

그리스 차량 판매공사의 문제 해결 사례

김 용 덕 두산정보통신(주) 차장

프로젝트 개요

그리스의 차량 판매공사는 승용차, 트럭, 특수장비 차량 등을 수입 판매하는 업체로서, 인적자원, 회계, 경리 등 많은 양의 자료를 관리해야 하며, 이 자료들을 보관해야 한다.

초창기의 이러한 응용프로그램들은 1980년대 초반에 IBM의 미들레벨 컴퓨터에서 개발되어 졌으며, 아직까지도 사용되고 있는 실정이다.

그러나, 초창기의 응용 프로그램들은 수준 낮은 디자인, 충실하지 못한 주석문, 불충분한 시험 가동 등의 문제점을 현재까지도 가지고 있는 상태이다. 그러나, 초창기의 이러한 시행착오를 통하여 프로그램 기술이 축적되어져 왔고, 점점 구조화 되어 졌으며, 급기야는 방법론으로까지 적립되기에 이르렀다. 사람들은 이러한 방법론에 의거하여 프로그램을 개발해 왔던 것이다.

따라서, 2000년 연도 문제 해결의 핵심요소는 이러한 방법론의 적립이라고 할 수 있을 것이다. 그리고, 이러한 방법론의 핵심 요소는 바로 자세한 주석문이라고 할 수 있겠다.

주석문은 컴퓨터 언어와는 달리 수기로 작성되어진 것이 많고, 이러한 문제 때문에 자동으로 보완되어 지지 않고, CASE tool과 같은 도구를 사용할 수도 없으나, 얼마나 주석문을 자동화하여 CASE tool 등을 통한 자동적인 수정보완이 가능한 수준으로 만들어야 한다.

다른 핵심 요소라면, 표준화를 들 수 있다. 같은 언어와 같은 방법론에 의하여 코딩되어진 프로그램은 그만큼 시간을 줄여 줄 것이므로, 이 역시 2000년 문제의 핵심이라 할 수 있겠다.

2000년 연도문제를 풀어 나가면서, 우리는 또다른 문제점들을 발견·해결하였고, 이와 유사한 초창기의 문제점들을 새로이 발견하게 되었다.

〈 표 1 〉 프로젝트 개요

| 구 분 | 내 용 |
|-----------|---|
| 업 무 개 요 | 승용차, 트럭, 특수장비차량 수입 판매 인적자원, 회계, 경리 등 다량의 자료관리 및 보관 |
| 주 요 문 제 점 | 80년대 초반 방법론없이 개발 및 산출물 표준 부재 |
| 인 원 구 성 | 컨설턴트 2명, 프로그래머 3명, 업무지원 2명 |
| 소 요 기 간 | 95년 여름 ~ 96년 5월(약 1년 소요) |
| 전 제 사 항 | 초창기 개발자 확보 및 전 과정 외부 컨설팅 |

추진현황

작업절차

〈 표 2 〉 작업단계별 절차

| 단 계 | 작업항목 | 작업내용 | 비 고 |
|-----|-----------|--------------------------------|-----------------------------|
| 1단계 | 기 초 작 업 | 프로젝트 계획 수립 DB / 프로그램 조사 | 2개월 소요 컨설턴트 1명 |
| 2단계 | 작 업 정 리 | DB / 프로그램 목록 작성비 사용 프로그램 제거 | 3개월 소요 컨설턴트 1명, 프로그래머 1명 |
| 3단계 | 실 행 | S / W 라이브러리 구축 DB / 코드 변환 | 3개월 소요 전원 참여 |
| 4단계 | 검 사 / 가 동 | 모듈단위 시험 통합 시험 및 운영 | 4개월 소요 전원 참여 |

단계별 추진현황

문제해결 제1단계 - 기초 작업

1995년 여름 1명의 컨설턴트가 2개월의 시간을 투여하여 프로젝트에 대한 계획을 수립했다. DB들 전체 프로그램들에 대한 문헌 및 주석 자료가 전부 다

수집되어 분석되어졌다. 이 작업은 DB와 DB내의 전체 필드들에 대한 대조 작업으로부터 시작되었고, 이 작업의 결과물은 나중에 DB 관리의 교과서로 쓰이기 까지 했다.

그리고 난 후, 전체 프로그램의 각각의 모듈을 조사했으며, 모듈들의 중요점을 분석했다. 모든 프로그램들의 문헌은 중점사항을 중심으로 재기록 되어졌

고, 수기로 기록된 것은 모두 바뀌게 되었다.

