



## IT・ソフトウェア産業の課題について

株式会社 NTTデータ 代表取締役社長 はまくち 濱口 ともかず 友一



### はじめに

ただいま御紹介いただきましたNTTデータの濱口でございます。今回このような機会をいただきまして、大変感謝しております。今日は「IT・ソフトウェア産業の課題」と題してお話をさせていただきたいと思っております。

これからはもう、ソフトウェアとコンピュータが世の中のあらゆることと関係を持つようになっていくと思っております。本日は、そういう社会の到来を踏まえて、三つの課題についてお話をさせていただきたいと思っております。一つ目は、コンピュータシステムについてです。経営者の方々は、現状のコンピュータシステムに必ずしも満足されていません。なぜ満足されていないのか、どのようにしていけば良いのかについてお話しさせていただきます。二つ目は、お客さまとの契約について、何が問題でこれからどのようにしていくべきなのか。そして三つ目に、ソフトウェア産業の人材育成をどのようにしていけば良いのかについて、お話をさせていただきたいと思っております。

### ソフトウェアに対する価値観の差

このようなことをお話ししようと思った理由は、昨年MITのマイケル クスマノ教授とお話をした際、「日本のソフトウェア産業の生産性や品質は大変優れている。5~6年前のデータで見ると、システム導入後の不具合件数は米国やインドの1/10程度だし、プログラミングの生産性も高い。でも、日本のソフトウェア産業が必ずしも成功しているとは思わない。ビジネスで成功しているのは米国であり、サイエンスとして成功しているのはヨーロッパである」と言うのです。日本のソフトウェアは、ドメスティックな産業であり、そしていわゆる工場生産型であると彼は分析していました。では、日本のソフトウェア産業をグローバルに成功させるためには、何を变えれば良いのかと聞きましたら、大学教育であるとのことでした。大学教育を变えるにはどうすれば良いのかと言うと、「大学の先生になりそうな優秀な学生をアメリカの大学で学ばせれば良い。そうしたら我々がきちんと教育して、日本に帰すから」と言うておりました。中国やインド、開発途上国はそのような発展プロセスを確かに経ているので、ま

ざら冗談として切り捨てるわけにもいかないと思います。いま一度、大学のソフトウェア教育の在り方を考えてみる必要があるのかもしれない。

ソフトウェアの特徴は、皆さま御存じのように目に見えないということと、基本的に何でもできるということです。これが最大のメリットであるとともに、デメリットにもなるわけですね。目に見えないということと、何でもできるということ、これに対する価値観が日本ではなかなか理解されません。ここに大きな問題点があると私は思います。これまで、日本ではソフトウェアをハードウェアと同じように製造物であると捉えてきた。このため、工場の中での生産性をどうやって上げていくか、品質をどうやって上げていくかということを中心に考えてやってきた。ここところが、米国あるいは欧州と、捉え方が違ってきた原因だろうと私は思います。やはり、ハードウェアとは似て非なるものであると考えるべきなのです。

これは聞いた話なのですが、自動車は大体2~3万点の部品からできていて、ジェット機はその10倍くらいの部品からできているのだそうです。宇宙ステーションになると更にその100倍くらいだそうです。大規模なソフトウェアは、部品点数で言うと宇宙ステーションに匹敵するか、それ以上であるという説もあるそうです。そのように複雑なソフトウェアをどのようにして作り上げていくかということが、非常に大きな問題でして、よくソフトウェアを家に例えたり、自動車に例えたりするようなことがありますけれども、私はそのくらいの複雑さのものに例えること自体が間違っているのではないかと思います。ソフトウェアは、もっと違った複雑なものであるという認識の下に立つことが、グローバルに通用する産業となるためには必要ではないかと思っています。

### 1. なぜ、経営者はコンピュータシステムに満足していないのか

経営者から見て、満足のいくコンピュータシステムはないと言われる。これも日米の比較をしますと大変よくわかるのですが、日本の経営者は、コンピュータシステムを経費節減の方法と捉えているという調査結果があります。これに対



して、欧米の経営者は、コンピュータシステムを新しいサービスを作っていくためのツールであるとか、あるいは顧客を拡大していくためのツールであると捉えており、前向きな投資という考えでシステムを作っています。日本は経費節減、欧米は前向きな投資というところがかなり違うわけです。私は、コンピュータシステムは投資と捉えて、どういうリターンを期待するかというところから始めるのが正しいのではないかと考えています。

