

TDX 1천만 회선 돌파 기념 숨겨진 TDX 개발 이야기(1) -메모콜 프로젝트



TDX 전자교환기는 우리나라 기술개발사에 빛나는 금자탑을 세웠다. 최근에는 1천만 회선을 돌파함으로써 기념비적 족적을 남겼을 뿐만 아니라 정보통신기술에 대한 우리의 자신감을 다시 한번 한껏 돋우고 있다. 1천만 회선 돌파로 더욱 미더워진 TDX교환기. 그 래서인지 개발 주역들에 대한 고생담은 들어도 들어도 실증나지 않는다.

그런데 그 개발사를 이야기하는 데 있어서 매우 중요한 대목이 간과되고 있어 아쉽다. 소프트웨어 부분에 관한 개발내역이나 그 주역들이 가려져 있다는 것이다. 어찌 보면 껍데기인 하드웨어 위주의 편향된 시각으로 쓰여진 전자교환기 개발사는 왜곡된 역사일 수 있다. 또한 후세에 실질적으로 도움이 되어야 할 기술 개발사가 자칫 흥미 위주의 무용담에 그칠 수도 있다는 우려도 있다.

이에 본지는 당시 소프트웨어 개발 현장에 있었던 천유식 박사를 어렵게 초대, 소프트웨어 개발 중심의 TDX 개발사를 다시 써보기로 하였다. <편집자 주>

프롤로그

전자교환기에서 가장 중요한 요소는 컴퓨터이다. 컴퓨터는 호 처리(call processing), 운영 관리, 자체 진단 등의 작업을 수행하는데 그 대부분을 소프트웨어가 담당한다. 이처럼 소프트웨어의 역할은 대단히 중요하며, 전자교환기 개발의 성패는 복잡하고 방대한 소프트웨어를 정해진 기간 내에 개발할 수 있는가에 달려 있다. 때문에 전자교환기 개발에 투입되는 조직도 통상 하드웨어와 소프트웨어 인력이 3:7 정도의 비율로 구성된다. TDX교환기 역시 소프트웨어 분야가 중대한 관건이었다. 또한 전자교환기 소프트웨어 개발은 시스템 소프트웨어를 국내에서 사실상 최초로 개발한 셈이 되는, 매우 중요한 역사적 이정표였다. 그럼에도 불구하고 TDX 관련 소프트웨어 개발에 관한 내역은 지금까지 별로 알려지지 못하고 있다.

필자는 1972년부터 1988년까지 전자교환기 소프트웨어 개발의 책임자 역할을 한 바 있다. 여기서 필자가 전자교환기 소프트웨어 개발의 역사를 조명하려는 뜻은 TDX 개발사의 전체를 밝히는 데 보탬이 되고자 함이다. 사실 지금까지 회자돼 온 TDX 개발에 관한 이야기는 성공한 프로젝트였기 때문에 주로 좋은 면과 누가 개발의 일등 공신이었는가를 위주로 많이 다뤄졌다. 그러나 돌이켜보면 그 뒷면에는 어두운 면도 많았으며 또한 별로 다뤄지지 않은 소프트웨어 분야에 대해 뭔가 말해야 할 필요성을 절실히 느꼈다. 어두운 면이나 함께 했던 동지들을 비판하는 내용이 거론될 수도 있는 글을 쓰는 것은 쉬운 일이 아니다. 이 글이 후세를 위해서나 다음에 이어질 다른 프로젝트를 위해서 보다 도움이 된다고 믿기 때문에 용기를 내었다. 이에 관련된 분들이나 기관들에게 혹시 누를 끼치는 일이 발생할지 모르므로 미리 사과를 드리며, 이 글이 정보 통신 분야의 발전에 이바지할 것을 굳게 믿는 마음에서 감히 양해를

드리고자 한다.

전자교환기 개발은 대부분의 사람들이 알고 있는 것보다 훨씬 이전인 1972년에 관련 개발이 시작되었으며 그 이후 개발 활동이 꾸준히 이어졌다. 흔히 1980년대 초에 처음 개발을 시작한 것으로 알려져 있는데, 이것은 전자교환기 개발에 1980년대부터 참여하기 시작한 소위 실세들이 전자교환기 개발의 역사를 그들 중심으로 기록한 결과이다. 그들은 TDX 개발에 참여하기 이전에는 모두 교환기 개발과 거리가 먼 분야의 경험을 가지고 있었으나, 제5공화국의 사회 분위기에 어울려 전자교환기 개발 분야에 진입하였다. 이런 연유로 TDX-1을 개발하는 동안에 무경험에서 오는 시행 착오를 많이 하였음은 물론 기존의 인력과 많은 갈등을 표출하였다.

제1부 암호명 ‘메모콜 프로젝트’ - 전자교환기의 효시

개척 시대의 이야기

우리나라 전자교환기 개발의 효시는 KIST(한국과학기술연구원)에서 수행한 메모콜(memo call) 프로젝트이다. 이 프로젝트는 1972년 8월부터 1973년 6월까지 약 10개월 간 진행되었다. 청와대에서 핫라인 용으로 쓸 수 있는 교환기, 즉 일종의 전자교환기를 개발하여 줄 것을 요청함으로써 출범한 프로젝트이다. 이 프로젝트는 비밀리에 추진해달라는 요청에 따라 ‘메모콜’이라는 암호명이 붙여졌다.

한국과학기술연구소 시절, 1972년 여름으로 기억된다. SE팀장인 변희성씨(미국 미네아폴리스 거주)가 방식기기연구실에서 요청이 왔으니 같이 가보자고 하였다. 당시 방식기기연구실장이던 안병성 박사(한국전자통신연구원)의 요청이었다. 필자로서는 안박사를 처음 만나는 자리이기도 했다. 내용은 메모콜에 필요한 프로그램을 2주 동안에 만들어 달라는 것이었다.

원래 청와대 핫라인용으로 개발

그 해 2월부터 시작하기로 되어 있었던 프로젝트가 개발 기간 1년의 반을 지지부진한 채 보내고 그때서야 예산이 배정된 것이었다. 이미 방침이 정해진 것처럼 보였으므로 못한다고 할 수는 없었고, 개발 기간을 검토해 보겠다며 방을 나왔다. 우리는 다각도로 머리를 짜냈지만 최소한 3개월이 필요하다는 결론이 나왔다. 안박사와 다시 협의하였다. 그러나 얻어낼 수 있는 기간은 결국 4주였다.

그 날로 부랴부랴 팀을 구성하면서 프로젝트의 수행에 필요한 준비를 시작하였다. 변희성씨가 팀장을 맡고, 이춘희씨(시스템공학연구소 실장), 이대식씨(한림대학교 교수), 양광숙씨, 우치수씨(서울대학교 교수) 및 필자를 구성원으로 팀을 구성하였다. 이들이 국내에서 최초로 전자교환기 소프트웨어와 시스템 소프트웨어를 만든 사람들이다. 이들 중에서 양광숙씨와 필자가 끝까지 메모콜 프로젝트를 수행하였으며, 필자는 계속적으로 다음 프로젝트도 수행하였다. 우치수씨와 필자가 제일 신참이었는데, 형편상 필자가 실질적으로 팀을 맡아 프로젝트를 수행하게 되었다. 그래서 필자가 팀장과 다른 분들의 도움을 받아 시스템 분석을 하여 분야별로 플로우 차트(flowchart)를 그린 후, 각자가 분담하여 프로그램을 하도록 각자에게 그 플로우 차트를 분배하였다.

소프트웨어 개발기간에 주어진 시간 겨우 4주

이처럼 필자와 동료들이 이 프로젝트를 맡게 된 데는 그만한 경험과 역력이 있었기 때문이었다. 그 당시 우리 SE팀은 변희성씨, 우치수씨 및 필자의 세 명으로 구성되어 CDC3300과 새로 들어오게 될 Cyber 72의 SE 역할을 하고 있었다. 그래서 엄격하게 통제되고 있던 컴퓨터실에 자유롭게 출입하고, 컴퓨터에 이상이 있으면 한국CDC의 CE들과 함께 원상 복구시키거나 새로운 버전을 올릴 때는 설치 작업을 하곤 하였다. 그 당시만 해도 KIST에 근무하는 대부분의 프로그래머들은 포트란이나 코볼로 프로그램을 만들고 있었고, 컴퓨터 구조나 어셈블러를 포함한 시스템 관련 내용을 잘 모르고 있었다. 그러므로 시스템과 관련된 오류가 프로그램에 발생하면 디버깅에 무척 애를 먹었다. 그 때, 그런 문제를

해결하는 마지막 보루가 우리 팀원들이었는데 가히 천사와 같은 존재였다.

우리 팀원들은 COMPASS(CDC 어셈블러)를 잘 알고 있었다. 우치수씨와 필자는 치열한 경쟁을 뚫고 입소 시험에 합격한 후, 3개월 간의 혹독한 교육을 거쳐 COMPASS(변희성씨가 강의)의 성적을 포함하여 최상위의 교육 성적을 올렸으므로 SE팀에 배치될 수 있었다. 그러므로 우리 팀이 메모콜 프로젝트를 가장 잘 수행할 수 있다고 생각하여 1차적으로 그 프로젝트에 참여하게 되었다. 다른 분들은 비교적 이 분야에 경험이 있는 분들 중에서 우선적으로 선발되었다. 여기에는 정만영 박사(당시 KIST 제2 부소장)의 요청에 따라 우리 전자계산실의 실장이시던 성기수 박사(동명정보대학교 총장)의 배려가 있었다. 그러나 참여한 모든 사람들에게 원래의 업무가 있었으므로 오랫동안 이 체제를 유지할 수 없었다. 1차 프로그램이 끝날 즈음에 다른 분들은 원래의 업무로 복귀하고 양광숙씨와 필자만이 끝까지 남게 되었다.

어셈블러 잘 하던 인력들이 중심

1차 프로그램을 작성하는 과정에서 첫 작업답게 에피소드도 많았다. 그 가운데서도 특히 기억에 남는 두 가지 이야기가 있다. 첫 번째는 이대식씨의 프로그램이었다. 그 프로그램은 매우 효율적으로 작성되어 메모리를 차지하는 공간(당시는 메모리의 공간을 얼마만큼 차지하느냐가 대단히 중요했다)이 작고, 수행 시간이 짧은 등 아주 훌륭했었다. 그러나 문제가 조금 있었다. 너무 교묘하게 기교를 부렸기 때문에 필자를 포함한 다른 사람이 이해하기가 극히 곤란하였다. 나중에 2차 버전을 만들 때, 도저히 안 되어서 결국 버리고 그 부분의 프로그램을 다시 작성하였다.

