

데이터베이스 소프트웨어의 시험 사례 분석

양해술*, 강배근**, 이하용***

Testing case analysis of Database Software

Hae-Sool Yang*, Bae-Keun Kang**, Ha-Yong Lee***

요약

데이터베이스의 의미는 '논리적으로 연관된 레코드나 파일의 모임'으로 방대한 자료를 효율적으로 관리하기 위해 많은 양의 자료를 처리하는 곳에서 널리 사용되고 있다. 이러한 데이터베이스를 생성하고, 관리하고, 사용자의 요구(Request)에 따라 응답(Response)을 보내는 프로그램들을 데이터베이스 관리시스템이라고 부른다. 본 연구에서는, 이와 같이 중요한 데이터베이스 소프트웨어 제품의 품질 수준을 파악할 수 있는 지표를 도출하여 지표산식을 정의하였다. 또한, 지표의 결과를 산출하기 위해 필요한 수집항목을 선정하며 수집과 분석을 통해 실질적으로 어떤 결함 유형들이 주로 발생하고 있는지를 확인하였으며 데이터베이스 소프트웨어에 대한 시험 평가 모델을 개발하여 시험 사례를 분석하였다.

Abstract

The meaning of Database in order to manage the data which is huge in the meeting of the record which logically had become the fire tube or file 'efficiently' is widely used from the place which controls a many double meaning data. Like this data base it creates, it manages, the programs which send an answer back according to demand of the user as DBMS it calls. Like this it will be able to grasp the quality level of the data base software product which is important index from the research which index it buys it defined. Also, in order to produce the result of index it selects the collection item which is necessary and collection and analysis it leads and what kind of defect types occur substantially mainly, and it confirmed and the test and evaluation model in about data base software and a tentative instance it developed it analyzed.

▶ Keyword : 데이터베이스(Database), 소프트웨어 시험(Software testing), 사례 분석(Case analysis)

• 제1저자 : 양해술 교신저자 : 강배근

• 투고일 : 2009. 03. 25, 심사일 : 2009. 04. 09, 게재확정일 : 2009. 05. 14.

* 호서대벤처전문대학원 교수 ** 혁신경영기술융합대학원 석사 *** 서울벤처정보대학원대학교 교수

※ 본 연구는 지식경제부와 정보통신연구진흥원의 대학IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2009-(C1090-0902-0032))

I. 서론

정보를 누구나 쉽게 얻을 수 있고 공유할 수 있을 때 가치가 있는 것이기에 가능한 많은 사람이 이용하기 위해서 통신 위성을 올리는 '텔레텔식'의 꿈의 통신을 이용하고 있다. 즉, 분산된 어떠한 정보를 어떠한 매체에 단일화시켜 저장하여 빠르고, 아주 쉽게, 아무 곳에서든 많은 사람이 이용할 수 있게 한다. 종이 대신 그 자리를 채울 수 있는 것은 단지 컴퓨터의 저장 매체이며 그에 따르는 소프트웨어이다[4]. 데이터베이스의 사전적 의미는 논리적으로 연관된 레코드나 파일의 모임이다. 다시 말해 데이터베이스는 방대한 자료를 효율적으로 관리하기 위해 생겨났고, 현재는 많은 양의 자료를 처리하는 곳에서 널리 활용되고 있다[2,3]. 이러한 데이터베이스를 생성하고, 관리하고, 사용자의 요구(Request)에 따라 응답(Response)을 보내는 프로그램들을 데이터베이스 관리시스템이라고 부른다.

본 연구의 2장에서는 데이터베이스 소프트웨어를 시험하기 위한 일반적인 특성 및 요구사항과 국제품질표준에 대해 기술하였다. 그리고 국제표준을 참조하여 품질 수준을 측정할 수 있는 메트릭을 제안하였다. 3장에서는 데이터베이스 소프트웨어의 시험환경을 구축하고 결합내역을 분석 하였다.

II. 관련 연구

1. 데이터베이스 시스템의 발전

데이터베이스의 어원은 시스템디벨로프사에서 개최한 심포지엄에서 처음 사용하였으며 데이터베이스의 독립성 개념을 도입한 데이터베이스의 모체는 제너럴 일렉트릭사에서 개발한 "Integrated Data Store"에서 찾아볼 수 있다. 또한 데이터베이스 표준화 작업은 CODASYL의 DBTG에서 시도되었다.

그 후 데이터베이스에 대한 연구가 꾸준히 진행되었는데 대표적인 데이터베이스 시스템으로는 관계데이터베이스시스템, 객체지향 데이터베이스 시스템, 망 구조 데이터베이스 시스템, 계층 구조 데이터베이스 시스템이 있다. 최근에는 관계 데이터베이스 시스템을 이루며 객체지향 데이터베이스 시스템의 활용이 점차 확대되어 가고 있는 상태이다[5,6].

