

日本企業には管理職以外のキャリアパスはほとんど無く、**優秀なエンジニアを育成する土台が無い**。長く続いた景気低迷の余波もあり、多くの企業で IT 部門は衰退し、エンジニアは評価も得られず疲弊している。企業は内製力を失い、IT 活用は進まず、過剰な IT ベンダー依存が非効率で高コストな状態を作り出している。私はまさにその真っ只中にいた。IT は企業にとって欠かせないものでありながら、それを支えるエンジニアは評価されない。なぜこんな矛盾が多くの日本企業で発生するのか不思議でならなかった。

その理由は「ひとり情シス」として IT 環境を立て直す過程で分かってきた。根底にあるのは流動性の低い雇用環境と、それを前提とした組織運営、そして過剰な協調性や労働集約的な発想だ。そのような特殊な状況で**エンジニアが幸せになるには、技術を身につけるだけでは足りなかった**のである。「**組織の中でうまくやる方法**」が技術面でのスキルとともに必要だったのだ。しかし、そんなことは誰も教えてくれないし、どこにも善かれていない。

長い間社内で評価されなかった状況が大きく変わろうとしている。経営層との距離が急速に縮まったことで、別の意味で大変にはなったが、IT 投資に対する理解を得やすくなった。投資の重要性を理解しない人に理解してもらうための資料作成から開放され、大きな仕事にも挑戦しやすくなった。

その結果、社内の IT 活用に弾みがつくだけでなく、自分のスキルもさらに高まっていく。**思い通りに動かない部下に悩まされる管理職とは無縁の下っ端で、自分の命令を忠実に実行してくれるコンピュータをたくさん従えて、大好きなプログラミングを満喫する一方で、投資の際には CIO（最高情報責任者）や CTO（最高技術責任者）のような力を持つ**。そこそこの報酬も得られるなら、エンジニアにとって最高の居場所と言えないだろうか。

もちろん、市販のソフトウェアやフリーソフトで解決できればそれでもよいが、**望み通りのソフトウェアを探すのには非常に手間がかかる**。要望ごとに毎回調査をするのは非効率であり、そもそも誰でもできる作業なので**わざわざエンジニアがやる必要もない**。一方、プログラミングは一度マスターすれば応用は簡単であり、作れば作るほど効率が上がり価値が増す。

エンジニアが優位な立場を持つには、サーバーの調達部分を握ることと、システムの調達部分を握ることである。それこそが仮想環境でありプログラミングなのである。私はそれを手に入れた。10人のIT部門で管理していた200台の物理サーバーを仮想化して一人で運営できているという話も、これにより理解できるのではないだろうか。

だが以前は、投資に積極的であったことやスピードが重視されていたこともあり、手っ取り早く成果を得られる「ベンダー丸投げ」が当たり前のように行われていた。

その結果、目先優先の個別最適システムを増やすことになり、運用面で非効率で高コストな体質を生む原因となっていた。IT部門が関わらないシステム構築も横行し、稼働間際になってサーバーの設置場所や電源の空きがないといった、初歩的なトラブルに見舞われることも少なくなかった。

IT部門が関わらないシステムは、担当事業部門の組織変更やプロジェクトの終了を機に放置されることが多い。放置されたサーバーやシステムは、最終的にIT部門が尻ぬぐいをするという暗黙の流れができていた。

ITの知識がない中間管理職に、ITのことを上に伝えるのは酷な話かもしれない。変に伝えて、経営から突っ込まれても答えられないだろう。だから、その状況は何をやっても伝わらないことを意味する。極端なことを言うと、トラブルさえ起こさなければ何をやっても分からないということでもある。

平社員の話は聞かないが、それなりの肩書のある人の話は聞くという状況はよくあることだ。以前コンサルタントと仕事をしたことがあったが、同じ事を言っているのに、経営者は私の話は聞かないが、コンサルタントの話なら聞いた。

そのときはコンサルタントから「経営者の説得は私がやりますので、技術的な部分をお願いします」と提案された。しかし、経営者には「ほら、コンサルに任せたらうまくいったらろう」と言われ、信頼関係の問題なんだなと思いついた。