다음은 이춘희씨의 프로그램과 관련된 이야기이다. 분당한 분야별로 각자가 프로그램을 하도록 필자가 그린 플로우 차트를 제공하였으나, 프로그램 작성 마감일이 가까워 와도 전혀 의논이나 반응이 없었다. 신참이 고참에게 독촉을 할 수도 없고 해서 필자 혼자 끙끙거리고 있는 중에 마감일이 되었다. 이춘희씨는 그 날이 되어서야 작성한 프로그램을 필자에게 넘겨주었는데, 나중에 다른 프로그램과 연결하여 시험한 결과 플로우 차트와 한 명령어도 틀리지 않아 디버깅이 전혀 필요하지 않았다. 필자만 쓸데 없는 걱정으로 애태웠던 것이다. 그 이후 필자가 이 분야에서 일을 계속하면서 보아 왔지만 그런 일은 처음이자 마지막이었다.

연구실의 떡방아 소리

1972년 가을 KIST 본관 2층에서 '공공공' 떡방아를 찧는 소리가 울려 나오기 시작하더니 밤낮없이 찧어 대고 있었다. ASR 33 텔레타이프teletype의 입출력 속도가 초당 약 10~15자 정도여서, 한 페이지 분량의 자료를 읽거나 프린트하는 데도 꽤 시간이 걸렸다. 그러므로 거의 종일 ASR 33 텔레타이프의 입출력 소리가 그치지 않았다. 텔레타이프의 동작하는 소리가 마치 떡방아를 찧는 소리와 비슷하다고 하여 연구실에 떡방앗간을 차렸다고 놀림을 받기도 하였다. 조용하던 연구실에 난데없이 나타나 밤낮없이 시끄러운 소리를 내고 있으니, 주위 연구실의 사람들은 은근히 다른 곳으로 옮겨 주기를 바라고 있었다.

메모콜을 개발할 때의 개발 환경은 지금은 상상하기도 어려운 환경이었다. 그 때 컴퓨터 장비로는 NOVA 미니 컴퓨터와 ASR 33 텔레타이프가 전부였고, 프로그래밍 도구로는 어셈블러assembler와 2진 로더loader가 전부였다. 그 때의 개발 환경에서 프로그램을 만드는 과정을 간략하게 설명하면, 코딩 용지에 코딩→종이 테이프에 천공→어셈블링 결과를 종이 테이프에 출력→2진 로더를 통한 어셈블링 결과 로딩→시험과 디버깅→프로그램 완성 등이다.

좀더 자세히 설명하면 다음과 같다. 먼저 코딩 용지에 프로그램을 쓰고, 그것을 텔레타이프의 키key를 통해 입력하면 종이테이프(paper tape)에 ASCII 형태로 천공이 된다. 종이테이프에 천공된 프로그램을 입력하면서 어셈블러를 통해 어셈블링하면 2진 형태로 종이테이프에 천공되어 출력된다. 이와 같이 출력된 종이테이프를 텔레타이프에 걸고 2진 로더를 통해 메모리에 로딩 시키면 원하는 목적 코드가 컴퓨터에 실리는 것이다. 이것을 시험하고 디버깅debugging하면 대망의 프로그램이 완성된다. 지금 생각하면 차라리 웃고 말, 번거롭고 힘들고 시간이 걸리는 과정이었다.

프린트 문제가 작업 지연시킨 주범

이 과정에서도 가장 개발 환경을 어렵게 한 것은 프로그램을 프린트하는 일이었다. 텔레타이프의 입출력 속도가 초당 10~15자 정도인 것이었다. 주석(comment)을 제대로 붙인 명령어는 1초에 한 줄도 프린트하지 못하는 셈인데, 대체로 분당 30~40줄 정도 프린트한다고 보면 된다. 좀 긴 모듈의 프로그램들을 프린트할 때는 밤에 퇴근하면서 프린트를 시작해놓고 이튿날 아침에 결과를 거두는 경우도 꽤 많았다. 종이테이프가 엉키거나 하여 이튿날 아침에 거두어야 할 프로그램이 다 프린트되지 않은 상태에서 중단된 경우의 낭패감이란 이루 말할 수 없었다. 웬만하면 프로그램을 새로 프린트하는 일은 하지 않았기 때문에, 프린트된 프로그램은 수정분 투성이에다 누더기 같이 되는 경우가 많았다. 천공된 종이테이프도 새로 천공하기가 힘들어 오래 붙여서 사용하였기 때문에 입력하다가 걸리는 경우도 많았다.

제일 어려울 때가 디버깅 즉 오류를 추적하여 정정하는 과정이었다. 디버깅을 하기 위하여 프로그램을 동작시키면 텔레타이프를 통해 한 시간이 넘게 걸려서 로딩한 메모리의 내용이 순식간에 깨져 버리는 것이었다. 한 순간의 디버깅을 위하여 한 시간이 넘게 로딩 하는 등의 준비를 하여야 하니 너무 힘이 들어 동작시키는 것을 한 명령어 단위로 해보기도 하였다. 그러나 이렇게 되면 이걸 컴퓨터를 사용하는 것이 아니라 단순한 기계를 조작하는 것과 다를 바가 없다. 단순히 디버깅을 준비하기가 힘들어서 그렇게 한 것만은 물론 아니다. 또 다른 이유는 범용인 NOVA 컴퓨터를 전자교환기 같은 특수한 용도에 적용하는 과정에 필연적으로 발생하는 문제에서 비롯된 것이다.

원시적 개발 환경으로 생고생

그 당시 우리는 기계어(machine language) 디버깅 방식을 많이 사용하였다. 컴퓨터의 패널panel을 이용하여 직접 디버깅을 하는 것을 의미하는데, 컴퓨터의 패널에 부착된 스위치를 조작하면서 디버깅을 하는 방식이다.

NOVA 컴퓨터의 토글 스위치toggle switch를 통하여 필요한 번지나 데이터를 입력하고 수행시키는 것이었다. 매우 원시적인 방법처럼 보이나 운영체제나 그것에 준하는 프로그램을 개발할 때는 위력을 발휘하기도 한다. 이 방식은 다음에 사설전자교환기를 개발할 때도 많이 활용되었고, 또한 필자가 뒤에 114 번호 안내 전산 시스템을 개발하면서 HP1000 컴퓨터의 RTE 운영체제를 고칠 때도 활용되어 진가를 발휘하였다.

지금은 상상도 하기 어려운 개발 환경인 셈이다. 필자는 그 당시 그 일을 하기 이전에 CDC 3300을 사용한 경험이 있었다. 훨씬 더 좋은 도구와 장비를 쓰다가 이와 같은 원시적인 도구와 장비를 써야 하는 것에서 필자는 개발 환경의 중요성을 뼈저리게 느꼈다.

그 당시에도 그 정도의 도구와 장비보다는 훨씬 더 좋은 것을 얼마든지 갖출 수 있었으나, 경험 부족과 준비 부족으로 그렇게 된 것이다. 얼마 뒤에 초당 300자의 속도를 가진 종이테이프 판독기(paper tape reader)를 사용할 수 있게 되어 디버깅이 훨씬 수월해졌는데, 그 정도는 처음부터 준비하여야 하였다. 그 때의 그 기쁨이란 지금도 잊지 못하고 있다. 정말 살 것 같았다.

첫 전자교환기 K1T-CCSS 등장

프로젝트가 진행됨에 따라 소프트웨어(NOVA 컴퓨터)와 하드웨어(교환기 본체)의 통합과 시스템 시험이 뒤따랐다. 하드웨어 부분도 처음 만든 것이어서 자체 시험이 영성하였으며, 시스템 시험을 위한 시스템 차원의 정리나 준비도 형편없었다. 혹시 잘 될지도 모른다는 기대감에 부풀어 모두가 은근히 기대하면서 소프트웨어와 하드웨어의 통합 시험을 시작하였으나 결과는 실패였다. 소프트웨어와 하드웨어가 서로를 알아보지 못하고 전혀 반응을 보이지 않았다. 소프트웨어는 소프트웨어 나름으로, 하드웨어는 하드웨어 나름으로 각각 확인하였는데 양측 모두 이상이 없었다. 그러니 아마 상대방에 이상이 있을 것이라는 주장만 나올 뿐이었다.

사흘 간을 밤낮으로 찾은 끝에 새벽녘에 원인을 겨우 찾았는데, 범인은 입출력 장치(I/O device) 번호였다. 10진법과 8진법의 혼동으로 장치 번호가 틀리니 인터럽트interrupt를 인식하지 못하여 데이터를 주고받을 수 없는 것이었다. 코드를 바꾸는 것으로 간단히 해결하니, 그 때부터 컴퓨터와 교환기 간에 데이터를 주고받을 수 있게 되었다. 간단하기 짝이 없는 문제를 가지고 그렇게 헤매었다고 생각하니 허

탈하기까지 하였으나, 디버깅이란 것이 다 그런 것 아니겠는가. 필자는 그 이후부터 소프트웨어와 하드웨어를 통합할 때 항상 장치 번호를 먼저 확인하는 버릇이 생겼다.

성탄절에도 철야 작업

그 무렵, 때는 크리스마스와 연말연시로 젊은 사람들에게는 황금의 시기였다. 그러나 우리는 매일 거의 철야 작업에 파묻혀 있어야 했다. 시스템 시험을 하는 방법은 대체로 기능을 수행시켜 본 후, 제대로 안되면 소프트웨어 측에서 징후를 찾아내는 방식이었다. 정정은 먼저 소프트웨어 분야인지 아니면 하드웨어 분야인지를 알아내고, 다음에 해당 분야에서 수정하곤 하였다. 결과적으로 시스템 시험은 주로 소프트웨어 측에서 하고 하드웨어 측은 소프트웨어 측에서 징후를 찾아낼 때까지 무한정 기다리는 형태였다. 소프트웨어 측에서 징후를 찾아내면, 비상 대기조 같이 대기하고 있던 하드웨어 측도 달라붙어 같이 정정하곤 하였다. 기다림에 무료하진 하드웨어 측은 곧잘 잡기 놀이를 하거나 술을 마시기도 하였는데, 소프트웨어 입장에서는 그것이 오히려 편했다. 옆에서 지켜보고 있으면 서로 피곤하기 때문이다. 이런 상태가 해를 넘겨 몇 달 동안 계속되었다.