관계 데이터베이스 시스템은 E.F.Codd에 의하여 관계 데이터베이스 모델이 제안되었으며[1], 이를 기반으로 실험용

관계 데이터베이스 관리 시스템인 SYSTEM R이 IBM에서 개발되었다[2]. 또한 데이터베이스 언어인 SQL이 ANSI/X3 SPARC에 의해 관계 데이터베이스 관리 시스템의 표준언어로 채택되었다. 현재에도 데이터베이스에 관한 연구가 활발히 진행 중인데 그 대표적인 예로 미국의 전자계산기학회의 데이터베이스 분과위원회와 데이터베이스 관리연구회 등을 중심으로 연구가 진행되고 있다.

2. 데이터베이스 시스템의 특성

데이터베이스 시스템은 다음과 같은 몇 가지 특성을 가지고 있다.

2.1 실시간 접근성

오늘날과 같이 급변하는 상황에서 일괄처리(batch processing)에만 의존하는 정보는 한정되어 있기 때문에 컴퓨터가 접근할 수 있는 저장 장치에 수록된 데이터베이스는 수시로 비정형적인 질의에 실시간으로 처리하고 응답할 수 있어야 한다. 여기서 실시간 처리(real-time processing)라고 하는 것은 생성된 데이터를 즉시 컴퓨터에 바로 반영할 수 있게 하는 처리 방식을 말한다. 일반적으로 온라인 처리라 하면 보통 이 실시간 처리를 의미한다.

2.2 지속적인 변화

데이터베이스의 내용으로 표현되는 데이터베이스의 상태는 정적이 아니고 동적이다. 즉, 데이터베이스는 새로운 데이터의 삽입, 기존 데이터의 삭제, 갱신을 통하여 실세계를 정확히 반영하는 데이터를 유지하면서 변화한다. 데이터베이스는 항상 변화하는 실세계를 반영해야 하기 때문에 자연히 지속적으로 변하게 된다. 이 지속적인 변화 특성 때문에 데이터베이스를 정확하게 관리하는 것이 더욱 어려운 문제가 된다.

2.3 동시 공유

데이터베이스는 서로 다른 목적을 가진 응용들을 위한 것이기 때문에 여러 사용자가 동시에 하나의 응용 프로그램에 의해 접근되는 데이터와 다르며 비록 여러 프로그램이 같은 데이터를 공유한다 하더라도 완전히 순차적으로 공유하는 개념과도 다르다. 같은 내용의 데이터를 여러 사람이 서로 다른 방법으로 동시에 공유한다는 것은 그 관리 면에서 복잡하게 될 뿐만 아니라 그 조직 면에서도 매우 복잡하게 된다. 왜냐하면 모든 데이터베이스의 데이터 요소들이 처음부터 항상 동시 공유가 가능하도록 조직 관리되어야 하기 때문이다.

2.4 내용에 의한 참조

데이터베이스 환경 하에서 데이터의 참조는 수록되어 있는

데이터 레코드들의 주소나 위치에 의해서가 아니라 데이터의 내용, 즉 데이터가 가지고 있는 값에 따라 참조된다. 일반적으로 참조하길 원하는 데이터의 자격 요건을 제시하면 이 조건을 만족하는 모든 레코드들은 하나의 논리적 단위로 취급되고 접근된다.

3. 국제 품질 표준 ISO/IEC 9126의 체계

소프트웨어 제품평가에 관한 국제표준인 ISO/IEC 9126에서 정의하고 있는 각 품질특성의 체계는 [표 1]과 같다 [8,9,10,11].

표 1. ISO/IEC 9126의 품질특성의 체계

Table 1. System of quality characteristic of ISO/IEC 9126

기능성	적합성	사용자의 목적하는 바에 따라 적절한 기능을 제공하는 능력
	정확성	올바른 혹은 동의된 효능 결과를 제공할 수 있는 능력
	상호 운용성	하나 이상의 지정된 타 시스템과 서로 작동하는 능력
	보안성	인가되지 않은 사람이나 시스템의 액세스를 방지하여 정보 및 데이터를 보호하는 능력
	준수성	그 응용에 관한 표준, 규정, 관례 등을 따르는 능력
	신뢰성	프로그램이나 데이터의 결함으로 인한 기능장애를 피할 수 있는 능력
사용성	이해성	사용자가 소프트웨어를 활용하기 위한 방법이나 조건, 적정성 등을 파악할 수 있게 하는 능력
	습득성	사용자가 소프트웨어의 응용을 배울수 있게 하는 능력
	운용성	사용자가 소프트웨어를 운영하고, 제어할 수 있도록 하는 능력
효율성	친밀성	소프트웨어가 사용자에게 호감을 갖게 하는 능력
	준수성	사용성에 관한 표준, 규정, 관례, 스타일 등을 따르는 능력
	시간 반응성	주어진 조건에서 기능을 수행할 때 적절한 응답시간, 처리시간, 처리율을 제공하는 능력
유지보수성	자원 효율성	주어진 조건에서 기능을 수행할 때 적절한 양과 종류의 자원을 사용하는 능력
	준수성	효율성에 관한 표준, 관례 등을 따르는 능력
	유지보수성	소프트웨어의 약점이나 장애 원인을 진단하고 변경될 부분을 식별할 수 있게 하는 능력
이식성	변경성	지정된 변경사항이 구현될 수 있게 하는 능력
	안정성	변경에 따른 예상 밖의 결과를 최소화 하는 능력
	시험성	변경되었을 경우 검증 받을 수 있는 능력
다른 조직 또는 하드웨어, 소프트웨어 환경으로 옮겨질 수 있는 능력	준수성	유지보수성에 관한 표준, 관례 등을 따르는 능력
	적응성	상이한 환경에 적응하는데 필요한 최소한의 조치만으로 이식될 수 있는 능력
	설치성	지정된 환경에 설치될 수 있는 능력
공존성	공존성	다른 독립적인 소프트웨어와 공존자원을 사용해야 하는 공동 환경에서 공존할 수 있는 능력
	대체성	동일 환경에서 같은 목적으로 사용되는 다른 소프트웨어를 대신하여 사용될 수 있는 능력
	준수성	이식성에 관한 표준, 관례 등을 따르는 능력

