

ユーザー参加型のシステム開発

—ユーザー合意における“翻訳”の必要性—

システム開発における上流工程での“ユーザー合意”の重要性が強調されて久しく、ユーザー合意を得やすいアジャイル開発などさまざまな開発手法が提案されている。しかし依然として「ユーザー合意が不足していたためにプロジェクトが失敗した」という声を聞くことが多い。本稿では、その理由と、失敗しないためのポイントについてあらためて考えたい。

失敗をもたらすユーザー合意の不足

システム開発プロジェクトで、納期・品質・コストのいずれかで目標を達成できないとき、これを「失敗」と呼び、一説によると7割のプロジェクトは失敗だという。失敗の要因としては「上流工程でのユーザー合意の不足」「複数システム間（複数ベンダー間）のトラブル」など、複数関係者の意思疎通の難しさが多く挙げられる。

本稿では、ユーザー合意の不足について、主に大規模開発に絞ってその原因や解決策について論じたいと思う。なお、ここでいうユーザーとは、社内システムのエンドユーザーである社員（従業員）とし、大規模開発とは、推進体制にそれなりの人数が必要で、スケジュール管理、タスク管理が必要な規模（全体開発工数が500人月を超えるレベル）を想定する。

昔から上流工程でのユーザー合意は難しいといわれるのは、実物を見たり触ったりすることが外部仕様（システムが備えるべき機能やデータの流れ）を決める段階でできないからである。これは、新規開発でもパッケージ利用でも同じである。パッケージでは、その

製品の標準的な使用法を試してみることは可能だが、実際にユーザーの要望を反映した実物を触ることは、やはり外部仕様を決める時点ではできない。

この話を家の建築に例えてみよう。例えば設計図で窓の位置や仕様を見せられても、光の入り具合や開け閉めのしやすさは想像するしかなく、それでいいと思っても本当に合意したことにはならない。完成してから「思っていたのと違う」ということになれば、これは失敗である。これと同じことがシステム開発でもいえる。システム開発が建築よりも難しい点は、システムは目に見えにくく、受け入れテストがユーザー合意から1年後、2年後に行われるケースも多いということである。

システム開発にはさまざまな工程があるが、一般的にはシステム化構想・計画から要件定義、そして外部設計（外部仕様の設計）まではユーザー合意を行う必要がある。専門的な知識のないユーザーからすると、よく知らない言葉で詳細に書かれたシステム設計書を大量にレビューしなければならない状態が続く。担当している現業との掛け持ちであったりすると、現業に追われてモチベーションを維持できず、「よく分からないが、まあ大



丈夫だろう」と表面的な合意をしてしまう。開発者（情報システム部門や外部の開発ベンダー）にしても業務のことを熟知しているとは限らない。設計書のとおりにつくったシステムが、後のテストの段階になってユーザーから「意図していたのと違う」と言われてしまうことはよくある。これがユーザー合意不足の実態である。

大規模開発における課題

近年は、インターネットやスマートフォンの普及などにより、誰でも当たり前のようにITを活用しており、社内システムのユーザーも以前よりは具体的にユーザーインターフェースをイメージできるようになっている。また、業務の複雑さが増しているため、外部仕様の決定には現場のユーザーの意見が大きくなる。ユーザー合意の重要性はますます高まっているのである。

もちろん、ユーザー合意の重要性を考慮したシステム開発・導入の手法も登場している。クラウド型サービスを利用する場合はそもそも「開発」を必要としないため、1週間～1カ月程度で、ユーザーの要望を聞いて実物に落として確認することができる。アジャイル開発もそうした手法の1つで、全体を小さな範囲に区切り、設計からテスト、リリースを短期間で繰り返す。パッケージを利用した開発も、完成に近い形を早い段階で確認できるという意味ではアジャイル開発に近い発

想といえる。2年後といったはるか先ではなく、スタート時の記憶も新しい1カ月後には出来上がった実物が確認できれば、ユーザーにとって心強いだろう。

しかし、複雑な基幹系の業務システムの場合、自社独自の業務のやり方があり、システム間連携も密になることから、カスタマイズができないクラウド型サービスでは対応できず、また細かく区切ることが困難なためアジャイル開発には向かないのである。このような複雑なシステムは、従来どおりの大規模開発の手法を取ることが多い。すなわち、前節で述べた課題は依然として残るのである。

“翻訳”の重要さ

大規模開発できちんとユーザー合意を行うためには、やはりユーザーが頑張るしかないのか、ということそうではない。ユーザー合意不足のそもそもの問題点は、ユーザーの言葉（業務の言葉）が開発者の言葉（システムの言葉）に正しく置き換わっていないことである。われわれはこの言葉の置き換えを“翻訳”と呼んでいる。翻訳がきちんとできていれば、小規模開発であろうと大規模開発であろうと問題なく遂行されるはずである。

それでは、翻訳とは具体的にどういうものを指すのだろうか。次ページの表1にシステム開発における翻訳のポイントを挙げてみよう。プロジェクト内のメンバーに、このようなポイントを押さえ、適切にドキュメント化

