시스템의 각 기능들이 정의되어진 대로 작동하는가를 검증하는 검증시험(Validation test), 그리고 소프트웨어의 개발순기를 통해 시스템의 기능이 의도된 대로 정확히 수행되는지와 그 자체기능 혹은 다른 기능과의 연동으로 인한 시스템의 성능저하를 검증하는 체계적인 과정인 확인시험(Verification test)으로 나뉘어진다. 또한 시험은 동적시험(Dynamic testing)과 정적시험(Static testing)으로 나뉘어 지는데, 동적시험은 실제로 시험사례(test case)를 조성후 시스템을 동작후 나타난 결과가 기대한 결과를 만족하는지를 check하는 시험이며, 정적시험은 프로그램의 특성을 측정하는 기술로서 프로그램의 syntax, procedure간의 parameter일치, typing, 사용자요구사항 및 specification의 정확한 구현 등을 검증한다. 일반적으로 시험수행은 크게 블럭 시험(Block test), 통합 시험(Integration test), 기능 시험(Function test), 시스템 시험(System test)으로 나뉘어진다.

• 블럭 시험(Block test)

블럭시험(block test)은 시스템의 블럭내의 해당 유니트에 잔존하는 에러를 모두 제거시키는 데 목적이 있다. 구현단계에서 개발된 소프트웨어 및 하드웨어 제품들이 사용자 기능규격과 요구사항에 정의된 수준을 충족하는지를 명시된 내용의 구현여부를 검증하고, 각각의 모듈(module)이 하나의 단위로서 독립적으로 수행되는지를 시험한다. 이 시험에 사용되는 기법은 주로 모듈(module)의 제어구조상에 존재하는 특정한 경로들을 실행시키는 structural test(white-box)기법을 많이 이용한다. 시험사례(test case)는 선택(selection), 반복(loops)등의 부분들을 수행함으로써 시스템의 논리적 경로를 증명하는 내용이다.

• 통합 시험(Integration test)

통합시험은 블럭시험(block test)이 끝난 블럭들을 설계된 구조로 통합하면서 각 블럭간에 관계되는 인터페이스를 시험하는 단계이다. 이 시험에는 검증과 프로그램 구성이라는 상대적인 문제에 중점을 둔 functional test(black-box)기법을 주로 사용한다. 이 단계에서 생성되는 시험사례(test case)들을 실제 운용환경과 흡사한 시험환경을 조성하여 랜덤하게 취해질 수도 있고, 합리적인 시험전략에 의해 미리 정해진 결정적인 형태, 즉 기능의 적절한 동작과 적합한 입력에 대한 출력의 정확성을 증명하는 내용이어야 한다. 그리고 이 단계에서는 검출된 결함이 수정될 때 새로운 결함이 생성될 가능성이 매우 높다.

• 기능 시험(Function test)

기능시험은 통합시험에서 결함들이 검출되어 수정된 후 패키지가 된 이후 적용되는데 때로 검증시험(validation test)으로도 불리기도 한다. 이는 특정한 기능을 구성하는 기

능간에 상호작용의 시험으로 개발자는 개발한 기능과 정합 되는 다른 기능간에 상호작용을 시험한다. 또한, 소프트웨어의 기능들이 사용자의 적절한 기대치에 부합되는가를 확인하는 과정이다. 적절한 기대치들은 사용자요구사항에 의해 정의되며, 업무수행요령은 아래와 같다.

- 기능단위의 수행과정 및 블럭들간의 상호작용에 중점
- 사용자 입장이 아닌 시스템 개발자의 입장에서의 기능시험
- 시험환경 및 필요한 도구들을 개발, 제공하는 담당부서 배정[2]

• 시스템 시험(System test)

시스템 시험이란 소프트웨어와 하드웨어 그리고 관련정보(information)등을 모두 합친 실제의 운용조건하에서 시스템의 모든 기능들을 연동 시켜서 시험하고, 한계조건하에서 시스템이 정상적으로 동작되는가를 검증하는 것이다. 또한 시스템이 시스템규격의 모든 내용을 만족하는지를 확인하는 시험이다. 시스템 시험 단계에 수행되는 시험은 아래와 같다[한기철, 1990].