4. 데이터베이스 품질수준 메트릭

본 연구에서는 [그림 1]과 품질 수준 메트릭을 품질 특성별로 제안하여 데이터베이스 소프트웨어를 시험하기 위한 방법을 모색하였으며 데이터베이스 소프트웨어의 품질 요구사항을 분석하여 ISO/IEC 9126과 12119의 품질특성과 대응시키고 이로부터 데이터베이스 소프트웨어를 평가하기 위한 항목을 도출하며, 평가 항목에 대한 평가 방법을 구축하고 시험 및 평가 기준을 설정하여 평가 메트릭인 [표 2]를 구축하였다[18].

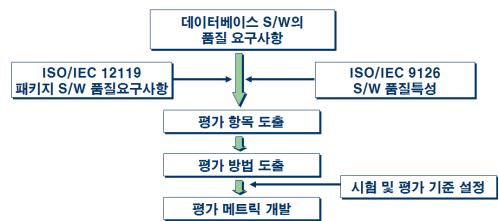


그림 1. 데이터베이스의 메트릭 도출
Fig. 1. Metric draw of database

표 2. 데이터베이스 품질수준 메트릭

Table 2. Database quality level Metrik

기능성	기능 구현 정확성 정보 제공	계산식	기능 구현 정확성 정보 제공 = B(문서에 정확히 기술되어 있는 기능 수)/A(프로그램에서 제공하는 전체 기능 수)
		결과 영역	$0 \leq \text{기능구현 정확성 정보 제공} \leq 1$
신뢰성	기능 구현 정확성	계산식	- Success TC : i 번째 기능 확인을 위해 수행한 테스트케이스 중 성공한 건 수 - Total TC : i 번째 기능 확인을 위해 수행한 테스트케이스 수 $- B = \frac{\sum_{i=1}^n \text{Success}_{TC_i}}{\text{Total}_{TC_i}}$
		결과 영역	$0 \leq \text{기능구현 정확성} \leq 1$
신뢰성	다운 회피율	계산식	다운회피율 = $1 - B(\text{다운 회수})/A(\text{발견된 결함수})$
		결과 영역	$0 \leq \text{다운회피율} \leq 1$
신뢰성	고장 회피율	계산식	고장회피율 = $1 - B(\text{고장 회수(심각한 결함이 발생한 수)})/A(\text{발견된 결함수})$
		결과 영역	$0 \leq \text{고장회피율} \leq 1$
신뢰성	오작동 회피율	계산식	오작동회피율 = $1 - B(\text{결함 발생 수})/A(\text{사용자가 오작동을 수행한 수})$
		결과 영역	$0 \leq \text{오작동회피율} \leq 1$
효율성	평균 처리 시간	계산식	- 평균 처리 시간 = $1 - \min(1, B(\text{처리 평균 시간})/A(\text{처리 평균 시간의 한계값}))$ $- B = \frac{\sum_{i=1}^n T_i}{N}$ - T_i = i 번째 테스트의 처리시간 - N = 처리 시간 테스트 케이스 수
		결과 영역	$0 \leq \text{평균 처리 시간} \leq 1$
사용성	기능 학습 용이성	계산식	기능 학습 용이성 = $A(\text{학습을 쉽게 할 수 있는 기능의 수})/B(\text{전체 기능의 수})$
		결과 영역	$0 \leq \text{기능 학습 용이성} \leq 1$

도움말 접근 용이성	계산식	도움말 접근 용이성 = A(해당 점검표에서 Y로 측정된 항목의 수)/B(해당 점검표에서 평가 대상이 되는 항목의 수)
	결과 영역	$0 \leq \text{도움말 접근 용이성} \leq 1$
유지 보수 수성 정보 제공	계산식	환경설정 변경 안정성 정보제공 = A(환경 변경으로 인해 발생할 가능성이 있는 오류에 대한 정보 제공 여부)
	결과 영역	환경설정 변경 안정성 정보제공 = Y or N or NA
이식성	설치 정보 제공	계산식 설치 정보제공 = (Y로 측정되는 항목 수) / (데이터 항목 수)
	결과 영역	$0 \leq \text{설치 정보제공} \leq 1$
	설치 가능성을	계산식 설치 가능성 = B(성공한 설치(설치 재시도) 횟수)/A(시도한 설치(설치 재시도) 횟수)
결과 영역	$0 \leq \text{설치 가능성} \leq 1$	