우리의 조사결과는 830여개의 DB File들(물리적 파일 171개, 논리적 파일 659개), 5개의 모듈(1,201개의 RPG 프로그램, 1,276 화면 프로그램, 381 CL 프로그램)로 분석되어 졌다.

- (1) 프로젝트 일정 계획
- (2) 문헌자료 조사
- (3) 프로그램 모듈 조사

< 표 3 > 작업대상 분석 결과

| 구 분 | 작 업 대 상 |
|----------|---|
| Database | 830개(논리 : 171개, 물리 : 659개) |
| 모듈수 | 5개 |
| 프로그램 | RPG : 1,201개, 화면 : 1,276개, CL : 381개 |

문제해결 제2단계 - 작업정리

우리는 두번째 단계로 분석되어진 1단계의 결과물들로 목록을 작성하고, 사용하지 않는 프로그램들을 최적화시켰다. 이 작업에는 1명의 프로그래머와 1명의 컨설턴트를 3개월간 투입하여 진행하였다. 나중에 우리가 작성한 목록은 2000년 문제해결시 진도표로 활용되었다.

- (1) 프로그램 및 DB 목록표 작성
- (2) 비 사용 프로그램 검토 및 삭제
- (3) 일정계획표 보완

문제해결 제3단계 - 실행

우선적으로 동일한 컴퓨터에 들어 있는 병렬구조 컴퓨터의 소프트웨어 라이브러리를 새롭게 구축하고, 시험해 보았다. 그후, 지금까지 사용하던 응용프로그램과 자료들을 새로 작성한 것들로 덮어 씌웠다. 물론, 그전에 전체 회사의 자료는 백업을 해둔 상태였다.

그러나, 현재 사용중인 2자리 연도의 DB를 새로운 4자리연도 DB로 변환하는 부분에서 문제(즉, 19와 20을 필드에 삽입하는 문제)가 발생하였다. 이를 위해서는 3GIGA의 자료를 5일동안 멈추지 않고 계속 전송해야 하는데, 이는 현실적으로 어려운 일이다. (에러 발생시 처음부터 다시 해야함)

우리는 소프트웨어 라이브러리 병렬 구조를 새로 구축하기 위하여 첫번째로, 참조 파일들을 고정시키고(하드웨어에서 사용중인 DB 필드 디렉토리), DB에서 작업을 실시하였다.

프로그램을 표준에 의거하여 SOURCE 코드를 고치고 재 컴파일을 실시하였는데, 이때 바뀐 부분은 컨설턴트의 주도하에 전부 종이에 프린트 시켜서 바뀐 부분을 비교했다.

- (1) 자료 백업(회사 전체 자료)
- (2) 소프트웨어 라이브러리 구축
- (3) 소스코드 수정
- (4) DB 변환

문제해결 제4단계 - 검사 및 시험가동

첫번째로, 프로그램을 시험 가동한다. 프로그램을

새로 작성하거나, 변경한 경우 이를 실행한 프로그래머가 검사를 실시한다. 그뒤 컨설턴트가 이를 검증하는 두번째 검사를 실시한다.

그러나, 현실적으로 이 두 단계의 검증작업은 상당한 피리가 있다.

프로그래머는 하나의 커다란 프로그램을 새로 작성하거나 전체적인 수정을 하기 보다는 여러개의 프로그램 중에서 조금씩을 수정하게 되기 때문에 수정보다도 검사작업의 양이 엄청나게 많아지는 것이다. 이와는 반대로 컨설턴트의 경우 각각의 프로그램을 자기가 전부다 작성한 것이 아닌 관계로 이를 일일이 이해하는데 너무나 많은 시간이 소요되게 된다.

그래서, 프로그램의 검사는 프로그래머의 검증 단계를 생략한 컨설턴트에 의한 단한번의 최종 검증으로 결정되었다. 검증시간을 줄이기 위해서는 순차적인 검색이 되어야 할 것이다.

〈 표 4 〉 검사단계 비교

| 검사단계 | 1 단계 | 2 단계 |
|-------|---------------|------------------|
| 검 사 자 | 프로그래머 | 컨설턴트 |
| 검사대상 | 자기가 작성한 프로그램 | 전체 프로그램 |
| 장 점 | 오류발견 및 수정 용이 | 프로그래머의 오류 검증 |
| 단 점 | 수정량 대비 검사량 과다 | 프로그램 이해에 과다시간 소요 |

검사시간 절약을 위해 프로그래머가 프로그램 수정과 테스트를 병행하는 것으로 간주해서 1단계 검사는

생략하고, 2단계 최종 검증만 하는 것으로 확정함. 우리는 검증을, 첫째 모듈단위로, 둘째 회사의 업무처리에 입각한 순차적 프로그램간의 비교로 실시했다. 이 과정동안 프로그래머는 하루 8~10시간 일하면 되는 반면, 컨설턴트는 하루 14~18시간을 투입해야 하였다. 전체적인 작업은 4개월간 지속되었다.

