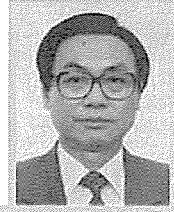


海外 소프트웨어의

受注開發



(주)한국데이터뱅크/고문 김증모

1 1966년에 우리나라 初期 컴퓨터導入을 위하여 日本에서 프로그래밍教育을 받은 因緣으로 80年代 初盤에는 日本關係會社에 勤務하게 되었고, 다시 그 後半부터는 日本의 소프트웨어를 受注開發하는 業務에 從事하게 되었다. 그동안 經驗했던 일중에서 몇 場面을 紹介하려 한다.

近來에 와서 首都圈周邊에 住居를 둔 外國인들이 자주 눈에 띈다. 차림으로 보아 事務職 從事者는 아닌 것 같다. 狀況은 틀릴지 모르나 目的은 비슷한 經驗이 우리에게도 있었다.

日本の 景氣가 한참 好況일 때 情報處理技術者가 상당히 不足했던 時期가 있었다. 그 當時 도쿄周邊의 賃貸料가 比較的 低廉한 아파트를 빌려 많은 우리 프로그래머들이 여러명씩 常駐하면서, 研修形態로 比較的 싼 派遣費를 받고 그곳 소프트웨어開發에 參與한 일이 있었다. 그런데 90年代初 不法就業者數가 늘어나고, 景氣도 下降局面에 들어서면서 비자制度가 嚴格해져 一定條件의 就業이 아니면 常駐할 수 없게 되었다. 그래서 새로운 方法의 摸索이 必

要하게 되었다.

바람직한 方法은 아니지만 이런 派遣作業形態에서 얻은 所得도 적지 않았다. 優先 言語疏通이 圓滑하게 되었고 必須的인 文書化를 爲始한 作業基準과 慣習의 熟達, 保有 技術力の 認定, 나아가서는 人間關係의 維持 等を 基礎로 受注開發로 業務形態를 改善하는 基礎를 마련할 수도 있었다. 특히 要求된 機能뿐만 아니라 品質, 納期를 重視하는 소프트웨어開發에서는 어쩌면 必須的인 過程이었는지도 모른다.

勿論 이런 過程이 없었던 境遇도 있었다. 그 代身 費用을 들여가면서 試驗的으로 조그만 일을 맡겨서 그 結果를 評價해 보는 것이었다. 그런데 受注開發의 경우 國內에 開發環境이 갖추어지지 않으면, 文書化以外的 作業은 現地 作業으로 進行할 수밖에 方法이 없었다. 한편, 日本의 컴퓨터메이커들은 소프트웨어關係 子會社뿐만 아니라 많은 外注會社를 維持하고 있었다.

도쿄에서 北西쪽으로 1時間余의 距離에 오오메라는 조그만 都市에 T社의 컴퓨터 生産工場이 있었다. 이 工場에서는 소프트웨어部門의 한層의 절반은 自體組織에서 쓰고, 나머지 半에는 10余 外注 소프트웨어會社要員들의 事務室로 割當하고 있었다. 여기서 우리는 試驗팀으로 4名이 構成되어 한달間 12,000스텝規模의 프로그램 3個로 構成된 開發業務를 詳細設計에서 테스트工程까지를 作業하였고,

部課長級 4人으로부터 口頭試驗같은 開發管理面的 質疑應答式 審査를 받았으며, 外注業體로 評價받아 基本契約을 거쳐 個別契約으로 이어질 수 있었다. 當時의 소프트웨어 種類는 미니컴用 業種別 패키지였다. 87年 6月의 일이다.

그 當時에는 過渡期的인 作業形態도 있었다. 國內에서 文書化作業과 코딩을 마치고 프로그램은 테이프에 收錄한 다음, 現地에 出張가서 發注社의 테스트環境에서 테스트하여 納品하는 形態였다. 이 境遇 短期出張으로 長期滞在問題는 解決되었으나 테스트期間中 要員들은 밤낮이 따로 없었다.

90年代에 들어서서는 이와같은 課題가 劃期的으로 解決되었는데, 그 手段으로 登場한 것이 지금도 不可缺하게 活用되고 있는 國際通信回線이다. 卽 호스트 컴퓨터와 端末間 또는 서버와 클라이언트間을 專用回線으로 接續하여, 現地에 出張하여 使用하는 것과 다름없는 開發環境을 國內에서 利用할 수 있게 된 것이다. 이 通信回線을 利用한 그룹웨어를 통하여 必要한 資料의 送付와 業務協議 등도 相當部分 解決할 수 있게 되었다.

勿論 이런 事前 整地作業같은 業務外 課題만 있었던 것은 아니다. 業務推進面에서는 發注側에서 要求하는 標準化된 開發節次에 따라 成果物을 作成하는 것, 一定 品質水準을 維持하면서 주어진 納期를 遵守하는 것, 이런 與件을 充足하면서 水準以上の 生産性を 올리지 못하면 프로젝트

로서 持續될 수 없게된다. 이런 課題는 이미 適應經驗을 거친 現地會社에 비하여 劣勢일 수밖에 없고, 時間을 두고 解決해야하는 課題라 할 수 있다.

開發作業面에서의 例를 들면, 品質管理基準에서 單位테스트에서는 프로그램 命令 10스텝에 1件以上の 테스트케이스를 設定하여 테스트해야하고, 모든 命令은 적어도 한번 經由하도록 테스트데이터가 作成되어야 하는 것 등이 있으며, 프로그램의 構造는 標準패턴에 따라 構成되어야 하는 등이다. 이 標準패턴을 使用하면, 完成된 프로그램중 코딩해야하는 部分은 全體의 절반程度에 지나지 않으므로 상당한 生産性向上 效果가 發揮된다고 할 수 있다. 이런 CASE TOOL을 使用하여 開發할 수 있는가하는 것이 受注要件이 되기도 한다.

이와 같은 要件을 具備하고 安定되며 持續的인 受注를 確保하기 위해서는 이미 開發, 納品한 소프트웨어에 대한 補修對應과 蓄積된 開發經驗을 통한 該當시스템의 改善提案 等 營業的 對應도 必要하게 된다. 結局 競爭力의 確保는 어느 部門에서도 必要 不可缺한 要素가 아닌가 한다.