III. 데이터베이스의 시험절차

1. 데이터베이스 소프트웨어의 표준 환경 구축

본 연구 대상인 데이터베이스 소프트웨어는 현재 널리 활용되고 있는 DBMS 상의 변경 데이터를 테이블 및 컬럼 단위로 추출하는 CDC(Change Data Capture) 프로그램이나 삽입, 변경, 삭제 및 검색 기능을 제공하는 관계형, 데이터 관리 기능, 무장애 운영을 보장하는 고가용성 기능, 사용자 편의성을 제공하기 위한 관리도구 등을 제공하는 데이터베이스 관리시스템(DBMS) 등의 기능을 수행하는 소프트웨어로서 데이터베이스 소프트웨어의 표준 시험 환경은 [그림 2]와 같이 구축하여 시험 적용 하였다[12,13,14].

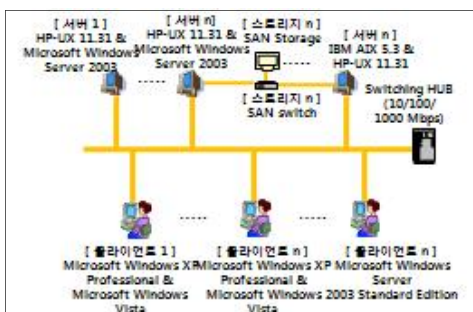


그림 2. 데이터베이스의 표준 시험 환경 구축
Fig. 2. Standard testing environment construction of database

데이터베이스 소프트웨어의 표준 시험 환경 구축을 위해 서버에 사용되었던 프로그램으로는 Microsoft Windows Server 2003과 HP-UNIX, IBM AIX가 사용되었으며 클라이언트에 사용된 프로그램으로는 Microsoft Windows XP Professional, Microsoft Windows Vista, Microsoft

Windows 2003 Standard Edition이 사용되었고 네트워크에는 10/100Mbps 스위칭 허브를 사용하였다.

2. 시험대상 데이터베이스 소프트웨어

본 연구에서 시험한 데이터베이스 소프트웨어를 주요기능별로 구분하여 정리하면 [표 3]과 같다[15].

표 3. 시험대상 소프트웨어 구분
Table 3. Testing target software division

구분	DB-1	DB-2	DB-3	DB-4	DB-5
내용	데이터베이스 관리시스템	관계형 데이터베이스 관리시스템	데이터베이스 모니터링 솔루션	CDC(Change Data Capture) 프로그램	DBMS 성능관리 솔루션
주요 기능	<데이터베이스 엔진> 데이터베이스 테이블 관리 기능 데이터 삽입/변경/삭제/검색 기능 사용자/권한 관리 기능 백업/복구 기능 <이중화> Fail-over 기능 로드 밸런싱 기능 <관리 도구> 대용량 SQL 실행기 배치 등록률 GUI를 통한 관리 <시스템 연동> JDBC ODBC Embedded SQL	<데이터베이스 엔진> 데이터베이스 테이블 관리 기능 삽입/변경/삭제/검색 기능 사용자/권한 관리 기능 <이중화> <이중화> (replication) Fail-over 기능 <관리 도구> iSQL(대용량 SQL 실행기) iLoader(배치 등록률) Admin Center(GUI를 통한 관리) 등	<DBMS 인스턴스> 조회, 추가 및 삭제 기능 <DBMS 모니터링 이벤트 및 통계 항목 설정 기능> <DBMS 구성, 용량 및 성능 집계 정보 분석 기능 등>	CDC(Change Data Capture) 프로그램 -변경 데이터 추출(테이블 및 컬럼 단위) 기능 -모니터링 기능 -장애 대처 기능 -결과 기록 및 조회 기능 -환경 설정 기능	DBMS 성능관리 솔루션 -DBMS 및 OS 실시간 모니터링 기능 주요지표(Using CPU, Redolog Buffer Wait, Resource Manager Wait 등), RAC, Session, SQL처리현황, Service, CPU, Memory 등 -DBMS 성능 분석 기능 -세션정보, SQL수행, 응답시간 기반 이벤트 분석 등 -SQL 튜닝 기능 -다양한 형태의 보고서 작성 기능 -환경설정 기능 등
서버	-SQL Test Suite v6_pc_isql -JDK 1.6 -TPC-H (tpch 2.8.0) -TPC-C (Benchmark SQL-2.3.2)	-JDK 1.4.2	<DBMS> MS-SQL Server Standard(모니터링 데이터 저장용 DBMS) Oracle 10g(모니터링 대상 DBMS) <DBMS> MS-SQL Server Standard(모니터링 데이터 저장용 DBMS) Oracle 9i(모니터링 대상 DBMS)	-시험 대상 제품(서버 모듈) -시험 대상 제품 구동을 위한 프로그램 : JDK 1.4, Oracle 10g -일반 응용프로그램 : 백신 프로그램 등	-시험 대상 제품의 DB서버 모듈 -DBMS: Oracle v10.2.0.3.0 -시험 대상 제품의 Web서버 모듈 -WebServer : Tomcat v5.0 -JDK v1.5
클라이언트	원격접속프로그램(zterm 등) -일반 응용프로그램: MS-Office, 한글2005, 백신 프로그램 등	-JDK 1.5 원격접속프로그램(zterm 등) -일반 응용프로그램: MS-Office, 한글2005, 백신 프로그램 등	-Internet Explorer 6.0((3)번 클라이언트) -Internet Explorer 7.0((4)번 클라이언트) -일반 응용프로그램: Microsoft 오피스 2003, Acrobat Reader 8.0, V3 7.0 등	-시험 대상 제품 구동을 위한 프로그램 : JDK 1.5 -일반 응용프로그램: MS-Office, 한글2005, 백신 프로그램 등	시험 대상 제품의 클라이언트 모듈 -DBMS: Oracle v9.0.2.0 -JDK v1.5 -일반 응용프로그램 : MS-Office 2003, 아래한글 2005, 백신 프로그램(V3) 등
기타	<스토리지> EMC CX3-20 (부하 PC) 부하 생성 도구 : SilkPerformer 2007	<스토리지> EMC CX3-20 (부하 PC) 부하 생성 도구 : SilkPerformer 2007			
네트워크	10/100/1000Mbps 스위칭 허브				

