

成功するシステム開発に向けて

大和田 崇

1. システム開発は 15%しか成功しない

国内では、官公庁、民間合わせて毎年約数兆円の IT 投資が行われている。景気低迷が長引く中、IT サービス産業が比較的堅調に推移してきたのは、官公庁による積極的な IT 投資によるところが大きい。e-Japan に代表されるような政府の IT 化政策は、需要サイドからの財政政策という意味だけでなく、成長セクターへの弾力的な予算配分によって、その分野の競争力を国際的に強化するという、産業政策的な側面も見逃すことができない。

IT を利用する民間企業においても、従来の単純な情報システムの枠組みを脱却し、戦略的な IT システムを構築することによって、国際的な競争力を強化することが期待されてきた。

つまり IT は利用セクター、提供セクターともに、経営体質を強化する重要なツールであり、日本経済の機関車としての役割を求められてきたといえよう。

ところが、多額の資金をつぎ込んだ IT システムの開発案件の多くが、当初の予定通りには稼働していないとの声が多く聞かれるようになってきている。

「15%の成功確率」。これは何の数字と思われるだろうか。実は、米国の主な IT 投資プロジェクトの実績をもとにした統計的な IT 投資の成功確率である(図1)。残念ながら国内プロジェクトに関する明確なデータは存在していないが、プロジェクト管理が甘いといわれる国内プロジェクトの成功確率はさらに低いことが予想される。一部の試算では、国内の IT 投資のうち約 7 兆円、投資総額の約半数がいわゆる不良資産化して実際に稼働していない状況だという。

この米国のデータによれば、稼働に至らず中止してしまうプロジェクトが 30%、失敗プロジェクトが約 50%という結果になっている。つまり、プロジェクトの 3 分の 1 が完成に至らず、破棄されてしまうのである。失敗プロジェクトの内訳は、スケジュールの遅延とコスト・オーバーである。スケジュールの遅延は平均 1.9 倍、コスト・オーバーの平均は 2.1 倍である。

無事システムの稼働にこぎつけたとしても、その多くが、予算と比較して平均 2 倍のコストとスケジュールをかけているということになる。

もちろんこの数字は統計データであり、すべてがこのようなプロジェクトばかりではないはずだ。しかし、実際のシステム構築に携わっている人にとっては、かなり現実的な数字ではないだろうか。

それでは、IT システムの開発プロジェクトはなぜこのようにうまくいかないのだろうか。

国内のアンケート調査では、納期が遅延してしまう最大の理由は管理力不足、特に上流工程での見積もりや計画の不備によるところが大きいという結果が出ている。また、コスト・オーバーの原因としては、発注者側の体制が不十分、開発者側の見積りが不完全、業務知

識の不足などが列挙されている。

2．投資目的が不明確なシステムは失敗する

情報システムはその構造上、開発途中で発生した欠陥は、実際にシステムを稼働させるまで顕在化させることが難しい。

IT プロジェクトが失敗する原因の多くがシステム設計の段階に発生している。システム要件があいまいなまま開発をスタートさせたり、開発途中で仕様変更を何度も繰り返すことが欠陥の発生を誘発する。システムに求める機能が明確になっないことや途中での変更が増えることによって、設計段階のミスが増加するからである。

例えば、設計変更が重なり、スケジュールの遅延した規模の大きいプロジェクトでは、予定通りに開発が進んだ場合に比べて、潜在的な欠陥数が3倍に増えると言われている。

発注者側は、システム投資の目的を明確にし、業務のどの部分をシステム化するのできるだけ詳細な計画を作成する必要がある。受注者は利用者側が持つニーズを的確に理解し、システム設計に生かしていく必要がある。つまり発注者側は、システム構築に対する知識が求められ、受注者側は、発注者側の業務に関する知識が求められるのである。両者がうまく噛み合えば、スムーズにシステム開発を実施することができるだろう。結果として、プロジェクトが失敗する確率も下がってくることになる。

ところが、現実には双方のコミュニケーションがうまくいかないことが多い。

発注者側の担当者は、当然のことであるがシステム構築の専門家ではないことが多い。かといってシステム開発は大規模なものは数年に一度しかないため、専門の人員を常に確保しておくこともあまり現実的な選択肢とはいえない。このため、発注者側の担当者は、通常業務をこなしながら、新規のシステム開発を行う形になる。