IT部門がどうであろうと、ユーザーにとっては自分の仕事を全うするために、システムが使えればよいと考えているだけである。

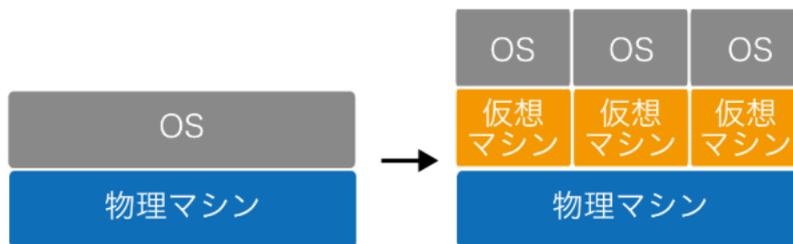
能力成果主義では、他人のことより自分のミッションを優先するのは当たり前と言えれば当たり前である。私自身ががんばっても評価につながらず、低い評価が長く続いたせいで、評価やミッションに対して無頓着になっているのかもしれない。

しかし、自動化したい時やシステムが欲しい時は、誰もが近寄って協力を求めてくる。居候する部門の仕事については皆で協力しようと言われ、その一員として駆り出される。しかし、こちらが困っている時は誰にも協力してくれない。

## P2V(Physical to Virtual)

~~~~~

### 仮想マシン



仮想マシンとは1台のコンピュータで複数のコンピュータを動かす技術のことです。

専門的な用語で少し難しいイメージを持ってしまいましたが、簡単にいってしまえば通常物理的に存在しているマシンを、仮想的に作り出して動かしているだけです。具体的には Windows PC 上で Mac を動かす、Windows PC 上で Linux を動かす……などの方法があります。

普通にハードウェアを稼働させるのでは、CPU などのリソースが余ってしまい本来の性能を引き出すことができません。そこで活用したいのが仮想マシンの技術です。

仮想的に複数のサーバを作り出すことで、通常であれば余剰するリソースを最大限に活用することができます。

~~~~~

データベースはリレーショナルデータベースというだけあって、いろんなデータと関係性を持っている。マスターメンテナンス作業はこの関係性を意識しながら、煩雑な登録変更削除処理をしていた。手順書はあったが、複雑な手順のため誤入力などの作業ミスもあり、それが原因で予期せぬ不具合や二次災害が発生し、調査や対策が難航することもよくあった。

おそらく、システム開発当時は費用削減のため、利用頻度が少ない割に開発費用がかかるマスターメンテナンス画面を作らなかったと考えられる。それにより長期にわたって、単価の高いエンジニアにしか

できない作業になり、エンジニアが本来やるべきことができなくなっていた。その損失を考えると、最初にケチったことの代償は大きい。

しかし、外部委

託でサーバーを IT ベンダーに握られ、「ユーザーがいじったら責任は持てない」と IT ベンダーに言われたら手も足も出せない。

### **COBOL をやらずに済んだ幸運**

当時、国内のシステム開発で使用するコンピュータは国産のメインフレームで、開発言語は COBOL という構成がほとんどだった。同期の連中が実践でプログラミングやシステム開発を学んでいる中、私は目的の無い勉強という名の暇つぶしをしていた。私とその「勉強」で使える環境は、国産メインフレームとは違い、主に米 DEC（デジタルイクイップメント）のコンピュータ VAX や UNIX だった。

数億円もしたらしい大型コンピュータを自由に使える機会はそうはない。しかし使い方を先輩に聞くわけにもいかず、自分で辞書を片手に英語の技術ドキュメントと格闘しなくてはならないことが非常に苦痛であった。しかし、**国産メインフレームとは違うコンピュータに関わったことが、その後の人生を大きく変えていくことになる。**

国内向けの開発では国産メインフレームで COBOL を使う中、海外開発部に配属された私は、いち早く UNIX 系 OS と C 言語に接することができた。これは非常に大きな人生の変化点であった。

使えない世代の中でも底辺に位置していた私がその後、技術者として力をつけていく最初のきっかけとなったのだ。今流行りの技術や世間の風評に惑わされず、自分の将来に役立ちそうな技術を選択することの大切さを実感する契機ともなった。もし最初の配属が国内開発部で、私も COBOL をやらされていたら今頃どうなっていたことだろうか。