3. 결함내역 및 속성분석

3.1 결함 내역

품질특성 및 결함속성별로 데이터베이스 소프트웨어에 대한 5종류의 사례를 시험한 결과 결함 건수 및 내역 등을 정리하면 (표 4)와 같다[16,17].

표 4. 데이터베이스 소프트웨어의 결함 내역
Table 4. Defect particulars of Database Software

결함속성	결함 건수	비율	결함 내용	비율
기능성				
기능 오류	65건	35.1%	검색 결과 표시 오류	0.5%
			경계값 처리 오류	5.9%
			내림 및 오름차순	1.1%
			정렬 기능 오류	1.1%
			잘못된 기능 정보 제공	3.8%
			Domain Run 기능 오류	4.4%
			STUFF 함수 기능 오류	0.5%
기능 및 정보 미제공	13건	7.0%	기능 정보 미제공	5.9%
			세션 타임아웃 설정	0.5%
			메뉴 미제공	0.5%
기능 및 정보 표시 미흡	9건	4.9%	기능 정보제공 미흡	3.2%
			날짜 출력 기능 제어 미흡	0.5%
신뢰성				
프로그램 종료 및 중지	5건	2.7%	프로그램 비정상 종료	2.7%
오조작 방지 오류	2건	1.1%	오조작 방지 오류	1.1%
기타	1건	0.5%	비정상적 데이터 파일 소실	0.5%
사용성				
기능 미제공	9건	4.9%	정보 미제공	1.1%
			틀림 미제공	1.1%
			메뉴얼 정보 미흡	1.6%
표시기능 오류 및 미흡	53건	28.6%	정보 표시 미흡	3.8%
			미우스 포인터 표시 미흡	1.1%
			키보드 입력 포커스 제어 미흡	1.1%
			잘못된 오류 메시지 제공	4.3%
잘못된 기능 및 정보제공	15건	8.1%	잘못된 정보 제공	2.2%
			불필요한 레이블 제공	0.5%
기타	3건	1.6%	불필요한 메시지 제공	0.5%
				0.5%
유지보수성				
기타	2건	1.1%	문제 진단 정보 오류	0.5%
			환경 설정 변경 오류	0.5%
이식성				
프로그램 삭제정보 제공 및 미제공	4건	2.2%	프로그램 삭제 오류	1.6%
			프로그램 설치 오류	0.5%
일반적 요구사항				
제품기능 정보 오류 및 미제공	4건	2.2%	제품 정보 미제공	0.5%
			버전 제공 일관성 오류	0.5%

3.2 결함속성 분석

본 연구에서 시험한 데이터베이스 소프트웨어의 결함속성을 품질특성별로 많은 결함이 발생하였던 기능성과 신뢰성, 사용성, 유지보수성, 이식성, 일반적 요구사항의 결함 내역을 살펴보면 (그림 3)과 같다.

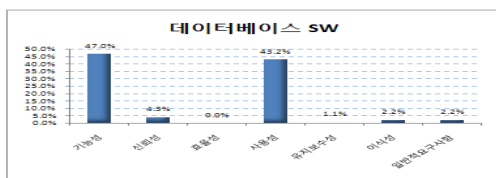


그림 3. 데이터베이스의 결함 내역
Fig. 3. Defect particulars of database

3.2.1 기능성

데이터베이스 소프트웨어의 기능성 관점에서의 결함은 기능 오류가 71.7%로 나타나고 있으며, 기능 및 정보 미제공이 14.9%, 기능 및 정보 미표시가 10.3%로 각각 차지하였다.