受注者の側は、効率よくプロジェクトを進めるために、開発経験の多いアーキテクチャやパッケージを使いたいと考える。また発注者のニーズが的確にまとまらない状態で開発を進めた場合、非常にリスクが大きいものになってしまうことから、できるだけ、自社が開発しやすいシステムになるよう、本来発注者が行うべきシステム化計画の部分についてもより積極的に関与していくことになる。

結果として、本来発注者が求めていたシステムとはまったく異なるものが開発され、途中で大幅な変更が加えられることになる。先にも触れたように、開発途中で設計変更は、欠陥を誘発し、これが納期の遅れやコストのオーバーという形で顕在化してくるのである。

3．ITサービス産業はまだ発展途上？

発注者はできるだけ安い価格で必要なシステムを構築したいと考えている。逆に受注者はできるだけ高く自社のシステムを販売したい。当たり前のことであるが、本来、両者は利益相反の関係にある。

情報産業白書に掲載されたアンケート結果によれば、開発者と受注者の意識にはかなりのズレがあることが分かる。発注者は「拡張性・柔軟性の高さ」や「低価格でのシステム

開発」を希望している。ところが、受注者側におけるこれらの意識は薄い。逆に、「ネットワークの構築・保守」「先端技術の採用」など技術的な部分に大きな関心を寄せている（図2）。

このような発注者と受注者のスタンスの違いはある意味で当然のことである。適正なマーケットが形成されているのであれば、両者はお互いのスタンスの違いを認識した上で交渉を行い、双方が納得できるレベルで価格や納期が決定されるはずである。

最大の問題は、国内のITサービス業界には、このような健全なマーケットがまだ十分に形成されていない点にある。

通常、モノやサービスを購入する場合、それが高額のものであればなおさらだが、契約書を結ばないでやり取りを行うというケースはまずあり得ない。ところが、システム構築の場合は、そのようなケースが少なからず見受けられる。交渉の段階において、何をいくらでいつまでに納入するのか明確に定めることをしないまま、一部の開発がスタートしてしまうのである。

また契約が成立したとしても、修正についてはどこまで無償なのか、品質はどのレベルまで受注者が保証してくれるのかなど、細かい点についてはあいまいなままになっていることも多い。

このような不透明な契約慣行の存在は、発注者、受注者双方にとって不利益である。極端なケースでは、発注者側の責任によるシステムの修正についても開発者に負担を強要したり、逆に、受注者側が不備のあるシステムを納入して、有償で修正するというケースもある。ようやくこの業界にもサービス・レベル・アグリーメントという概念が普及しつつあるものの、現段階においてはまだ発展途上と言わざるを得ない。

4．システムを計るモノサシがない

このようなあいまいな契約慣行がなかなか解消しない背景には、情報システムという商品そのものが目に見えにくい存在であるという点があげられる。このため、金額や品質について数字で評価するためのモノサシがなかなか普及せず、結果として、契約を厳密に行う商習慣が定着しなかったと考えられる。

システム開発の成功確率を上げ、健全な競争市場を確立するためには、システムのコストや品質を定量的に評価する尺度の導入が不可欠である。

従来のITシステムの見積りは工数と単価によって設定されていることがほとんどである。工数は、1人のエンジニアを1ヶ月の期間投入することを示す「人月」と呼ばれる単位を用いるのが標準的である。この工数に、1ヶ月あたりのエンジニアの単価をかけて最終的な構築コストを算出する。

ところが、この工数をベースにした見積り方法は公正さという観点では適切なものであるとはいいがたい。システム開発は極めて属人性が高く、同じシステムであってもどのエンジニアが担当するのかでその生産性は大きく異なってくるからである。筆者のシミュレーションでは、エンジニアの能力によって最大6倍のコストの違いが生じる可能性がある（図3）。

さらに問題なのは、投資するシステム資産について金額以外のモノサシが普及していないことである。例えば土地であれば、坪というモノサシがあり、坪あたりいくらを投資し、いくらを上げたのか容易に評価することができる。

ところが IT システムの場合、金額しか基準がない。同じレベルのシステムであることを示す基準値がないままに、どちらが良い、悪いというような議論が行われている。

5 . システム定量化の方法論

IT システムを定量的に評価する方法論には大きく分けて二つの方向性がある。一つはシステムが持つ機能の量を測定する方法論、もう一つはシステムを開発する組織（ベンダー）の能力からコストを求める方法論である。