이 과정에서 안병성 실장을 포함한 하드웨어 측의 고생이 많았는데(우리는 디버깅하느라 느낄 겨를이 없었다). 특히 하드웨어 측에서 제어 부분을 맡고 있던 여재흥씨(한화정보통신 전무)의 고생이 많았다. 프로젝트의 중요성이나 프로젝트에 대한 애착심 때문에 정만영 박사의 관심도 대단했다. 밤늦게는 “잘 되어가?”, 아침 일찍엔 “좀 진전 있어?” 식의 물음을 거의 습관적으로 되풀이 하면서 하루에도 몇 차례씩 시스템 시험 장소에 들렀다. 그 분은 원래 소프트웨어 자체를 잘 아는 편이 아니었으나, 소프트웨어의 중요성을 그 때부터 피부로 경험하신 것 같다. 한국통신기술연구소 시절에 필자가 114 번호 안내 전산 시스템을 개발할 때와 국철전자교환기를 개발할 때도 소장이었던 그 분은 필자를 포함한 소프트웨어 측의 주장을 항상 귀담아 들었다.

73년 봄 메모콜 프로젝트 완료

이렇게 하여 메모콜 프로젝트는 1973년 봄에 일단 마무리되고 개발된 전자교환기의 이름을 K1T-CCSS라 칭하였다. 좀더 자세히 K1T-CCSS에 대해 설명하면 다음과 같다. K1T-CCSS는 480회선을 주교환기와 4대의 종교환기로 나누어 처리할 수 있는 전자교환기이다. 주교환기에 240회선, 4대의 종교환기에 각각 60회선을 수용할 수 있다. 시분할(time division) 방식의 전전자교환기로서 제어 컴퓨터는 NOVA 1200을 사용하였다. 여기서 1200은 메모리 사이클로서 1200ns를 의미한다. K1T-CCSS에 사용된 메모리의 크기는 12K words(1word=16bits, 프로그램의 크기는 약 10K words)였다. 주요 기능은 먼저 개인 번호 방식에 의거 가입자가 위치를 옮길 때 위치 등록을 하고, 피호출자를 연결할 때는 가능성이 가장 높은 번호부터 수색하는 기능이다. 우선 순위 방식에 의거 가입자를 A, B, C의 3등급으로 구분하고, 상위 계층의 우선적인 통화를 확보하기 위하여 하위 계층의 통화를 제한하거나 차단하는데 이 경우 음성으로 통보를 한다. 최상위 등급의 경우, 50회선까지 연결하여 음성 회의를 할 수 있고 회의 도중 필요하면 회의 기능을 잠시 중단하고 특정한 사람과 협의를 한 후 회의를 다시 계속할 수 있다. 단축 다이얼 기능도 있었다.

국내 최초의 시스템 소프트웨어 탄생

K1T-CCSS의 프로그램 자체는 NOVA 컴퓨터의 소프트웨어를 전혀 사용하지 않았다. 이것이 K1T-CCSS의 프로그램을 만든 사람들을 국내에서 최초로 시스템 소프트웨어를 만든 사람들이라고 부르는 이유이다. K1T-CCSS의 경우 비록 기능이 제한되고 소형이기는 해도 운영 체제(OS)의 역할을 수행하는 프로그램을 만들었으며 다른 프로그램들도 이 프로그램과 매우 밀접하게 연결되어 있었기 때문이다. 일반적으로 전자교환기 소프트웨어를 컴퓨터 내장 시스템(computer-embedded system)에 적용하는 특수 응용 소프트웨어로 분류하고 있는데, 이것은 전자교환기 소프트웨어가 매우 대형이어서 시스템 소프트웨어가 차지하는 비율이 상대적으로 작아진 결과이다. 다음에 K1T-CCSS의 프로그램을 루틴 routine별로 설명하였다.

각 버퍼buffer와 큐 테이블queue table을 찾아가면서 항상 돌고 있는 것이 Main Loop Routine이다.

Real Time Clock Routine을 포함하여 주기적으로 시간에 맞추어 수행이 필요한 루틴은 이 루틴의 제어를 받는다. Interrupt Routine은 시스템에 인터럽트가 걸렸을 때 처리하는 루틴이다. Real Time Clock Routine은 real time clock에서 100ms마다 인터럽트가 걸리면 호출되어 시간에 관련되는 사항을 처리하는 루틴이다. Data Input/Output Routine은 컴퓨터가 외부 회로에서 데이터를 받아들이거나 외부 회로로 데이터를 내보내는 루틴이다. 이 루틴들이 K1T-CCSS의 프로그램에서 운영체계의 역할을 한다고 볼 수 있다. 그 외에 기능 수행에 관련되는 많은 루틴들이 있으나 설명은 생략하겠다.

사설전자교환기, TDX-1 등의 바탕

이 프로젝트의 결과는 시스템의 안전성 문제로 상용화되지 못하였다. 그러나 전자교환기 답게 그 풍부한 기능은 기존의 기계식 교환기와는 비교를 할 수 없을 정도로 다양하였다. 이것은 소위 말하는 컴퓨터 시스템 개발 방법론에서 발전의 1단계인 기능의 완성 단계에 해당한다. 다시 말해 이 K1T-CCSS는 우리나라에서 개발한 최초의 전자교환기인 것이다.

뿐만 아니라 이 프로젝트는 그 후 사설전자교환기(PABX)를 개발하는데 결정적인 역할을 하였다. 사설 전자교환기는 최초로 KIST에서 개발하여 삼성GTE통신에서 상용화 하였으며, 동양통신(주), 금성통신(주) 등에서도 이 기술을 모태로 하여 독자의 사설전자교환기를 개발한 바 있다. 이 사설전자교환기 개발의 의의는 매우 크다. 전자교환기에 핵심적인 개념인 시분할 방식이 성공적으로 도입되었고, 전자교환기 소프트웨어의 골격이 만들어진 것이다. 그 다음에 개발되는 국설전자교환기인 TDX-1은 이것을 바탕으로 이루어진 것이다.

TDX 1천만 회선 돌파 기념 숨겨진 TDX 개발 이야기(2) -사설 전자교환기 개발

제2부 사설 전자교환기 개발



자랑스런 얼굴들

1973년 6월, 메모콜 프로젝트는 K1T-CCSS의 개발 보고서를 제출하는 것으로 모든 작업을 끝내면서 막을 내렸다. 그러나 KIST나 프로젝트 책임자 입장에서는 개발한 결과도 아까울뿐더러 6천만 원이라는 자금을 이미 써버렸으니, 어떻게 해서든지 메모콜 프로젝트의 결과를 활용할 방안을 찾아야 했다.

그래서 메모콜 프로젝트 팀을 완전히 해체하지 않은 상태에서, 프로젝트의 결과를 활용할 방안을 찾는 한편으로 다음 프로젝트를 준비하도록 하였다. 필자도 Cyber 72의 SE역학을 하면서, 언제든지 다시 참여할 수 있는 상태로 다음 프로젝트의 준비를 하고 있었다.

당시 우리는 메모콜 프로젝트를 수행하면서 뼈저리게 그 필요성을 느꼈던 컴퓨터 과학 분야의 기본 지식을 획득하려고 무척 애를 썼다. 국내의 컴퓨터 과학 분야는 프로그래밍 언어의 사용법을 제외하고는 거의 황무지나 다름이 없기 때문이었다. 국내에서 컴퓨터 과학 분야의 지식을 획득하려면 스스로 공부하여 자력으로 해결하는 길밖에 달리 방법이 없었던 시기였다.

자력으로 컴퓨터 과학 지식 획득

이때 먼저 공부를 시작한 책이 H. W. Gschwind의 「design of digital computers」였는데, 몇 명이 미리 공부했다가 일과 시간 이후에 세미나를 하는 형식으로 진행하였다. 지금 기억으로도 무척 어려워 무지무지하게 고생한 생각이 난다. 그 다음이 J. J. Donovan의 「systems programming」이었는데, 앞의 책에 비해 비교적 무난하게 진행할 수 있었다.

우리는 다음 프로젝트에 필요한 프로그래머를 확보하는데 직접 참여하기도 했다. 그 해 가을에 있었던 프로그래머 공개 채용에서 인력을 확보했다. 우리 팀에서 프로그래머에 대한 수요가 가장 많았으므로, 채용에 합격한 예비 프로그래머들의 교육 책임자를 필자가 맡아 직접 교육을 시켰다. 덕분에 교육이 끝날 즈음 우리 팀에 필요한 프로그래머들을 우선적으로 선발할 수 있었다. 필자는 그 당시의 인연 때문에 그 때 교육을 받은 사람들로부터 그 이후에도 ‘교장 선생님’이란 호칭을 듣게 되었다.

한편 그러는 동안에도 다음 프로젝트의 계약이 진전되고 있어서, 1974년 초에 소프트웨어 개발을 담당할 팀을 정식으로 구성하였다. 이 팀이 사설전자교환기 KIST-500의 소프트웨어 팀이며, 필자가 정식으로 팀장이 되었다. 그 당시 우리 팀은 ‘NOVA 팀’이라고 불렸으며, 전자계산실 내에서 최정예 팀이었

다.

그 때의 자랑스러운 얼굴들로 주혜경씨(현 삼성SDS 이사), 김동주씨(현 삼성전자 전무), 윤병흥씨(현 KT연구개발원 팀장), 한영철씨(현 삼성전자 상무) 및 고건씨(현 서울대학교 교수) 등을 들 수 있다. 이들은 대부분 프로젝트가 끝날 때까지 참여하였다. 이들이야 말로 국내에서 시스템 소프트웨어 개발을 체계적으로 시작한 얼굴들이며, 전자교환기 소프트웨어 개발을 본격적으로 시작한 사람들이다. 지난 호에 소개한 K1T-CCSS가 '시작'에 의의를 둘 수 있던 시험적인 전자교환기였다면, KIST-500은 상용화를 염두에 두고 본격적이고 체계적으로 개발을 추진한 사설전자교환기이기 때문이다.

소프트웨어팀은 '최정에 팀' 인식

이것을 단적으로 나타내는 것이 프로그램의 크기인데, K1T-CCSS의 프로그램이 사용하는 메모리의 크기가 약 10K words인 반면, KIST-500의 프로그램이 사용하는 메모리는 그 크기가 약 26K words였다. 그 이후에 이승호씨(현 한국전산원 연구위원)와 차백추씨(현 현대전자 이사)가 참여하였고, 계속해서 홍영식씨(현 동국대학교 교수), 차운옥씨(현 한성대학교 교수), 조옥경씨, 정왕호씨(현 한화정보통신 이사) 등이 참여하였다. 이들 대부분은 그 동안 우리나라의 전자교환기 개발과 정보 통신 분야의 발전에 크게 공헌하였고, 지금도 중요한 위치에서 각자의 역할을 다하고 있다.