表1 システム開発での“翻訳”のポイント

分類	項目	チェック内容
ドキュメントの書き方	構造	目次を見ればストーリーが読み取れるように構造化されている。
	目的・位置付け	読み手に立った目的・位置付けが記載されている。
	言葉の定義	ユーザー業務の独自の言葉およびシステム用語が定義されている。
	記載の粒度	業務的な観点で適切な粒度で要約されている。細かい要件は参考という位置付け。
確認の方法	網羅性の担保	しかるべき業務的な分類がされ、重複や漏れがないことが分かりやすく説明されている。
	確認ポイント	ユーザーに何を確認してほしいかが明確になっている。確認ポイントに対してユーザーはYes/Noの回答で原則OKとなる。
進め方	ユーザー意見の発散への対応	会議の目的から外れる発言に対して、本来の確認ポイントに誘導することができる。
	ユーザーとの意思疎通	必要に応じてホワイトボードなどを使って認識を合わせることができる。

または説明ができる人材が果たしているだろうか。

これらは、システム開発に限らずビジネスで相手へ物事を伝えるための当たり前の能力である。しかし、それが業務の特質とシステム開発の専門性を理解した上でとなると一段とハードルが上がる。

翻訳が最も重要となるのは「要件定義」と「外部設計」の間である。すなわち、定義された要件に基づいて新しい業務を設計するタスクをユーザーの視点で行える（翻訳できる）ことが、結果としてシステム開発全体の品質を向上させるのである。

翻訳が正しく行われないと、システム開発ではその分、減点が発生してしまう。積み重ねられた減点は、後工程で大きな反動となって表面化する。修正しようとすれば大きな手戻りが発生するため、ユーザーが納得できない状態のままリリースを迎えるか、リリース

を遅らせる（コスト増加を伴う）かの選択を迫られることになる。

誰が翻訳能力を持つべきか

では、翻訳能力は誰が持っていればよいのか。この問いに答える前に、ユーザー合意の一方の当事者であるユーザー側の体制を確認しておこう。

システム開発におけるユーザー側の体制には、ユーザーが所属している事業部門の部門長はもとより、現場で業務の中心となっているキーパーソンも必要である。これは、ユーザー合意が、後工程で「それは違う」と言われないためのものと考え、他のユーザーへの影響力がある人が必要という点について説明は不要であろう。もし、そういう人をシステム開発に割けないのであれば、事業部門としてもシステム開発のリスクが大きい点は認識しておくべきだろう。

さて、事業部門のキーパーソンを開発体制に配置できたという前提で、誰が翻訳能力を持っているべきかという議論に話を戻そう。ユーザー側が翻訳能力を持っていることは、理想的ではあるが現実的な話ではないだろう。ユーザーは当然ながらシステム開発の専門性を深める立場になく経験も積めない。このため、翻訳能力を持つべきなのは、管理者、開発者、または外部のいずれかということになる。

最も望ましいのは、管理者（情報システム部門）が翻訳能力を有することである。ユーザーと開発者の間に入る立場であり、また、開発ベンダーのコントロールにも大いに役立ったためである。ただし、習得には相応の時間が必要となる。情報システム部門内の人材を育成しようとするのであれば、担当するプロジェクトをある程度絞り、より深く要件に踏み込んで当事者としてシステム開発を経験させることが重要である。

それでは開発者が有しているのはどうか。実際に多くのシステム開発の現場ではそうなっているだろう。特に問題なく運用されているシステムでは、優秀な開発者がユーザーを意識して活動しているはずである。問題があるとすれば、新たな環境でのシステム開発において新たな開発ベンダーを選定する場合であろう。開発ベンダーといっても多種多様である。一般的に翻訳能力を備えている開発ベンダーは価格が高い。しかし、翻訳能力が高

いかどうかは実際にプロジェクトを進めてみないと分からない。また、価格の安い開発ベンダーに翻訳能力がないとも言いきれない。強いていえば、よく知られた開発ベンダーであれば人材も豊富で、それだけ翻訳能力を備えている確率は高い（相対的にコストも高い）ということであろうか。

最後に、外部からの調達についてである。外部調達は、差し迫った問題があるときには最も機動的な手段になり得る。当然、その分のコストは発生するが、開発失敗のリスクと比較してどう捉えるかである。例えば、開発ベンダーの人員が100人以上もいるようなプロジェクトでは、要件確認担当のユーザーとは別に、発注者側のPMO（プロジェクトマネジメントオフィス）として10名程度の人員は必要になるだろう。ユーザー側の事業部門もしくは情報システム部門でそれだけの人員がそろわない場合、不足を補うために外部調達を行う、というイメージである。

翻訳能力を有するコンサルタントなどは、往々にしてシステム開発の上流工程を専門とすることが多いため、できるだけ早い時期（システム化構想など）に参画させた方が得策だろう。開発ベンダーはすぐには変えられないが、コンサルタントは開発が軌道に乗れば役目終了とすることができる。野村総合研究所（NRI）もシステム開発の翻訳の専門家として、ユーザー合意の一助になることができれば幸いである。 ■