

## 第2章 無し無し無しプロジェクト

前回は、今から17年ほど前、世の中がY2K問題(2000年問題)対策を進めた最終段階の1999年12月31日23時30分、私は汎用計算機のコンソール前にいて、机の上には最悪の場合を想定してヘルメット、消化器、ハンマーを置いていたことを書きました。いま思えば笑い話ですが、当時は真剣そのものでした。

### 1. 忘れられない同僚のひと言

私がプロジェクトリーダーをつとめる「Y2Kプロジェクト」では、残念ながら製品系で何件かのY2Kトラブルが発生しました。そのひとつ、トラブル対応のため2000年1月1日元旦の朝一番機、青森県の三沢空港に向う同僚がいました。

彼のいま一番の心配は、  
「飛行機、大丈夫かなあ(落ちないよなあ)、  
だったよ！」

とは、後日のプロジェクト打ち上げ会の席で打ち明けた話でした。彼のこの述懐は、私にとって忘れられないひと言になっています。



それほどまでにY2K問題というのは大惨事となる危機感があり、世界各国が緊張していました。したがって、世界中のどこの企業もこの2000年問題への対応のために、数年前からプロジェクトチームを組んでいました。それこそねずみ一匹通さないほどに万全な準備をする大プロジェクトだったのです。

### 2. Y2K(2000年)問題の大プロジェクト

私の担当した「Y2K対応プロジェクト」では、情報システムの改修は、20以上のサブシステムがあり、プログラム数も6000本に及ぶ莫大な規模でした。

プロジェクトのスタートは1996年、と2000年に向けてさらに4年遡ります。世の中は、「悲惨な大惨事までカウントダウンが始まっている！」との記事が溢れはじめていましたが、我々は、まだまだ危機感を感じていませんでしたね。

ですが、数ヶ月間の調査が進み、その影響する範囲、規模が明らかになってきました。しか

も、それに対応するリソースが貧弱であることが明確になってきたら、  
「これはまずい！ このまま2000年を迎えたらどうなるのか？」

### 3. プロジェクトプロブレム :ヒト無し、モノ無し、カネ無し

プロジェクトリソースとは、ヒト、モノ、カネ、情報ですが、Y2Kプロジェクトはこの時期、その大半のメドはついていませんでした。

Y2Kで対応しなければならないシステムは常に稼働しています。そのシステムサービスの人員も、通常業務を続けながらY2Kにパワーを割けなければならないのです。十分な余裕がある体制ならばまだしも、情報システム従事者は3K職場(キツイ、厳しい、帰れない)といわれていた当時は、そんな余裕のあるはずもなく、ヒト無し。

モノ(サーバなど)は、本番系とは隔離したテスト系を準備する必要があるのですが、全システムが対象ですから、ほぼ同規模が必要になります。しかしながら、コストパフォーマンスを絶えず求められていた現場です。当然ながらハード・サーバにそんな余裕もなく、モノ無し。

さらに、この仕事は、端的に表すと今実現している機能を維持する事ですから、何か効果を産む改善、改革でもありません。当然ながら潤沢な資金を黙って提供してくれる経営層がいるはずもなく、カネ無し。

ボスに相談に行きました。

「2000年が来ることは子供だって判ってたはずだろ、そんなことも考えていなかったのか？」

お叱りが第一声でした。

まさに、「無し、無し、無しのプロジェクト」です。

ただし、専門家の間でもマスコミでも、2000年問題にしっかり対応しないと大災害が発生する、と大騒ぎとなりつつある状況でした。

### 4. 見切り発車！

このプロジェクトは、この大災害情報にプラスして、弊社が顧客に提供している製品からもY2Kトラブルが内包していることが明らかになった。一変、事業所横断プロジェクトの位置づけでスタートすることになりました。

しかしながら手放しでは、喜べません。潤沢な予算もつくわけもなくスタートです。ヒト無しが

改善出来る当てもないスタートです。スタート当初から暗雲の漂うプロジェクトだったのです。まさに見切り発車です。

## 5. プロジェクトの進め方、一見順調だったが？

Y2Kプロジェクトは、計画面と技術面と費用面の3つの面からはじまりました。計画面はまず、ターゲットとスケジュールのネットワーク化です。

システム単位に担当を決め、納期(完了時期)を定めることからはじめました。特徴は、一般的にプロジェクトは遅れることが日常茶飯事です、時間は止まりません。2000年はカレンダー通りに確実に来ます。つまり遅れは許されないプロジェクトです。

と言うことで1999年10月をそのターゲットに決めました(スケジュールのネットワーク化については別の機会に紹介)。

次に、対応方法の検討など技術面がクリアされないと、見積もりも出来ません。そこでプログラム改修方法(プログラムを直接改修する、サブルーチン化し組み込む、ファイル形式を [RDB リレーショナルデータベース](#)に変更する)などの標準ルールを決めました。また、対象箇所を検索するプログラムも、自社特有の条件も対応するように自社開発し、漏れ撲滅も図っています。これで対処法が定まりました。

これで、納期と手段が定まりました。また、現体制でプロジェクトに投入できるリソース(人員)もこの頃には、あるていどは確保できました。この結果、足りないプログラマーの数と工数(人工)も見えてきます。

最後は、費用面です。ようやくですが、3つの面の手順をまとめたところで費用の予算化も可能となりました。これから3年半に及ぶ計画が立案され、スタートへとつながりました。

## 6. 破綻満載のプロジェクト

どこまでは、一見まともな滑り出しにみえます。

しかし、もともとが「無し無しプロジェクト」です。実際の計画書には、当然のごとくあちらこちらにほころび(空欄)のある「絵に描いた餅のプロジェクト」です。

プロジェクトリーダーなのに、「ちょっと見ると破綻の満載のプロジェクトだなあ、これでなんとかなるのかなあ？」と、漠然とした不安感で満載されていたプロジェクトであるのが実感でした。

破綻プロジェクトの顕著な特徴は、初期段階に十分な準備もなく走り始めるがありません。

「Y2Kプロジェクトは、まさに失敗の王道？」のようです。  
しかし、期限は明確で遅れは許されないプロジェクトです。  
こちらの都合で時間を止めてくれる訳もなく、まさに前途多難な船出だったのです。

第1章	第2章	第3章	第4章
-----	-----	-----	-----