한편 팀을 구성하였으나 아직 프로젝트의 계약이 진행 중이어서 우리의 일과는 주로 계약될 프로젝트에 대한 준비와 공부를 하는 것으로 이루어졌다. 컴퓨터에 관한 책으로 「How To Use Nova Computer」와 J. J. Donovan의 「systems programming」을 주로 공부하였고, 전자교환기에 관한 것으로 K1T-CCSS와 미국이나 일본의 전자교환기에 관한 자료들을 뒤적였다. 또한 개발 환경의 경우도 NOVA 컴퓨터의 개발 환경을 최대한 활용할 수 있게 하면서 필요한 것은 일부 개발도 하는 등 K1T-CCSS에 비해 월등히 좋게 구성하였다.

이렇게 최고 수준의 인력을 확보하고 교육, 훈련, 개발 환경 등의 준비를 철저히 하였으므로, 프로젝트가 시작되지만 하면 아주 잘할 수 있을 것 같았다.

KIST-500의 개발

드디어 계약이 체결되어 미국 GTE사로부터 50만 달러의 연구비를 받아 1974년 4월부터 2년 간 사설 전자교환기 개발을 할 수 있게 되었다. 당시로서는 50만 달러라는 어마어마한 액수의 연구비를, 그것도 외국 업체로부터 받는다는 것은 엄청난 사건이었다. 그 때 50만 달러의 외화는 꽤 큰 돈이었으며, 결과적으로 국가의 재정에도 기여하게 되었다. 이렇게 시작하여 개발한 500 회선 규모의 구내용 사설전자교환기의 이름이 'KIST-500'이다. KIST-500 개발 프로젝트는 애초 계획보다 좀 지연되어 1977년 봄에 완료되었으며, 연구비도 6만 달러 정도 더 받았다.

어떻게 해서 GTE사가 50만 불이라는 거액의 연구비를 투자하게 되었는지에 대한 해석은 구구하다. 물론 메모콜 프로젝트의 결과인 K1T-CCSS가 큰 역할을 한 것은 사실일 것이다. 그러나 그것만 가지고 GTE사가 그와 같은 거액을 선뜻 투자한 것을 설명하기는 미흡하다고 생각된다. 필자의 견해로는 아마 KIST-500 프로젝트가 한국의 국설전자교환기 시장에 진출하는데 필요한 교두보로 활용할 가치가 충분하다고 본 것 같다. 그 때 우리나라의 전화 적체는 심각한 상태였으며, 이 문제를 해결하기 위해서 전자교환기를 외국으로부터 도입할 가능성이 매우 높은 분위기였다.

미 GTE 50만 달러 개발비 선뜻 지원

이러한 분위기는 프로젝트를 수행하는 데도 적지 않은 도움을 주었다. 이미 전자교환기를 개발한 선진국의 교환기 회사들이 그것을 한국에 팔고 싶었기 때문에 우리에게 호의적이어서, 전자교환기의 첨단 자료를 수집하는 것이 아주 용이하였다. 필자가 프로젝트를 수행하는 동안에 벨기에의 BTM사에 가서 ITT의 M10C 교육을 1975년 1월부터 4월까지 4개월간 받을 수 있었던 것도 이 때문에 가능하였다. 물론 교육에 의해 알게 된 새로운 지식들은 그때그때 KIST-500 개발에 반영하였다.

KIST-500의 소프트웨어 개발은 비교적 순조롭게 진행되었다. 우수한 프로그래머들을 뽑았으며, 사전에 교육과 훈련을 하는 등 미리 준비를 잘하였기 때문이다. 이미 필자가 전자교환기 개발에 한 번 참여

했던 것도 좋은 경험이 되었다. 필자는 소프트웨어 전체에 대한 개략(general) 플로우 차트와 데이터 테이블data table을 그려 임무에 따라 각자에게 배부하고 계속적으로 보완하면서 소프트웨어 전체를 관리하였다. 상세(detailed) 플로우 차트와 프로그램, 그것에 따른 시험은 임무를 맡은 각자가 담당하였다. 일의 내용도 재미가 있었고 프로젝트와 팀에 대한 긍지도 높아 모두 매우 능동적으로 열심히 일하였다.

소프트웨어 개발 방법론의 필요성 느껴

GTE사는 정기적으로 검토(review)를 하였는데, GTE사의 전문가들은 그 때마다 새로운 요구사항, 문서화, 표준화 등을 주문하였다. GTE사의 새로운 요구사항과 전자교환기에 대한 우리의 지식이 늘어남에 따라 소프트웨어가 점점 커지고 복잡하게 되었다. 이에 따라 필자가 맡은 소프트웨어 전체의 관리가 점점 어려워져 나중에는 혼자서 감당하기가 힘든 지경에 이르렀다. 그 당시에는 다른 방법이 없었기 때문에 근근히 프로젝트를 마치기는 했다. 그러나 더 큰 프로젝트는 이 방법으로는 효율적인 수행이 불가능함을 깨달았으며, 소프트웨어 개발 프로젝트의 체계적인 관리가 필요함을 절감하게 되었다.

이 때부터 필자는 소프트웨어 개발 방법론에 관심을 갖기 시작하였다. 결국은 이 분야를 전공하게 되어 석사와 박사학위 논문도 이 분야에 대해 쓰게 되고, 지금까지 이 분야에 관한 세 권의 전문 서적도 집필하게 되었다. K1T-CCSS가 컴퓨터 시스템 개발 방법론에서 발전의 1단계에 해당하는 기능의 완성이라면, KIST-500은 2단계에 해당하는 컴퓨터의 2중화와 결함 포용(fault tolerant) 설계를 포함하는 신뢰성의 완성이라 볼 수 있다.

GTE사의 검토와 인수 시험 과정에서 우리는 번번이 곤욕을 치러야 했다. GTE사 사람들은 전자교환기 분야의 전문가들로 구성되었으므로 우리의 약점들을 잘 알고 있었기 때문이다. 문서화, 표준화 등은 우리에게 생경한 분야였기 때문에 번번이 지적을 받았다. 시험을 할 때도 그랬었는데, GTE사 사람들은 시험하는 과정에서 어떻게 하면 우리가 만든 전자교환기에 오류가 나올 것임을 매우 잘 알고 있었다. 교묘한 방법이나 조합을 동원하여 오류가 나오게 하여 우리를 난처하게 하였고, 버튼button 전화기의 버튼을 문지르듯이 누르면 우리가 시험할 때는 안 나오던 오류가 나오곤 하였다.

최초의 국산 컴퓨터 '세종 1호' 등장

KIST-500 개발 프로젝트와 관련하여 빠뜨릴 수 없는 것이 최초의 국산 컴퓨터인 '세종 1호'이다. 세종 1호 컴퓨터는 NOVA 컴퓨터의 호환 컴퓨터이다. 마이크로프로그램으로 제어되기 때문에 NOVA 컴퓨터와는 전혀 다른 컴퓨터이나, NOVA 컴퓨터의 모든 명령어를 수행할 수 있었고 전자교환기에 편리한 명령어를 일부 추가한 컴퓨터이다. 그 당시로는 불가피한 선택이었는데, 상용화되는 전자교환기에 NOVA 컴퓨터를 그대로 장착하는 것은 문제가 많았기 때문이다. 그 때 운영 체제, 컴파일러compiler, 로더, 디버거debugger 등의 시스템 소프트웨어를 개발할 능력이나 여력이 있었다더라면, KIST-500 개발 프로젝트는 우리나라 컴퓨터 개발을 크게 앞당기는데 기여하였을 것이 틀림없다. 세종 1호 컴퓨터의 개발에는 강진구(姜鎭求)씨(현 삼보마이크로시스템 사장)가 주요한 역할을 한 것으로 기억하고 있다.

가열된 스카우트 열풍

KIST-500 개발 프로젝트는 우리나라에서 전자교환기 개발의 새로운 장을 여는 계기가 되었다. 이것을 계기로 관련 회사들이 앞 다투어 사설전자교환기 개발에 착수하였다. 금성통신(주)에서는 GS-100의 개발을 최용일씨(현 LG정보통신 중앙연구소 소장)가 맡아 수행했고, 동양정밀공업(주)에서는 OPEX-50/100의 개발을 KIST-500의 제어 부분을 담당하다가 1976년 초 옮겨 간 여재흥씨가 맡아 수행했으며, 삼성 그룹은 GTE사와 합작 투자의 형태로 전자교환기 분야에 진출을 시도하고 있었다.

이렇게 되니 얼마 안 되는 전자교환기 개발 요원에 대한 스카우트 열풍이 불게 되었고, KIST-500 이후의 프로젝트가 구체화되지 못한 상태였으므로 프로젝트에 참여하고 있는 사람들이 동요하기 시작하였다. 여재흥씨는 이미 스카우트되어 갔고, 필자도 그런 와중에 휩쓸리게 되었다.

필자와 제일 먼저 연결된 회사가 동양정밀공업이다. 먼저 간 여재흥씨의 영향도 있었고, 동양정밀공업과 ITT사의 합작 투자가 내정된 상태였으므로 ITT사의 훈련을 받고 온 필자에게는 자연스러운 연결이

었다. 그러나 이것은 GTE사와의 계약과 관련, KIST 내에서 문제가 제기됨으로써 성사되지 못했다. 애초 계약 시 KIST와 GTE사는 원만한 기술 전수와 개발품의 상품화를 위해서 GTE사가 합작 투자하는 생산 회사에 개발품과 각 분야의 핵심 요원들이 함께 넘어가도록 하는 조건에 합의했기 때문이다. 즉, GTE사가 곧 삼성 그룹과 합작 투자하여 회사를 만들 것이니 삼성으로 가도록 하라는 것이었다.

삼성에서 GTK-500 상용화

고민 끝에 필자는 1976년 9월 삼성전자(주)의 과장급 연구원으로 옮겨 갔다. 그 때는 이미 KIST-500의 개발이 마무리 단계에 있었으며, GTE사가 마지막 인수 시험을 하고 있었다. 필자는 삼성전자에서 합작 회사의 설립을 준비하는 한편, KIST에 위촉 연구원으로 나가면서 KIST-500 개발의 마무리 작업에 참여하였다.

1977년 3월이 되자 KIST-500과 더불어 이주형씨(현 삼성전자 고문), 강진구(姜鎭求)씨 등과 함께 소프트웨어 측에서도 김동주씨, 윤병흠씨 및 한영철씨가 삼성전자로 넘어왔다. 그 때는 합작 투자에 대한 절차가 추진 중이었기 때문에 삼성전자 내에서 팀 형태의 임시 조직으로 있었다. 1977년 말에 모든 절차가 완료되어 드디어 삼성GTE통신(주)이 탄생했는데, 이 회사가 삼성 그룹이 통신 분야에 진출하여 세운 최초의 회사이다. 강진구(姜鎭求) 당시 삼성전자 사장이 겸직으로 사장을 맡고, 김영한 이사(작고), 이주형 부장, 강진구(姜鎭求) 과장(하드웨어), 필자(소프트웨어) 등이 주요 간부였다. 물론 GTE 측의 중역과 간부도 있었으며, 중요한 결정은 반드시 합의하여 결정하는 방식이었다. KIST-500을 개발한 사람들 중에서 안병성 실장, 박항구씨(현 현대전자 부사장) 등 일부만 남았고, 소프트웨어 팀은 해체되었다. 그 당시 핵심 요원으로서 유일하게 혼자 남은 박항구씨가 진로 때문에 무척 고민한 것으로 알고 있다.