システムが持つ機能を求める方法論として代表的なものがファンクション・ポイント法である。

ファンクション・ポイント法は、外部から見える機能の量を分類しその総数を数えるというものである。1970 年代に米国の IBM で開発された。

ファンクション・ポイント法では、システムが持つ機能を 5 つに分類し、それぞれの機能が何個あるのかを計測していく。最後に機能ごとに定められた重み付けを行い、システム全体の機能量を算定する。

ファンクション・ポイント法は、システムが持つ機能の量とコストには一定の相関関係があるとの立場に立っており、システムの構造については着目しない。このため、システムの利用者側からは非常になじみやすい指標といえることができる。

これに対して、システムの構造や開発組織を軸にコストを評価する方法論として代表的なのが COCOMO 法である。

COCOMO 法は、米国の政府調達の実績をもとに作られた数式モデルで、システムの構造や開発組織の能力などを評価し、パラメータとして入力することで開発コストを算出するというものである。

COCOMO 法のベースになる実績データは現在も更新されており、常に最新の統計データをもとにコストを推定することができる。

ファンクション・ポイント法は、人月に代わる見積りの指標として注目を集めている。すでに岐阜県などでは、試験的にファンクション・ポイントをベースとした入札が行われている。また、会社全体の IT 資産をすべてファンクション・ポイント・ベースで定量化するという試みも一部の企業では始まっている。

6 . 定量化の意義

システムのコストや品質について、定量的な指標で評価を行うことができると、開発計画の立案や契約の方法も大きく変わってくる。コストの上限が明確になり、成功や失敗の定義もはっきりしてくることになる。結果として、システム設計の精度が向上し、プロジェクトの成功確率も上がってくるだろう。

また、定量的な評価手法が定着してくると、IT投資の評価もすべてキャッシュ・フロー・ベースで一元化できるようになる。企業経営の世界では、土地や設備といった投資に関する評価は、その資産が生み出すキャッシュ・フローを現在価値に割り戻す手法が一般的になっている。ITシステムの投資についても、同じ枠組みで比較検討が可能となる。

これまでのIT投資はどちらかというと、IT化そのものが目的であり、導入の目的と求める効果について十分な議論がなされてきたとはいいがたい。そのため、IT投資の目的として「業務効率の改善」といったような定性的な表現が使われることが多かった。

確かにITを導入することによって、業務にかかる工数や時間を減少させ、効率化を図ることは可能である。しかし、その効果を数字として顕在化させるためには、IT導入によって余剰となった人員を収益部門に再配置するといった具体的なアクションを伴わなければならない。

キャッシュ・フローによる評価の枠組みでは、企業の価値はその企業が将来生み出すキャッシュ・フローの現在価値によって決定される。つまり、極論を言うと、キャッシュ・フローの増加を伴わない投資は行うべきではないということになる。IT投資についても同様の枠組で評価することができれば、投資目的や具体的なアクションもより明確になってくるはずである。

最近、システムの構築コストに加えて運用コストをいかに評価するのかについても注目が集まってきているが、運用コストの評価は、キャッシュ・フローでの評価と非常に相性がよい。運用コストは、マイナスのキャッシュ・フローとなるので、システムが生み出すキャッシュ・フローから運用コストをマイナスした上で現在価値を求めればよい。これによって、システムのライフサイクルコストについてもより明快な結論を導き出すことができるようになるだろう。

7. IT版コンストラクション・マネジメントの必要性

建設業界では、より透明性の高いプロジェクト遂行のためコンストラクション・マネジメントの導入が進んでいる。

コンストラクション・マネジメント企業は、受注企業とは独立し、発注者の側に立って、ゼネコンや設計会社と交渉を行う。自らは設計や施工を行わないため、発注者の利益を最大限追求することができる。

IT業界における大手ベンダーは建設業界におけるゼネコンに相当する。上流工程を担当するコンサルタントは、建設業界における設計会社（建設コンサルティング会社）といえよう。両者とは立場を異にするコンストラクション・マネジメント企業が介在することによって、コストや品質の透明性を保つことができるようになるのである。

米国におけるシステム開発は、一般に発注側の権限が非常に強いといわれている。担当者にはシステムの専門家を配置するか専門のアドバイザーを指名していることも多く、発注側の体制はかなり強固といえよう。

日本の場合、発注側の体制が一般に弱く、システムを設計するコンサルタントがアドバイザーを兼務している状況といえる。定量的なシステム評価の枠組みと、コンストラクシ

ョン・マネジメントの導入が一般的になれば、システム構築に関する諸問題は解決に向けて大きく前進するはずである。

**この原稿は、2003年4月に書かれたものです。*