우리는 이 작업을 95년 여름부터 시작하여 96년 5월에 종료하였다.(거의 1년 소요)

- (1) 모듈단위 시험
- (2) 통합시험
- (3) 시험운영

고려사항

대응방안

대응전략 수립

기존 시스템을 그대로 사용할 필요가 있는지 먼저 검토할 필요가 있다. 이럴 경우에는 그리스 차량 판매 공사의 사례와 같이 변환 계획을 수립해서 실행해야 하며, 작업량 감소를 위해서는 불필요한 프로그램 폐기 등 준비를 하는 것이 바람직하다.

만일, 시스템이 노화되어 계속 사용이 불가능할 경우에는 Client /Server 또는 인트라넷 환경으로 재개발을 계획하거나, 전사적 자원관리시스템(ERP) 등 Package에 의한 대체를 도모하는 것이 바람직하다.

재개발 또는 Package에 의한 대체는 별도의 문제이므로 여기서는 기존 시스템에 대한 변환 방법에 대

해서만 검토하고자 한다.

실행 주체 검토

실행주체는 자체 인력 또는 아웃소싱에 의한 외부 인력을 활용할 수 있다. 이것은 인력, 기술, 시간 및 비용 등 제반 사항을 고려해서 결정해야 한다.

작업의 특성이 전체 프로그램을 대상으로 관련된 부분만 수정하는 것이기 때문에, 기존 시스템 개발자 확보 및 내부 인력 주관하에 프로젝트를 진행하는 것이 바람직하다.

실행방안

접근방안 결정

프로젝트의 성공적인 실행을 위해서는 방법론, 변환 지원 Tool 및 처리방식을 사전에 검토해야 한다.

첫째, 방법론은 기본조사, 영향분석, 변환계획, 변환, 테스트 및 사후관리 등 각 단계에 대해서 체계적으로 통합관리가 되어야 한다.

둘째, Tool은 데이터 저장소 구축, 자동 계획/분석/변환 등 처리기능과 제공되는 산출물, 지원 플랫폼 및 비용 등을 검토해야 하며 신뢰성을 확보하기 위해서는 공급자에 대한 평가를 해야 한다. 그러나, 2000년 문제를 완벽히 해결할 수 있는 Tool은 없기 때문에 지나친 신뢰는 삼가는 것이 좋다.

셋째, 처리방식에 있어서는 연도 데이터에 대한 자리수 변경, 센추리 코드 추가 또는 프로그램 모듈에 의한 자동 인식(예를들면 50년을 기준으로 50미만은 1900년대, 50이상은 2000년대로 인식)하는 방법이 있다. 연도 자리수를 4자리로 변환해서 관련 프로그

램을 수정하는 것이 작업량은 많으나 가장 확실한 방법이며, 프로그램 모듈을 이용하는 것은 작업량은 적으나 연도 순으로 Sort가 안되는 문제점이 있다.

시스템에 따라서 이들 방식을 혼용해서 처리하는 방안을 검토할 필요가 있다.

개발 표준 작성

작업의 효율화를 도모하기 위해서는 사전에 산출물 작성 표준, 코딩 표준을 설정해야 하며, 표준 Library를 구축해서 공통 모듈화해야 한다. 소규모의 시험 프로젝트를 선행해서 테스트 셋트를 구축함으로써, 업무 규칙 및 시스템간 인터페이스를 충족할 수 있는 개발 표준을 작성하는 것도 하나의 좋은 방법이 될 수 있다.

또한 4자리 연도를 입력하게 되는 사용자의 불편 해소를 위해서, 연도 2자리 입력에 대한 변환 모듈 개발을 검토할 필요도 있다.

개발 환경 구축

그리스의 차량 판매 공사의 사례와 같이 작업 시간 과다에 따른 테스트 데이터 및 수정 프로그램 버전에 대한 철저한 관리가 필요하다. 따라서 가능할 경우 필요한 장비를 구매 또는 임대해서 기존 시스템에 대한 운영 환경과 개발 환경을 분리하는 것이 좋다.

과다한 데이터 변환 문제를 해결하기 위해서는 DB 유형(마스터, 트랜잭션 파일 등) 및 기간별(일별, 월별)로 구별해서 자료를 세분화해서 검토할 필요가 있다. 이런 구분에 의해서 사전 또는 선별 변환 작업을 수행함으로써 운영 정기 기간 단축을 도모할 필요가 있다. ●