エンタープライズアーキテクチャー (EA:Enterprise Architecture) という概念がございます。EAの父と呼ばれる、ジョン・ザックマン氏が20年ぐらい前に考案した概念で、EAは企業の戦略を実行に結びつける「設計図」と言われます。そのEAに沿ってコンピュータシステムを作っていくことが、投資に対するリターンを確かにしていく一つの方法であろうと思っています。EA、例えば「ザックマン・フレームワーク」は、「スコープ」「ビジネスモデル」「システムモデル」「テクノロジーモデル」「詳細仕様」の5段階構成でできています。最上位の概念であるスコープは、企業の戦略を意味しています。例えば、ある大手家電メーカーが戦略商品の世界同時発売ということを企図されましたが、その場合のスコープは世界同時発売ということになると思います。このスコープに基づいて、「ビジネスモデル」を作っていくのが第二段階目になります。第一段階のスコープは、まさに経営者が考えるべきことですが、第二段階のビジネスモデル、これは企業の経営企画部などの戦略部門のスタッフが考えていくことです。世界同時発売というスコープを実現するために何をしなければいけないか。例えばサプライチェーンマネジメントを全部変えなければいけない。それから、工場のラインを変えなければいけない。いろんなことを変えていかなければいけない。ここが大変重要になってきます。

そして、次の段階がシステムモデルというところになるわけですが、上位のスコープやビジネスモデルの検討なしに、システムモデルの検討から入ってしまうシステム開発プロジェクトがとて多いのです。ここに、大変大きな問題があると私は考えています。一番大事なことは、ビジネスモデルのところでした、ここに会社の優秀なスタッフを集めて、はっきりとした変革なりビジネスのデザインをしていくことが必要だと思えます。このビジネスモデルのところを外務からサポートするのが、いわゆるコンサルタントや、当社で言う「IT

ディレクター」といった役割になります。このビジネスモデルとシステムモデルのつなぎの部分をお客さまとベンダーが協力してきっちり仕上げる。ビジネスモデルとシステムモデルの間で何回も何回もフィードバックを繰り返して、戦略に沿った、そして価格的にも満足できるようなデザインをしていく必要があります。



ITUクラブで講演する筆者

## 2. 契約の問題

二つ目の契約問題ですけれども、これは日本では大変大きな問題でして、上述のザックマン・フレームワークの手順で言いますと、システムモデル以降の設計から開発・試験をして御提供するところまで、全部まとめていくらというような契約が非常に多くあります。例えば、会計のシステムを作してほしい、1億円で作ってほしい、サプライチェーンマネジメントのシステムを作りたい、3億円で作ってほしいということですが、ここで一番の問題は、設計がまだ何もできていないということです。先ほど、家や自動車などで例えるのはおかしいと申しましたが、何を作るかの設計書がないまま、値段や納期が最初に決まってしまうということが非常に多いわけです。このようなやり方でシステム作りを始めますと、お客様の方はもう値段と納期が決まったわけですから、いくらかでも機能を盛り込みたいということになる。他方、我々ベンダー側は何しろ予算や納期の中に収めなければいけないということで、お客さま側と我々ベンダー側のプロジェクト・マネジャーあるいはSEの間で、機能の追加や削減についての消



耗戦が始まるわけです。私は、これはお互いに非常に不幸なことだと思います。

これを防ぐためには、まずビジネスモデルからシステムモデルへつなぐ作業をしなければいけない。その上で、システムモデルをきちっと仕上げるのが、要点になると思います。そして、このシステムモデルに基づいて、何十人月、あるいは何百人月かけて、コンピュータシステムの設計書を完成させるわけです。この設計工程については、人月当たりの単金で実施させていただくことが良いと思います。その後、完成した設計書を前提に見積りをすれば、ベンダー側としても、かなり正確な開発コストを見積もれます。この段階で、いわゆるコンペなり入札なりをかけるということも良いと思います。このように、設計と開発を二つのフェーズに分けて行うことで、見積りの精度は上がりますし、上述したような消耗戦の必要がなくなるのです。私は、これがベストな方法じゃないかと考えております。

そのためには、私どもSI企業は、お客様にできるだけシステムの完成イメージを分かっていたらいいような設計を、このシステムモデルの中でしていかなければいけない。サービスレベルもきちっと決めていかなければならない。それから進捗度合、品質などについても、できるだけ「見える化」の努力をしていかなければいけないと思っています。

開発については、設計書に基づいて価格を見積もるわけですが、従来からの人月×単金による見積りだけではなくて、いろいろな方法を試していかなければいけないと思っています。例えば、ファンクションポイント法という方法がありまして、どれくらいの機能があるのか、画面数とか帳票数なども加味して見積もっていくやり方になります。もっと先には、システムだけでなくビジネスモデルも合わせて評価する必要がありますが、バリュー・プライシングと言って、どれくらいのリターンがあるかによっていくらで作るかを見積もる方法もあります。我々とお客さまでリスクをシェアしていくという方法もあるかと思っています。