결함 속성의 분포에 따르면, 사용하고자 하는 기능 정보가 제공되지 않은 경우의 결함이 대다수로 차지하고 있으며, 임의의 "Domain"을 선택하여 실행중인 상태에서 다른 "Domain"을 선택하여 실행시키는 것이 가능하지만 실제 작업은 수행되지 않은 경우의 결함도 역시 대다수 차지하고 있다.

메뉴얼에서는 DATE 자료가형이 "년/월/일 시:분:초"의 포맷으로 제공된다고 되어있으나, 프로그램에서는 "일/월/년 시:분:초"의 포맷으로 출력되거나 '리포트'클릭 후, 'Ratio' 탭의 'Recommend' 와 'Formula' 컬럼의 오름 및 내림차순 정렬 기능이 동작하지 않은 경우의 결함도 역시 나타나고 있다. 데이터베이스 기능성의 결함분포는 (그림 4)와 같다.

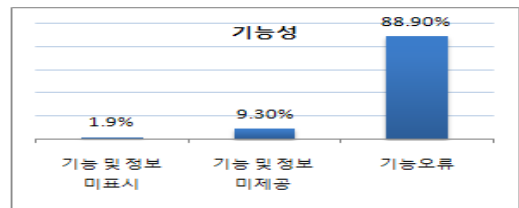


그림 4. 데이터베이스의 기능성 결함 내역
Fig. 4. Functionality defect particulars of database

3.2.2 신뢰성

데이터베이스 소프트웨어의 신뢰성의 결함은 프로그램 종료 및 중지시에 62.5%로 나타나고 있으며, 오조작 방지 오류는 25.0%로 나타나고 기타 결함은 12.5%로 나타나고 있다.

결함 속성의 분포에 따르면 가용 메모리 이상의 메모리를 필요로 하는 select를 수행하면 시스템이 비정상 종료되는 경우와 같이 어떤 기능을 활성화 시키면 프로그램이 종료되는 결함이 대다수 나타나고 있다.

3.2.3 사용성

데이터베이스 소프트웨어 사용성에서의 결함은 표시 기능 오류 및 미흡이 66.3%, 잘못된 기능 및 정보 제공이 18.8%, 기능 미제공이 11.3% 차지하였다.

결함 속성의 분포에 따르면, 프로그램의 정보 제공시 잘못된 정보를 제공되는 결함이 대다수로 나타나며, 그 외 정보 제공 되는 기능들의 정보 역시 충분히 제공되지 않아 정보 제공 및 표시가 미흡한 점이 대다수 차지하고 있다.

3.2.4 유지보수성

데이터베이스 소프트웨어 유지보수성에서의 결함은 모든 오류에 대해 [ERR-91010 : Syntax Error.]라는 동일한 메시지로 출력되어 정확한 문제 진단 정보를 제공하지 않은 결함 오류와 DB 생성시 DB이름을 변경하기 위해 Altibase.properties 파일의 DB_NAME 속성값을 바꾸어도 Create Database시 적용되지 않은 경우의 결함이 나타나고 있다.

3.2.5 이식성

데이터베이스 소프트웨어 이식성에서의 결함은 프로그램 삭제 오류가 높은 비율을 차지하고 있으며, 프로그램을 설치 시 정상적으로 설치가 안되는 경우의 결함이 나타나고 있다.

3.2.6 일반적 요구사항

데이터베이스 소프트웨어 일반적 요구사항에서의 결함으로는 버전 제공 일관성 오류와 기능정보를 제공하지 않는 결함이 다수 나타나고 있다.

3.3 데이터베이스 소프트웨어 시험 관련분석

3.3.1 시험 일정 분석

시험 일정 분석에서는 데이터베이스 소프트웨어 분야에 해당 하는 제품에 대해 시험환경을 구축하기 위해 소요된 일수와 시험에 소요된 기간, 시험 회수를 조사·정리하였다. 이 절에서는 환경구축과 시험 일수 및 회수에 관한 조사 결과를 제시하고 뒤에서 분야별 차이를 분석하여 [표 5]와 같이 나타 내었다.

표 5. 시험 일정 분석표

Table 5. Testing schedule analysis table

구분	환경 구축일수	시험일수	시험회수
DB-1	0.5	21	3
DB-2	1	15	2
DB-3	1	38	3
DB-4	1	23	2
DB-5	1	14	2
평균	0.9일	22.2일	2.4회

위 표와 같이 데이터베이스 소프트웨어 시험 가동시 그 환경을 구축하는데 걸리는 시간을 통계를 내서 보면 보통 0.9일의 시간이 걸리며 시험을 하는데 걸리는 기간을 보면 22.2일이 소요된다. 그리고 결함 시험 회수의 통계를 보면 2.4회로 보통 2~3회에 걸쳐 시험을 마친다는 것을 알 수 있다.