- 시스템의 성능(performance) 시험
- 시스템의 용량(capacity)시험
- Real time 시스템을 위한 concurrence 시험
- Stress/Load 시험
- 동작(Operational) 시험
- Interface 시험
- Installation 시험
- 신뢰성(reliability) 시험
- 환경시험
- Regression 시험

6. 시험결과서 작성 및 분석

시스템의 운용 및 시험시 발생된 제반 문제점, 기능보완사항등을 문제해결팀, 또는 개발자에게 의뢰하여 시스템개발과정에 효율적으로 반영하기 위해 시험자에 의해 작성된 시험결과서는 시스템의 성능(performance)과 능력(ability)의 품질을 나타내는 증거가 되는 문서이며, 개발팀은 그 보고서의 내용으로 개발된 기능의 response와 차후 보완되어야 할 기능을 예측할 수 있으므로 상세하고 정확한 문서작성이 요구된다. 교환기 같은 long term test가 요구되는 경우에는 각 시험단계를 분리하여 독립된 시험결과서의 작성이 필요하다. 또한 시험결과에 대한 공정성, 일관성, 객관성 등을 충분히 수용할 수 있는 기준(criteria)의 설정과 시험팀장의 시험결과에 대한 재검토가 필요하다. 또한 시험결과를 바탕으로 한, 시스템의 차후 상태를 예측하여야 하며, 분류된 오류에 대한 경향분석 및 시

명적인(critical) 시스템 문제점의 지적을 포함하며, 종합적인 판정을 위한 결론이 제시되어야 한다. 시험결과서에 포함되어야 할 내용은 아래와 같다[5].

6.1 시험결과서 작성

- 시험을 수행한 시험자
- 시험이 수행된 시스템 이름
- 시험이 수행된 장소
- 시험이 수행된 일자
- 시험방법
- 시험상태(합격, 불합격, 미판정, 유보, 오류등)
- 불량내역 및 문제점 해결방안
- 기타 참고사항

작성된 시험결과서를 바탕으로 시험결과에 대한 분석이 실시되며, 최종적인 시험결과 분석에 대한 보고서가 작성된다. 시험결과 분석에 대한 보고서에 포함될 내용과 일반적인 시험결과서의 작성 및 분석과정은 아래의 단계로 진행된다.

- 시험결과서 작성
- 시험결과서 review
- 시험결과서 분석
 - 치명적(Critical) 오류 검출
 - 통계학적 오류분류
 - 오류경향 분석
 - 시험사례(test case) 분석

6.2 시험결과서 보존

발견된 오류는 다양한 차트와 그래프로 표현함으로써 오류의 발생빈도, 발생경향, 발생근원지등을 용이하게 파악할 수 있으며, 시험과정을 추적하는데 편리함을 준다. 자료의 분석 및 표현을 위해 자주 쓰이는 기본도구는 아래와 같다.

- 차트/그래프: 오류경향 예측, 오류 데이터의 응축(Bar, Line, Circle)
- Check sheet/Tally sheet: 분류된 자료의 category화(오류종류, 오류근원지, 오류 발생 주기분석)
- Histogram/Frequency Distribution: 분석되어질 기능의 품질특성과 관측수와의 상관관계 표현
- Pareto's law: 소수의 오류가 다량의 faults를 유발한다는 이론의 도입
- Cause and Effect: 발생가능한 모든 오류에 대한 분류작업 구현으로 보완, 수정 작업을 위한 방향제시

7. 결론

시험의 근본적인 목적은 시스템 개발의 각 단계 및 전체단계에서 만들어진 제품의 오류를 찾아내는데 있다. 시험의 결과에서 오류가 없다는 것은 찾아낼 오류가 더이상 없거나 시험이 충분치 않아서 시스템내에 오류가 잠재해 있다는 것을 의미한다. 부적절한 시험으로 인해 잠재해 있는 오류들은 주로 현장에서 운용 중에 발견되며, 이를 정정하기 위한 노력과 비용은 개발단계에서 수정하는 것보다 더 많은 노력과 비용을 요구한다. 그러므로 교환기의 품질과 신뢰성을 보장하기 위한 대형시스템의 시험은 체계화된 계획과 그 계획의 실현을 위한 세분화된 지침이 필요하며, 무엇보다도 먼저 시험자가 숙지하여야 할 시험에 대한 test-mind가 정립되어야 한다. 시험의 품질을 보장하는 것은 전체 시스템의 요구되어진 품질을 보장하는 것과 같으므로, 기술적인 시험에 대한 접근(approach)뿐만 아니라 시험을 어떻게 수행할 것인가 하는 전략적인 측면에서의 시험작업의 중요성이 인식되어져야 한다.

참고문헌

- [1] 한기철(1990), "5ESS 개발 품질보증 제도(II)," 「한국통신 품질보증지」, 제3권 1호, pp. 37-40.
- [2] Keki R. Bhote(1988), "World Class Quality," *AMA Management Briefing*, New York.
- [3] The Software Quality Company(1988), *Software Quality Engineering*, New York.
- [4] Samuel T. Chanson, Antonio A. F. Loureiro, and Son T. Vuong(1993), "On the Design for Testability of Communication Software," *Proc. of International Test Conference*, pp. 190-199.