ソフトウェアというのは、何でもできるわけですが、作ればそれだけのお金がかかるものです。ですから、お客様が投資対効果を高めたければ、できるだけ作らない方がいい。これはどういうことかと言いますと、パッケージを使う、あるいは共同センターを使う、それからできるだけ作らないでいいように機能を削減するということです。ある先生がお

っしゃっていましたが、ソフトウェアというのは作らないことを決めることが、作ることを決めるのと同じぐらいに重要なことだそうです。

それからもう一つ、これは政府の調達等でよくあるのですが、設計と施工を分離することがあります。私は、ソフトウェア開発の場合はできるだけそこで業者を分離しない方が開発の効率が良いと思います。設計と施工で業者を分離すると、設計時に獲得したノウハウが活かさないためになんかのロスが生じるのです。もちろん、競争原理を働かせる、価格の妥当性を検証するなどの目的で、コンペや入札をすることは重要な方法だと思いますが、業者を分離しなくても、見積り等をきちっと確認することで、価格の妥当性を確保しつつ効率化が図られ、むしろ責任ある開発が行えるものと考えております。

### 3. 人材の育成

最後に、人材の問題ですけれども、日本の大学教育を含めて、ソフトウェア人材の育成を本当に真剣に考えていかなければなりません。最近、ソフトウェアの現場は、3Kとか言われていて、若い人に人気なくなってきました。日本だけかと思っておりましたら、欧米でも結構人気なくなってきました。傾向を見てみますと、先進国では理工系の人気落ちてきて、いわゆるBRICsではかなり人気があるという状況です。あまり体を使わない、汗をかかないでお金を稼ぐというのが、今の若い人に受けているのではないかと。そういうことがマスコミでも取り上げられて、一層そういう傾向を助長させているのではないとも言われています。また、製造業では、瑕疵が起きると社会的に厳しく批判されることもあり、若い人に人気がないのではないとも言われています。そういうこともあるかとは思いますが、私はこのソフトウェア開発について、日本の場合、面白さとか達成感とか、少なくなってきたということが一因ではないかと思っています。

日本のソフトウェア開発というのは、ファクトリータイプ（工場生産型）と言われておまして、昔のチャップリンの映画にあったような、流れ作業的な考え方が中心になっていません。現在の我々のソフトウェアの作り方というのは、例えば、設計者は設計だけをする人、プログラマーはコーディングだ



けする人と、上流から下流へブツ切りの作業になっているわけです。他方、製造業では、最近セル生産という方式で、製品の組立を、一人や一つのチームが最後まで作っていくという方法も出てきています。やはりソフトウェアも、私たちが昔やっていたころのように、設計からコーディングまで全部1人で担当する、そういう時代に少しネジを巻き戻す必要があるのではないかと考えています。昔の職人芸的なやり方と工場生産の流れ作業的なやり方、たぶん正解はこの中間にあるのだらうと言われておりますけれども、私どもは、創造性とか達成感をソフトウェア開発者に与えられるような生産方式を、これから考えていかなければいけないと思います。

また、ソフトウェアの最終生産物は0か1のコードです。その前段がソースリストと言われており、目に見えるものとしては、ソースリストが最終生産物になります。この最終生産物を作るのはプログラマーなのですが、日本ではプログラマーを、大変軽視してきたと私は思います。これから、もっとプログラマーを重要視していかなければいけない。プログラマーの処遇を良くして、復権させなければいけないと思っています。生産性の良いプログラマーと悪いプログラマーとを

比べますと、大体20倍くらいの能率差があると言われております。いいプログラマーにはいい処遇をし、短期間の講習を受けただけのようなプログラマーは相応の処遇で経験を積んでもらう必要があります。そのためにはスキルを評価・認定する資格制度も必要だと思っています。

とりとめないお話をしましたけれども、目標としては、このソフトウェア産業というものをエンジニアリングにしなければならない。そして、グローバルに通用するような産業に育てていかなければいけない、ということを念頭に置きながら、私どもも努力していきたいと思います。本日御出席いただいている皆様の中には、同業の方もいらっしゃいますし、お客様もいらっしゃいます。お客様と我々ベンダーが協力して、グローバルに通用するような産業に育てていかなければいけないと私は思っております。是非、これからも御協力のほどを宜しくお願いいたしまして、今日の話を終わりにさせていただきます。

御清聴ありがとうございました。

(2007年3月19日第355回ITUクラブ講演より)



ITUクラブで講演する筆者