끊임없는 소프트웨어 보완으로 기술 축적

삼성GTE통신에서 KIST-500을 정식으로 인수하여 붙인 이름이 GTK-500이다. 우리는 GTK-500 사설 전자교환기를 대대적으로 보완하고 회선 용량을 늘려 '센티넬' SENTINEL이란 상표를 붙여 상품화하였다. 주요 보완 내용은 회선 용량을 1,500 회선으로 늘리고 호텔/모텔 기능, 과금, 편리한 기능 등을 추가한 것이었다. 먼저 외국계 은행, 병원, 호텔 등에서 사용하기 시작하여 점차 범위가 넓어져 가는 경향이었다. 매 사용자마다 버전이 다를 정도로 사용자의 요구가 다양하였는데, 소프트웨어를 끊임없이 개발하고 보완하고 수정하였다. 일이 계속적으로 늘어나 사람이 많이 필요해져서 필자가 스카우트하기도 하고 그룹의 공개 채용 인원을 받기도 하여 인력을 확보하였다. 김경래씨(현 큐닉스컴퓨터 사장), 양승민씨(현 숭실대학교 교수), 이성규씨(현 삼성전자 이사), 정용석씨(현 코리아마이크로웨이브 이사), 이장식씨(미국 거주) 등이 스카우트된 사람들이고, 홍순호씨(현 삼성전자 이사), 이흥식씨 등이 공개 채용된 사람들이다.

KIT-CCSS의 개발에서 GTK-500의 상품화에 이르기까지를 정리해 보면, 그 당시 우리나라의 전자교환기 개발 능력을 점검할 수 있다. 전자교환기 개발에 필요한 많은 기술이 확보되어 국내의 역량을 모으면 국설전자교환기도 무난히 개발할 수 있다고 우리는 믿고 있었다. 물론 사설전자교환기 개발과 국설전자교환기 개발은 차원이 다르다고 말할 수 있다. 우리는 국설전자교환기 개발에 따른 문제점을 분석하고 있었는데, 시스템 기술, 대용량화 기술, 고신뢰성 기술, 체계적인 소프트웨어 개발 기술 등이 관건이라고 판단되었다. 그러한 문제점들은 충분히 해결할 수 있으며, 무엇보다도 전자교환기 개발에서 가장 중요한 부분인 소프트웨어 분야의 문제 해결이 가능하다고 생각하고 있었다. 결국은 필요한 연구비의 마련이 관건이었다.

국설전자교환기 개발의 꿈을 품고

1976년 2월에 국설전자교환기를 국내 개발하며 그 사이에 과도기적으로 선진국의 전자교환기를 도입하여 사용한다는 원칙을 정부 차원에서 결정하였다. 이와 관련하여 1977년 12월에는 한국통신기술연구소(한국전자통신연구원의 전신)가 설립되기도 했다. 그러나 시간적으로 급한 것이 외국 전자교환기의 도입이어서, 국내 개발은 안병성 박사, 박항구씨 등의 노력에도 불구하고 부진한 상태에 있었다.

KIST-500의 상품화가 이루어져 임무를 마친 필자는 국설전자교환기 개발의 꿈을 품고 1979년 3월에 다시 한국통신기술연구소의 전자교환기 개발 팀에 합류하였다. 이 때 동양정밀공업(주)에 가서 사설전자교환기 개발을 주도하고 있던 여재흥씨도 같이 합류하였다.

와서 보니 전자교환기 개발보다 훨씬 더 급한 프로젝트가 있었다. '114 번호 안내 전산 시스템' 개발 프로젝트였다. 대통령의 연두 순시 때에 연말까지 개통하여 서비스를 시작하는 것으로 보고했으니, 만사를 제쳐놓고 그 프로젝트부터 수행하라는 것이었다. 그렇게 해서 말은 114 번호 안내 전산 시스템 개발 프로젝트 때문에 정작 전자교환기 개발에는 제대로 참여할 수 없었다. 그 시스템은 1980년 5월에 서울 을지전화국에서 있는 부분 개통을 시작으로 약 10년 간 전국적으로 사용되었다.

열악한 여건에서도 매년 시험교환기 개발

그러나 그 프로젝트 때문에 전자교환기 개발, 특히 소프트웨어 부분은 적지 않은 타격을 입었다. 그 프로젝트가 예산, 인력 등을 포함하여 자원 배정에 최우선이었기 때문에, 한국통신기술연구소의 초창기 여건에서 전자교환기 개발의 자원 배정에 많은 제약을 가져왔다. 특히 소프트웨어 경우에는 더욱 자원 배정의 제약이 심해, 필자가 시스템 구조 설계에 참가하고, 교환기 소프트웨어에 경험이 없는 연구원이 한두 명 참가하는 정도로 여건이 열악했다.

그러나 이같이 어려운 여건 속에서도 매년 시험 교환기를 개발하고 제작하였다. 1979년에 개발하여 1980년 초에 제작한 것이 96 회선 용량의 1차 시험기였는데, 스위치 부분 등 극히 일부만 국내에서 개발하고 컴퓨터 등 대부분을 외국에서 사다가 만들었다. 200 회선 용량의 2차 시험기는 1980년에 개발하여 1981년 초에 제작하였다. 3차 시험기는 1981년에 개발하여 1982년에 제작하였는데, 1982년 7월부터 1983년 12월까지 용인의 송전우체국에서 시험 운용을 하였다. 나중에 국내에서 개발한 최초의 전자교환기를 TDX-1이라고 명명할 때, 이 3차 시험기를 TDX-1X라 명명하였다.

5공 출범으로 전자교환기 개발 후퇴

필자는 제5공화국의 출범이 결과적으로 전자교환기 개발, 특히 소프트웨어 부분의 시계를 한참 거꾸로 돌려놓았다고 생각한다. 전자교환기 개발에 상당한 수준의 인식을 가졌던 정만영 소장과 안병성 부소장이 연구소를 떠나게 되었고, 필자도 연구소의 대대적인 조직 개편에 따라 114 번호 안내 전산 시스템 프로젝트를 전담한다는 취지에서 전자교환기 개발 팀에서 완전히 빠지게 되었다. 대신 주로 하드웨어 경력을 가진 강진구(姜鎭求)씨가 삼성을 떠나 전자교환기 개발 팀에 다시 합류하여 소프트웨어를 맡았는데, 무척 힘들어했던 것으로 안다. 필자도 1982년에 앞의 프로젝트가 거의 마무리되고 TDX 개발 프로젝트가 급하게 되어 복귀하긴 했으나, 이전의 경험이나 지식을 충분히 발휘할 수 있는 여건이 아니었다.

한참 뒤에 TDX 사업단이 설치되었고, TDX 사업단의 개입으로 필자가 소프트웨어 개발의 책임을 맡아 소프트웨어의 체계적인 정리를 하게 되었다. 결과적으로 TDX 개발 프로젝트는 성공하게 되었으나, 그러한 과정에서 파생한 당시의 후유증으로 필자는 지금까지도 어려움을 겪고 있다.

TDX 개발 재개의 주인공 오명박사

필자는 한참 거꾸로 간 시계를 다시 돌려 놓은 분이 오명박사(현 동아일보 사장)라고 생각한다. 오박사가 어려운 여건과 주위의 많은 반대를 무릅쓰고 240억원의 TDX-1 개발 프로젝트를 성립시키고 추진한 것은 잘 알려진 사실이다. 하지만 이것보다 결코 못하지 않은 성과가 TDX 사업단의 설치라고 필자는 생각한다.

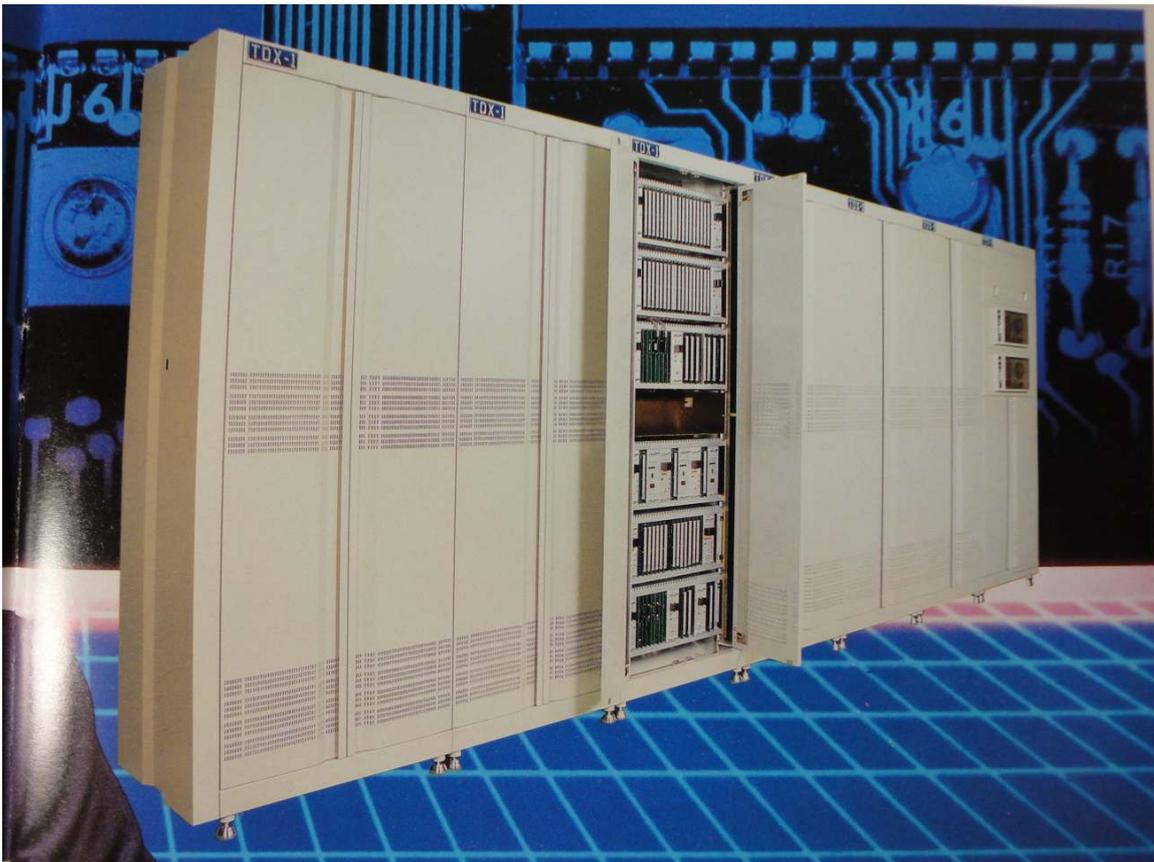
전자교환기를 도입하려던 당시 상황에서 주위의 반대는 많았다. 그러나 설사 TDX-1 개발 프로젝트가 실패하더라도 국가적으로 손해 볼 것은 없었다는 것이 필자의 판단이다. 외산 전자교환기 도입 계약 과정에서 국내 개발을 협상카드로 활용하여 계약금을 줄인다면 240억원 정도의 돈은 충분히 차익으로 얻어낼 수 있는 액수였기 때문이다. 실제 TDX 개발 덕분에 전자교환기 도입 계약에서 국가적으로 천문학적 액수의 이익을 본 것으로 필자는 알고 있다. 또한 TDX-1 개발에 들어가는 240억원은 상당 부분

이 국내에서 쓰여질 것이며, 그것에 의해 사람이 양성되고 기술이 발전되며 산업이 발전할 수 있는 것이다.