시험일수와 횟수는 직접적으로 소프트웨어의 품질과 연관된 것은 아니지만, 시험일수가 상대적으로 긴 소프트웨어의 경우에는 소프트웨어의 규모가 크거나 논리적으로 복잡하거나 기능 요소가 많은 등 여러 가지 요인이 관여될 수 있다. 시

험횟수의 경우가 상대적으로 많은 소프트웨어는 결함이 다수 발견되어 개선 과정에서 여러번의 시험이 필요했거나 재시험을 통해 해결되지 못한 결함이 다시 발견된 경우이다.

3.3.2 결함 정도 분석

데이터베이스 소프트웨어에서 나온 결함 내용들을 요약한 내용이며 그 결함 정도는 [표 6]과 같다.

결함정도 분포에 따르면 프로그램 운영에는 문제가 없으나, 기능이 정확하게 동작하지 않거나 사용자의 혼란을 야기하는 정도의 결함이 발생하는 경우인 M(Medium)이 61.1%로 대부분을 차지하고 있으며 프로그램 운영에 문제가 없고, 기능도 정확하게 동작하나 권고 사항 수준의 경미한 결함이 발생하는 경우인 L(Low)가 26.5% 그 뒤로는 기능이 정상적으로 동작하지 않거나, 시스템과 하드웨어 혹은 프로그램이 비정상적으로 종료되는 등의 치명적인 결함이 발생하는 경우인 H(High)는 12.4% 차지하였다.

표 6. 결함 정도 분석표

Table 6. Defect degree analysis table

구분	H(High)	M(Midium)	L(Low)
DB-1	2	31	16
DB-2	8	9	15
DB-3	7	45	5
DB-4	0	2	1
DB-5	6	26	12
합계	23	113	49
백분율	12.4%	61.1%	26.5%

전반적으로 Low 결함과 High 결함에 비해 상대적으로 Medium 수준의 결함이 많이 발생하고 있는데, Low 결함의 경우에는 시스템의 동작에 영향을 미치지 않으므로 테스트 과정에서 무시되거나 쉽게 발견되지 않는 측면이 있으며, High 수준의 결함은 소프트웨어의 주요 핵심 기능에서 발견되기 때문이다.

일반적으로 소프트웨어 개발 일정은 정해져 있는 경우가 많으므로 결함 정도의 분석을 통해 한정된 시간 내에 최대한의 개선효과를 획득하기 위해서는 결함수준에 따른 개선 우선 순위를 결정하여 일정에 반영할 필요가 있다.

3.4 데이터베이스 소프트웨어 결함 분석

데이터베이스 소프트웨어에서 가장 많이 나온 결함은 기능성의 기능오류에서 정보 표시 기능오류와 기능 정보 미제공이 높은 비율을 차지하고 있으며, 운영체제 소프트웨어에서 가장 많이 나온 결함은 신뢰성의 프로그램 종료 및 중지에서 프로그램 비정상 종료 및 중지가 높은 비율을 차지하고 있으며 향

후 개발하는 데이터베이스 프로그램에서는 이에 대한 개선방안이 필요하다고 본다.

IV. 결 론

소프트웨어 제품의 품질이 중요한 관건으로 대두된 지 오래이며 소프트웨어 제품 품질에 대한 인증의 중요성이 높아짐에 따라 다양한 소프트웨어 유형에 따른 품질시험 및 인증 방법에 대한 연구가 지속적으로 추진되고 있다.

본 논문에서는 데이터베이스 소프트웨어 제품의 품질 수준을 파악할 수 있는 지표를 도출하여 지표산식을 정의하고 지표의 결과를 산출하기 위해 필요한 수집항목을 선정하며 수집과 분석을 통해 실질적으로 어떤 결함 유형들이 주로 발생하고 있는지를 확인하였으며 데이터베이스 소프트웨어에 대한 시험 평가 모델을 개발하여 시험 사례를 분석하였다.

최근 국내에서 우수한 IT 인프라에 걸맞게 데이터베이스 산업이 규모면에서 충분한 성장을 이루어 냈지만, 데이터베이스 소프트웨어의 품질에 대한 인식을 제고하고 실질적인 품질향상을 위한 제도적인 정책이나 품질향상을 위한 활동이 미미하였다고 할 수 있다.

데이터베이스의 품질을 이루는 요소로 데이터의 안정성이나 빠른 접근성, 무결성 등을 포함하여 유지보수와 관련된 정보의 재활용성 등이 중요한 특성으로 부각되고 있다. 이러한 특성들을 평가할 수 있는 품질평가 방법을 확립하고 데이터베이스 분야의 소프트웨어에 대해 품질평가를 제도화함으로써 국내 데이터베이스 관련 소프트웨어의 품질향상에 기여하여 부가가치 창출은 물론 대외경쟁력 확보를 통해 국산 데이터베이스 관련 소프트웨어의 지속적인 보급에 기여할 수 있을 것이라 사료된다.

또한, 국내 데이터베이스 소프트웨어 시장에서 외산 제품에 대한 의존도가 높은 현실을 타개하기 위해서는 결국 고품질 데이터베이스 소프트웨어 개발을 유도할 수 있는 품질평가 제도의 정착 및 효과적인 평가방법의 개발과 적용을 통해 실질적인 품질향상에 대한 기여가 중요하다고 사료된다.