물론 이 정도의 판단은 고위 정책결정자로서 기본일 수 있다. 그러나 TDX 사업단의 설치는 전문가적인 식견이 없으면 불가능한 일로, 그 당시 오박사 외의 다른 사람은 아무도 이를 생각지 못하고 있었다. 그 시점에 TDX 사업단을 설치하지 않았더라면 TDX-1 개발 프로젝트가 표류했거나, 적어도 지금과 같이 성공한 프로젝트라는 찬사는 듣지 못했다고 본다. TDX 사업단과 관련된 이야기는 다음 호에 계속 하겠다.

TDX 1천만 회선 돌파 기념 숨겨진 TDX 개발 이야기(3) -TDX-1의 개발

제3부 TDX-1의 개발



선무당과 잡지파의 시대

여기서 잡지파란 소프트웨어에 대한 정규 교육이나 경력 관리에 의한 체계적인 능력 개발이 없이, 잡지를 읽고 손쉽게 얻은 지식을 바탕으로 소프트웨어 개발 프로젝트에 참여한 사람들을 일컫는 속칭이다.

복잡하고 대형인 소프트웨어 개발 프로젝트의 비극은 이러한 잡지파들과 선무당들이 소프트웨어 구조나 소프트웨어 개발 방법론에 관한 의사 결정에 관여할 때 발생한다. 이것은 소프트웨어 개발에 필요한 의사 결정을 제때에 하지 않는 것도 포함한다. 문제의 근원은 소프트웨어를 실제로는 잘 모르면서 자기도 잘 안다고 착각하는 데 있다. 모르는 것을 인정하면 대책이 있으나, 모르는 것을 안다고 착각하면 대책이 없는 것이다.

소프트웨어를 잘 아는지에 대한 간단한 진단법은 소프트웨어 개발에서 소프트웨어의 형상 관리가 무엇이고, 왜 필요하며, 어떻게 하는가를 질문해 보면 된다. 이 질문에 대한 대답이 '모른다' 이면서 소프트웨어 개발 프로젝트의 책임을 맡은 사람은 관리 방법을 바꾸어야 그 프로젝트가 성공할 수 있다. 이

경우 직접 관리 방식은 백전백패이니, 반드시 앞의 질문에 ‘안다’ 인 사람에게 위임하여 관리하여야 한다.

인력배치 혼란으로 개발 체계 와해

1982년부터 본격적으로 TDX-1 개발이 시작되었다. TDX-1X에 대한 시험 운용과 보완, TDX-1의 개발 등이 숨 가쁘게 진행되고 있었다. 갑자기 엄청난 규모의 프로젝트를 수행하게 되니 하루가 다르게 사람이 늘어나기 시작하였는데 제대로 교육이나 훈련을 하고 배치할 형편이 아니었다. 또한 1983년 2월에 연구소(한국전기통신연구소)가 대덕으로 이전하였는데, 핵심 개발 요원 중에서 여재흥씨와 강진구씨가 대덕으로 이전할 형편이 못되어서 프로젝트에서 빠지게 되었다. 필자도 앞에서 언급한 바와 같이 곁돌고 있는 상태여서 경험이 있는 핵심 요원은 박항구씨 혼자뿐인 것과 다름이 없었다. 결과적으로 개발 조직이 극도로 혼란스러울 뿐만 아니라 십여년 간 쌓아 온 전자교환기 개발의 체계가 와해되어 버렸다. 할 일은 많아 갈 길은 바쁜데, 사람은 많아 제 각기 자기의 영역과 주장을 내세우니 백가쟁명이라 제대로 결정되는 것이 없었다.

끝없는 회의가 연일 계속되었다. 잘 믿어지자 알겠지만 그 당시 많은 의사 결정이 다수결의 원칙에 따라 결정이 되었다. 그 유명한 TDX-1 개발의 중요한 의사 결정이 다수결로 이루어지다니, 프로젝트의 전체 목표에 따라 분야별로 정해진 임무와 시간 계획이 있기는 있었다. 그러나 의사 결정이 제때에 안되고, 되어도 승복하기 어려운 결정이 많으니 분담한 분야에서는 각자가 알아서 적당히 수행하는 경우가 많았다. 어차피 시스템이 체계적으로 분석되고 분할되어 있지 않아, 분야 간 연결이나 인터페이스는 나중에 문제가 생기면 우기거나 책임을 회피하면 되기 때문이다.

계속되는 허위 보고

이 경우 가장 심각하게 타격을 받는 분야가 소프트웨어이다. 시스템 차원의 결정이 안 되면 소프트웨어 자체의 분석이나 분할 작업을 할 수 없거나, 무리를 해서 하여도 나중에 결정이 되면 거의 기존의 작업 결과를 쓸 수 없게 되기 때문이다. 필자는 두 번의 전자교환기 개발로 문제들이나 그 문제들의 해결 방법을 알고 있었으나, 다수결 위주의 상황에서는 제대로 의사 결정이 되게 할 방법이 없었으므로 답답하기 이루 말할 수 없었다. 모르는 것이 약이라는 말처럼 차라리 모르거나 그것도 아니면 회피라도 할 수 있으면 좋겠는데, 계속되는 회의에 참석을 해야 하니 회피도 할 수 없어 매일매일이 고통의 연속이었다. 이것은 겪어 보지 않고는 알 수 없는 고통으로 필자가 지난 약 25년 간의 연구원 생활에서 이 때만큼 고통스러운 때는 없었다.

이렇게 프로젝트의 진행 상태가 나쁘게 되는 것은 프로젝트에 대한 진도 파악과 관련이 많다. 이런 상황에서 하는 진도 파악은 누구도 문제가 있다고 말하지 않으니 진도 파악 자체만으로는 프로젝트가 순조롭게 진행되고 있는 것으로 나타난다. 문제가 있다고 말하여도 의사 결정이 제대로 되지 않으니 문제가 풀리는 것도 아니며, 오히려 일을 제대로 못한다고 욕을 먹고 또 나중에 시스템 전체에 관련되는 문제가 발생할 때에 책임을 뒤집어 쓸 가능성이 커지기 때문이다.

결과적으로 거의 모두가 허위 보고를 하고 있었다. 스스로는 알면서 허위 보고를 하는 사람도 있었고, 분야별 진도 파악이 안 되어 모르고 허위 보고를 하는 경우도 많았다고 본다. 결론적으로 프로젝트의 내용을 잘 모르는 사람에게 이와 같은 상황에서 하는 진도 보고는 의례적으로 순조롭게 프로젝트가 잘 진행되고 있다는 것이 된다. 그러나 이와 같이 진행되는 프로젝트는 그 프로젝트가 끝나야 할 때쯤, 다시 말하면 전자교환기의 제작이 완료되어야 할 시점에서 그때까지의 허위 진도 보고가 끊어 터지면서 문제들이 한꺼번에 노출되는데, 그 때는 백약 무효이다. TDX-1 개발도 당시 이것과 유사한 조짐을 보이면서 표류하고 있었다.

TDX 사업단의 설치

1984년 1월 한국전기통신공사(한국통신)에 전자교환기 사업단(TDX 사업단)이 설치되고 서정욱 박사가 단장으로 부임하였다. TDX 사업단의 설치 목적은 TDX-1의 상용화를 위해서 TDX-1 개발 업무를 종합 관리하는 것이었다. TDX 사업단은 필요할 경우 TDX 개발팀(TDX 개발단)을 직접 지휘할 수 있게 되어

있었다. 이것은 외적인 당연한 목적 이외에 표류하고 있는 TDX-1 개발을 정상화하려는 내면의 목적도 있었기 때문이다. 여기서 TDX 사업단의 설치에 관련된 내면의 이야기를 이 정도로 언급하고 끝내는 것이 좋겠다. 이 사건으로 TDX-1 개발은 일대 전기를 맞이하게 된다. 그 때까지 가내공업식으로 수행되어 오던 TDX-1 개발이 조직적으로 추진되는 계기가 된 것이다.

1984년 초 어느 날. TDX 사업단에서 필자에게 올라오라는 전갈이 왔다. 주위의 이야기도 있고 해서 부르는 이유를 짐작할 수 있었으나, 그 동안 필자는 유사한 일로 여러 번 실망한 적이 있었으므로 시큰둥한 입장이었다. 전자교환기를 개발하는 과정에서 윗분들이 말로는 소프트웨어의 중요성을 이야기하면서 막상 의사 결정을 할 때는 엉뚱하게 하는 것을 여러 번 보아 왔기 때문이다. 혹시나 싶어 소프트웨어에 대한 TDX 사업단장의 인식을 확인하려고, F. P. Brooks의 『the mythical man-month』라는 책을 가지고 갔다. 그 날 TDX 사업단장과 한 시간 정도 이야기를 한 것으로 기억하는데, 중요한 내용을 간추리면 다음과 같다.

“현재 TDX-1 개발, 특히 그 중에서 소프트웨어 쪽에 문제가 많다는데, 정말인가?”

“그렇다.”

“TDX-1X를 어떻게 생각하는가?”

“원숭이 새끼라고 생각한다.”

“앞으로 어떻게 하는 것이 좋겠는가?”

“지금까지 결과에 연연하지 말 것이며, 특히 소프트웨어는 체계를 혁명적으로 정리해야 한다.”

원숭이에 관련된 내용을 좀더 설명하겠다. 원숭이 새끼는 아무리 잘 키워도 원숭이 밖에 안 되니, 어린 애를 낳아서 잘 키워야 훌륭한 사람으로 키울 수 있다는 뜻이다. TDX-1 개발이 성공하려면 새로운 사고가 필요하다는 의미였다. 이러한 이야기를 마치고 느낌이 와서 소프트웨어 프로젝트를 잘 관리하려면 소프트웨어 공학에 대한 이해가 필요하며 그것에 도움이 될 책이라면서 가져간 책을 드리고 왔다.

삼사일 후에 연구소에 오서서는 책은 잘 보았고 전에 국방과학연구소에서 한 연구원이 준 원본이 있다면 책을 돌려주었다. 비록 복사본이지만 필자가 애지중지하던 책이라 무척 반가워하면서 책을 살펴보았다. 필자가 읽을 때 줄친 부분 이외에 추가로 표시한 부분도 있었는데, 책의 내용을 이해한 흔적이 뚜렷하여 소프트웨어에 대한 인식을 확립할 수 있었다. 그 때, 필자는 기회가 주어지면 어떤 어려움을 무릅쓰고도 소프트웨어 개발을 책임지고 잘하겠다고 결심하였다. 그 이후에 TDX 개발단 선임실장 제도가 임시로 생겼으며, 필자는 S/W 담당 선임실장을 맡아 소프트웨어 개발을 지휘할 수 있게 되었다. 1985년 3월에는 소프트웨어 개발부가 생기면서 그 부장을 맡게 되어 소프트웨어 분야를 보다 용이하게 지휘할 수 있었다.

개발 외적인 저항 엄청난

TDX-1 개발에서 소프트웨어의 체계를 정리하는 일은 매우 힘든 일이었다. 이미 개발 업무가 진행된 상태에서 해야 했기 때문에 그것에 따른 걸림돌이 많았다. 현장 시험을 시작한 상태였으므로 비록 무효이기는 했으나 가입자들의 반응도 고려해야 하고, 현장에서 발생하는 문제점도 보완해야 하는 등 개발 외적인 고려 사항이 많아 노력이 여러 갈래로 분산될 수밖에 없었다. 또한 조직 내의 갈등으로 인하여 소프트웨어의 체계 정리에 대한 저항이 엄청났기 때문에, 설사 논리적인 문제가 있어도 가능한 한 지금까지의 내용도 활용해야 했으므로 이중 삼중의 부담이 되었다. 소프트웨어의 체계 정리 작업은 것처럼 어렵게 또 조심스럽게 진행되어, 약 1년 반 후인 1985년 가을에 첫 번째 버전이 현장에 설치되었다. 소위 말하는 ‘소프트웨어 혁명’의 결과인데, 이것에 대한 좀더 자세한 이야기는 다음에 계속하겠다.

TDX 사업단의 성과에 대해서는 의견이 분분하다. 인중 제도를 도입해서 TDX-1을 쓰게 만듦으로써 TDX-1의 상용화를 앞당기거나 상용화에 결정적인 역할을 하였다는 것에는 거의 의견이 일치하는 것 같다. 필자는 표류하던 TDX-1 개발의 정상화에 대한 역할을 이것보다 더 평가하고 싶다. 상용화가 개발이 성공한 이후의 이야기지, 개발이 실패했다면 상용화가 무슨 의미가 있는가. 그 당시 TDX-1 개발 프로젝트의 성패는 소프트웨어 개발에 달려 있었고, 소프트웨어는 그 때 수령에 빠져 들고 있었다.

TDX-1 개발 프로젝트에서 부딪힌 소프트웨어 위기를 그런 방법을 통해서라도 해결하지 않았다면, 오늘날과 같은 TDX는 아마 존재하지 않을 것이다.

개발 방법론이 위기에

TDX-1 개발 과정에서 개발 방법론과 관련하여 사연이 많다. 이와 같이 복잡하고 대형인 프로젝트에서 개발 방법론을 사용하지 않을 수 없다는 전제하에 개발 방법론 측면만 본다면 TDX-1 개발은 영점에 가깝다. 정상적인 발전 단계에 의한다면 TDX-1을 개발할 때에는 소프트웨어의 개발 방법론이 정립되어 품질 향상과 생산성 향상을 지향하는 제3단계에 진입하여야 마땅하였다. 그러나 수행 주체가 바뀌는 바람에 지난 십여년 간 축적된 제1단계와 제2단계가 없어져버린 결과가 된 것이다.

TDX-1 개발 초기에 전격적으로 시도된 SADT의 적용이 문제된 바 있었으며, 상황을 아주 나쁘게 만든 것은 AXE work methods를 TDX-1 개발 방법론으로 정해 버린 것이었다. AXE 10 전자교환기의 도입에 따라 함께 들어오게 된 것을 선진국 교환기의 개발 방법론이니 활용한다는 것이 화근이었다. AXE work methods를 TDX-1 개발에 적용했을 때에 여러 가지 문제가 많았지만, 소프트웨어 측면에서 심각한 문제를 야기시킨 두 가지만 예를 들면 다음과 같다.

첫 번째는 TDX-1은 분산 제어 구조인 반면, AXE 10은 중앙 제어 구조이기 때문에 발생하는 문제였다. AXE 10에서 같은 모듈로 된 프로그램이라도 TDX-1에서는 여러 개의 프로세서에 분산되는 것이었다. 소프트웨어 개발 방법론의 골격이 프로그램과 문서의 체계적인 연결인데, 프로그램과 문서가 연결이 되지 않아 힘들여 만든 문서가 무용지물이 되는 것이었다. 물리적으로 흩어져 있는 프로그램을 억지로 한 모듈로 하거나 아니면 프로그램 체계와 문서 체계를 물리적으로 달리 할 수밖에 없었다.

두 번째는 사용하는 프로그램 언어의 차이였다. AXE 10은 PLEX라는 고급 언어를 쓰고, TDX-1은 Zilog 어셈블러를 썼다. 고급 언어와 어셈블러의 사람에 대한 친숙도는 5~10배의 차이가 있어, PLEX를 기준으로 작성된 문서의 자세함으로는 TDX-1의 프로그램을 이해할 수 없었다. 다시 말하면 어셈블러로 프로그래밍되는 소프트웨어의 설계 문서는 고급 언어의 그것보다 훨씬 더 자세하게 설계 문서를 작성해야 하는 것이다.

원론 무시한 정책적 결정과 충동도

이런 원론을 무시하고 정책적으로 결정된 TDX-1 개발 방법론은 TDX-1 개발 과정에서 끊임없이 우리, 특히 소프트웨어 팀을 괴롭혔다. 소프트웨어 개발 방법론의 전문가가 아니면 소프트웨어 전문가도 이해하기 쉽지 않은 이런 문제점들을 어떻게 하드웨어 측 사람들이 이해할 수 있겠는가. 필자가 소프트웨어 책임을 맡아 소프트웨어의 체계를 정리하는 과정에서는 더욱 심했다. 책임자급에서 하도 심하게 논쟁을 하고 옆치락뒤치락 하고 있으니, 그럴지 않아도 문서화나 체계화를 싫어하는 연구원들이 그때까지의 적용 결과에 불신감을 표시하면서 아예 개발 방법론의 적용 자체를 거부하는 지경에 이르기까지 하였다. 이럴수록 답답한 쪽은 소프트웨어이다. 어느 분야가 개발 방법론에 더 민감하며 왜 이 같은 상황이 발생하였는가를 책임자들이 제대로 파악하여 대처하지 못한 데서 온 결과였다.

어렵게 수행된 소프트웨어의 체계 정리 작업의 결과는 우여 곡절 끝에 1985년 가을에 첫 번째 버전을 현장 시험 중이던 서대전 전화국과 유성 분국에 설치하였다. 마침 필자는 그 때에 사전 계획에 따라 2주간의 해외 출장을 가게 되었다. 출장 일주일쯤에 부장 직무대행으로부터 연락이 왔는데, 징계위원회에 회부되어 면직되게 생겼으니 빨리 귀국했으면 좋겠다는 것이었다. 그 당시 실험실에서 시스템 시험을 많이 한 후에 현장에 설치하였으나, 소프트웨어 체계를 혁명적으로 바꾸었기 때문에 현장 설치 초기에 소수의 오류가 소프트웨어에 발생한 것이 문제가 된 것이었다. 필자는 자리에 연연하고 싶지도 않았고 또 바로 안정될 수 있을 것이라는 확신도 있었기 때문에 당초 계획대로 출장을 마치고 귀국하였다. 그 때, 소프트웨어는 이미 안정되어 있었고, 그 동안에 있었던 소프트웨어의 체계 정리 작업이 합법적으로 진행되었음을 나타내는 결재 문서들을 소장실에 제출하는 것으로 일단락되었다.

그 당시 TDX 개발단의 분위기는 이 정도였으며 위기에 처한 필자를 내부에서는 아무도 변호해 주지 않았다. 소식을 들은 TDX 사업단장께서 “내가 지시한 것이다”라고 돌려대어 필자가 위기의 순간을 면하였고, 사안 자체가 원래 전혀 문제의 소지가 없는 것이었으므로 해프닝으로 끝나게 되었다.

어렵게 정리된 개발방법론

이렇게 TDx-1 소프트웨어 체계 정리는 어렵게 이루어졌고, 그것에 힘입어 TDx-1 개발 프로젝트는 성공적으로 끝났다. 성공한 프로젝트이기 때문에 주위로부터 찬사를 듣고 있는 지금은 서로가 자기의 공이 더 많다고 주장하고 있는 것이 현실이다. 그 때의 일을 생각하면 참으로 낯 뜨거운 일이다. TDx-1 개발시에 정말 애착을 갖고 프로젝트를 수행했던 사람들이 누구였는가를 한 번쯤 생각해 보기 바라며, 상처뿐인 영광과 함께 조용히 지켜 보고만 있는 사람들이 있다는 것을 생각하면서 자기의 공을 주장하기 바란다.

이렇게 정립된 TDx-1의 개발 방법론은 그 이후에 연속된 TDx 개발에도 적용되었으며, 또 다른 국책 프로젝트인 국가기간전산망용 주전산기 개발 프로젝트에도 활용되어 우리나라의 컴퓨터 시스템 개발 프로젝트에 대한 개발 방법론의 초석이 되었다.

“ 현재 TDx-1 개발, 특히 그 중에서 소프트웨어 쪽에 문제가 많다는데, 정말인가?”

“ 그렇다.”

“ TDx-1X를 어떻게 생각하는가?”

“ 원숭이 새끼라고 생각한다.”

“ 앞으로 어떻게 하는 것이 좋겠는가?”