향후 실질적인 활용을 통해 고품질 소프트웨어의 개발을 촉진함으로써 높은 부가가치를 창출하고 국제적으로 경쟁력을 갖춘 제품의 개발을 지원할 수 있을 것으로 기대한다.

참고문헌

- [1] E. F. Codd, "A Relational Model of Data for Large Shared Data Bank" 1970. 6.
- [2] G. Trajcevski, O. Wolfson, B. Xu, and P. Nelson, "Real-Time Traffic Updates in Moving Objects Databases", Proc. the 13th International Workshop on Database and Expert System Engineering Workshop, pp.3-12, 2002.
- [3] M. F. Mokbel, T. M. Ghanem, and W. G. Aref. "Spatio-Temporal Access Methods", Bulletin of the IEEE computer Society Technical Committee on Data Engineering, Vol.26, No.2, [[.40-49, 2003.
- [4] M. L. Lee, W. Hsu, C. S. Jensen, B. Cui, and K. L. Teo, "Supporting Frequent Updates in R-Tree : A Button-Up Approach", Proc. International Conference on Very Large Data Bases, pp.608-619, 2003.
- [5] E. Frenzos, "Indexing Objects Moving on Fixed Networks", Proc. International Symposium on Spatial and Temporal Database, [[.289-305, 2003.
- [6] Y. Xia and S. Prabhakar, "Q+Rtree : Efficient Indexing for Moving Object Database." Proc. International Conference on Data Engineering, pp.391-402, 2005.
- [7] ISO/IEC 12119, Software Engineering - Software product evaluation - Requirements for quality of commercial -off -the -shelf software products (COTS) and instructions for testing, 1994.
- [8] ISO/IEC 9126-1 : Software engineering - Product quality - Part 1: Quality model, 2001.
- [9] ISO/IEC TR 9126-2 : Software engineering - Product quality - Part 2: External metrics, 2003.
- [10] ISO/IEC TR 9126-3 : Software engineering - Product quality - Part 3: Internal metrics, 2003.
- [11] ISO/IEC TR 9126-4 : Software engineering - Product quality - Part 4: Quality in use metrics, 2004.

- [12] ISO/IEC 14598-1 : Information technology - Software product evaluation - Part 1: General overview, 1999.
- [13] ISO/IEC 14598-4 : Software engineering - Product evaluation - Part 4: Process for acquirers, 1999.
- [14] ISO/IEC 14598-5 : Information technology - Software product evaluation - Part 5: Process for evaluators, 1998.
- [15] 김금옥, 양해술, "임베디드 소프트웨어의 개발 프로세스와 품질평가방법", 한국컴퓨터정보학회논문지 제13권 제6호, 95-105쪽, 2008년 11월
- [16] 이화식, "대용량 데이터베이스 솔루션(I)", 엔코아, 2005.
- [17] 최인용, 양해술, "서비스지향 아키텍처 소프트웨어의 신뢰성 평가 모델", 한국컴퓨터정보학회논문지 제14권 제3호, 149-159쪽, 2009년 3월.
- [18] 양해술, "소프트웨어 시험평가 모듈 개선 연구", ETRI 컴퓨터·소프트웨어 기술연구소, 위탁과제 최종보고서, 2001. 11.



강배근

2004년 : 한국교육개발원 졸업(공학사)
 2008년~현재 : 호서대학교 혁신경영
 기술융합대학원 메카트로닉스학
 과 석사과정 재학 중
 <관심분야> : 소프트웨어공학(특히,
 S/W 품질보증과 평가 및 프로
 젝트관리, CBD기반기술



이하용

1993년 : 강원대학교 전자계산학과 졸
 업(이학사)
 1995년 : 강원대학교 대학원 SW공
 학전공 (이학석사)
 2005년 : 호서대학교 벤처전문대학원
 컴퓨터응용기술학과졸업(공학박
 사)
 1996년~2005년 : 경희대, 강원대, 선
 문대, 호서대 컴퓨터 공학부강
 사
 1995년~2002년 : 한국SW품질연구소
 선임연구원
 2005년~현재 : 서울벤처정보대학원대
 학교 교수
 <관심분야> : 소프트웨어공학(특히,
 S/W 품질보증과 품질평가, 품
 질감리, 객체지향 프로그래밍,
 객체지향 분석과 설계, 컴포넌
 트기반 S/W 개발방법론

저 자 소 개



양해술

1975년 : 홍익대학교 전기공학과 졸업
 (학사)
 1978년 : 성균관대학교 정보처리학과
 (석사)
 1991년 : 日本 오사카대학 정보공학과
 S/W공학전공(공학박사)
 1975년~79년 : 육군중앙경리단 전산
 장교
 1980년~95년 강원대학교 전자계산학
 과 교수
 1986년~87년 日本 오사카대학 객원연
 구원
 1995년~02년 한국S/W품질연구소 소장
 1999년~현재 호서대학교 벤처전문대
 학원 교수
 <관심분야> 소프트웨어공학(특히, S/W
 품질보증과 평가, 품질감리, 프로젝
 트관리, CBD기반기술, IT품질경영