“ 지금까지 결과에 연연하지 말 것이며, 특히 소프트웨어는 체계를 혁명적으로 정리해야 한다.”

TDx-1A/TDx-1B 개발

개발된 TDx-1을 금성, 동양, 대우 및 삼성의 네 회사에서 생산하기 시작하였으며, 1986년 3월 국산 전자교환기의 역사적인 개통이 가평, 고령, 우주 및 전곡의 네 곳에서 있었다. 크게 볼 때는 TDx-1 하나이지만, 세분하면 TDx-1A와 TDx-1B가 더 있다. TDx-1은 최초로 개발되고 개통된 전자교환기이고, TDx-1A는 TDx-1의 개량된 양산 모델인 셈이고, TDx-1B는 더 개량된 양산 모델인 셈이다.

각각의 개발 방식도 좀 다른 면이 있었다. TDx-1A까지는 연구소가 완전히 개발을 주도하였으나, TDx-1B의 개발에는 네 회사도 주도적으로 참여하였다. TDx-1의 가입자회선 용량은 9,600 회선이었는데, 전화국의 통상적인 회선 용량 단위인 1만 회선이 안되는 것이 문제였다. TDx-1의 회선 용량을 1만 240회선으로 늘리고 일부를 보완한 것이 TDx-1A이다. TDx-1A의 개발은 전체적으로 TDx-1 개발 경험이 있었으므로 무난하게 개발할 수 있었다.

1988년 초에 한국통신에서 사업개발단장의 교체가 있었고, 연이어 한국전자통신연구소에 중용량교환기개발본부(TDx-1B 개발본부)를 설치하여 줄 것을 요청하였다. 그 때, TDx-1B 개발과 TDx-10 개발의 병행 추진 방식을 놓고 TDx개발단 내에서 의견이 분분하였다. TDx-1B 개발은 TDx-1A의 용량을 두 배로 늘리는 작업이고, TDx-10 개발은 대용량의 전자교환기 개발로서 당시 시스템 정의 단계에 있었다. 그러므로 필자는 TDx-10의 시스템 정의 작업에는 많은 사람이 필요하지 않으며, TDx-1B 개발에 인력을 집중적으로 투입하여 빠른 기간에 TDx-1B 개발을 끝낸 후에 TDx-10 개발에 다시 투입하는 의견이었다. 우여곡절 끝에 필자의 의견이 채택되어 1988년 3월 TDx-1B 개발을 위한 대규모의 TDx-1B 개발본부가 발족되었다. 9개실에 TDx 개발단 인력의 약 50% 정도를 수용하는 규모였으며 필자가 본부장을 맡았다.

TDx-1B 개발본부의 발족으로 1986년 하반기부터 산발적으로 진행되어 오던 TDx-1B 개발이 급속도로 진행되었다. 약 3개월이 지나 가시적인 성과가 나오고 있을 즈음에 한국통신에서 사업개발단장의 교체가 다시 있었고, 연이어 TDx-1B 개발본부의 해체를 요구하였다. 결국 TDx 개발조직은 3개월 전에 조직 개편을 하기 이전으로 되돌아가고, TDx-1B 개발본부는 껍데기만 남게 되었다. 그 이후

TDX-1B 개발의 추진 주체가 한국통신의 사업개발단으로 넘어갔고, 필자도 TDX 개발 분야를 떠나게 되었다.

S/W 개발 체계 정립으로 후속 개발 가능

마지막으로 TDX-1A 개발과 TDX-1B 개발에 대해 소프트웨어 측면에서 지금까지 나온 견해들과 사뭇 다른 필자의 견해를 밝히고자 한다. TDX-1A 개발을 무난하게 할 수 있었던 것과 TDX-1B 개발의 추진 주체를 이전하고도 TDX-1B 개발이 성공한 것에 대해, 소프트웨어 개발의 체계가 이미 정립되어 있었기 때문에 가능했다는 사실을 눈여겨 보는 사람들은 극히 소수이다. TDX-1 개발시에 정립된 소프트웨어 개발 방법론은 그 정도의 변경은 쉽게 수용할 수 있게 되어 있었다고 본다. TDX-1A 개발과 TDX-1B 개발에 대한 TDX-1 개발 방법론의 기여를 과소 평가한 결과는 현재 당면하고 있는 국산 전자교환기와 관련된 문제점들의 근원을 제공하고 있는데, 이것을 인식하는 사람도 많지 않다.

가장 문제가 되는 부분이 TDX-1B를 개발할 때부터 시작된 소프트웨어에 대한 형상 관리의 부재이다. TDX-1B 개발은 나름대로 성공하여 대량으로 공급됨으로써, 형상 관리를 정립하는 개발 주체의 존재와 그것의 역할을 그 때부터 과소평가하기 시작한 것이다. 형상 관리의 기본은 모든 종류의 소프트웨어 개발 시에 개발 주체가 있어야 하며, 공동 개발에 참여하는 나머지는 그 개발 주체의 형상 관리 방식을 따라야 하는 것이다. TDX-1B를 개발할 때부터 흐트러지기 시작한 이 개념은 TDX-10 개발, TDX-100 개발 등에 영향을 미쳤다.

결과적으로 형상 관리의 주체, 다시 말하면 전자교환기 소프트웨어의 주인이 없어지게 되어 현재 여러 가지 문제점을 야기시키고 있는 것이다. 이런 현상의 시발점이 TDX-1B 개발본부를 해체하고 TDX-1B 개발의 추진 주체를 바꾼 때라고 생각하며, TDX-1B 개발의 성공적인 수행에도 불구하고 필자가 아쉽게 생각하는 점이다.

에필로그

소프트웨어 책임자의 행운과 불운

전자교환기를 개발하는 동안 필자가 소프트웨어 분야에서 본의는 아니지만 결과적으로 리더의 역할을 한 것은 운명적이라고 생각한다. 또한 아무리 기다려도 다른 사람이 쓰지 않아, 이와 같은 글을 할 수 없이 지금 필자가 쓰고 있는 것도 마찬가지일 것이다. 이것이 필자에게는 한편으로 행운이기도 하면서 또 한편으로 불운이기도 하였다. 필자와 비슷한 연령으로 다른 분야의 소프트웨어에 종사하고 있는 상당수의 분들이 이와 같은 경험을 하였을 것이라 생각한다.

행운의 대표적인 경우는 거의 처음부터 끝까지 책임자의 역할을 맡아 소프트웨어 개발을 지휘하면서 나름대로 뜻을 펼 수 있었으며 상응하는 대우도 받았다는 것이다. 불운이 더 많았는지 모르겠는데, 항상 하드웨어 위주의 경험을 가졌거나 진정으로 소프트웨어를 이해하지 못하는 상사를 모셔야 했고, 또한 그와 같은 동료에 둘러싸인 상태에서 개발 업무를 수행해야 했다는 것이다. 이러한 환경에서 필자는 항상 외로운 투쟁을 하여야 했고 때로는 독불 장군으로 미움을 받는 경우가 많았다는 것이다.

잘 알다시피, 전자교환기 같은 것을 개발하는 프로젝트의 수행은 시스템 전체의 기능과 성능을 하드웨어와 소프트웨어에 할당하는 것과 인력을 포함한 자원의 배분 등에 관한 의사 결정의 연속이다. 이 과정에서 설사 평소에 친한 친구라도 하드웨어 측의 사람이 소프트웨어 측의 편을 들어주는 경우는 절대 없다. 그러므로 합리적인 할당과 배분은 하드웨어와 소프트웨어를 총괄하는 상사나 프로젝트 책임자에 달려 있으며, 프로젝트의 성패가 이것에 달려 있다고 해도 과언이 아니다. 이 때, 소프트웨어를 진정으로 이해하지 못하는 상사가 눈에 보이지 않아 무엇인지도 잘 모르겠고, 그래서 더 머리를 아프게 만드는 소프트웨어를 좋게 보고 소프트웨어 측에 불리하지 않는 의사 결정을 할 수 있겠는가. 결과적으로 시스템 전체의 균형이 맞도록 제대로 의사 결정을 할 수 있겠는가.

소프트웨어 책임자의 불운은 여기서 끝나지 않았다. 넓게는 개발 방법론, 좁게는 개발 방식의 적용과 관련된 것이다. 전자교환기 개발과 같은 대형이며 복잡한 소프트웨어 개발은 체계적인 개발 방식의 적

용이 필수적이다. 이것은 본격적인 소프트웨어의 개발이 시작되기 전에 적절한 개발 방식의 채택이 필요하며, 또한 그 방식은 하향식 개발(top-down development)을 의미하는데 달리 표현하면 시스템 정의를 하고 다음에 설계를 하며 이것을 바탕으로 프로그램을 만든다는 의미이다. 하드웨어의 경우도 이와 같이 해서 나뉘어야 없지만, 하드웨어의 특성상 소프트웨어처럼 체계적인 개발 방식의 적용과 하향식 개발의 적용에 목을 맬 이유는 없다.

또한 소프트웨어는 개발 방식의 내용에 하드웨어보다 훨씬 민감하여 개발 방식이 바뀌면 그것에 따라 전체를 다시 개발하여야 한다. 그러므로 소프트웨어 책임자가 제대로 소임을 다하려면, 개발 방식의 채택과 초기 정착에 보다 도전적이어야 하고 개발 방식의 내용에 보다 민감하여야 하며 개발 방식의 변경에 보다 보수적이어야 한다. 이것 또한 소프트웨어 책임자가 상사와 주위 동료에 대응해 외로운 투쟁을 하게 만드는 운명적인 굴레인 것이다.

아직도 TDX-1이라 부르는 전자교환기를 개발할 때에 있었던 이야기 중에서 못다 한 부분이 적지 않다. 이 부분도 정확한 전자교환기 개발의 역사를 정립하기 위해서는 더 밝혀져야 한다고 생각한다. 특히 TDX-1 개발의 일등 공신이 누구인가를 두고 자천 타천의 인사들이 많다. 이것을 푸는 간단한 열쇠는 TDX 사업단의 설치 배경에서 찾을 수 있다고 생각한다. 왜 그 시기에 그런 방법으로 TDX 사업단을 설치하게 되었는가에 대해 아직 알려지지 않은 이야기가 있다. 그 당시 TDX 사업단의 설치 결정에 직접적으로 참여한 두 분 정도는 그 내막을 확실히 알고 있으며, 그 외 직간접으로 관련된 세 분 정도는 어느 정도 그 내막을 알고 있는 것으로 판단된다. 물론 필자도 그 내막을 알고 있다. 그러나 아직은 때가 이르다고 판단되며, 훗날 모든 것을 다 밝힐 수 있는 기회가 오기를 기대해 마지않는다.