

# システムテストにユーザ企業はどうかかわるべきか

松岡 博英 : 株式会社クロスフィールド

## 1. はじめに

基幹システム導入など、戦略的 IT 投資案件は、企業にとってあらゆる面で大きなインパクトを及ぼすプロジェクトである。当然ながら失敗は許されず、計画当初の Q（品質）・C（費用）・D（納期）を達成すべくゴールを目指す。

とはいえ、通常の物品購入（何を、いくらで買うかが予め明確）と異なり、システム導入は工程を経てシステムの仕様が固まり、最終的にプログラムとして納品されるという特徴があり、ユーザ企業（＝発注者）は納品時に初めて“出来上がり（＝品質）”を認識することになる。金額の大小に関わらず、ウォーターフォール手法で開発を行う限り、この事実は変わらない。結果的に、納品されたプログラム品質が低く、稼働延期を余儀なくされるケースが未だ数多く見受けられるのもこのためだ。費用を含むリスクの一部は、システムベンダとの契約の中で回避することも可能だろうが、本稼働遅延による機会損失はその限りでなくユーザ企業の痛手は計り知れない。

筆者は、20 余年にわたりシステム導入プロジェクトに関わり、様々な経験を重ねてきたが、未だこの問題を完全に解決する術を見いだせていない。しかしながら、解決のために、まだ、ユーザ企業にできることがあるのではないかと考えている。

以下に記載する内容は、あくまでも筆者個人の私見であることをご理解いただいたうえで、ご一読いただければと思う。

## 2. ユーザ企業の責任とは

プログラムの品質向上に向けてユーザ企業ができること、ユーザ企業が果たすべき責任とは？ 改めて考えてみたいと思う。

### （1）開発ボリュームをおさえる

パッケージシステムをノンカスタマイズで利用するならいざ知らず（それでもマスタ設定不備によるエラーはそれなりに発生するが・・・）、基幹システムともなると、パッケージシステムを採用しても相当なボリュームのアドオン、カスタマイズが発生するのが常だ。この量が多ければ多いほど確実にプログラムの品質低下リスクが増加することは既知の事実であろう。

ならば、まずもってユーザ企業にできる最大の取組課題は開発ボリュームをおさえることになるだろう。パッケージであればアドオン、カスタマイズを極小化することに尽きる。だからこそ、事前に可能な限り精緻な RFP を作成し、より自社のニーズ（＝システム要件）に合致するパッケージを選定する。しかしながら、それでもなお要件定義、基本設計のフェーズを経て追加開発要件は発生する。

### （2）ドキュメント（＝設計文書）をレビューする

要件定義、基本設計フェーズでの決定事項は、“要件定義書”、“基本設計書”という形でドキュメント化され、ユーザ企業はこれらを確認することで最終的な開発範囲を確定し、システムベンダに対して以降の作業（詳細設計、開発）の GO サインを出すことになる。一般的には、基本設計（＝外部設計）が完了すれば画面や帳票レイアウトはほぼ固まるため、ユーザも確認作業を重ねることでシステムの実現イメージを想定

できるようになり、記述の誤りや修正を依頼することも可能だ。そもそも、要件定義、基本設計に関しては、相応の時間を割いてシステムベンダと協議を重ねることで相互理解を図ることができる。つまり、双方がやるべきことをきちんとやればコミュニケーションロスによる認識齟齬が生まれるリスクも最小化できる。

### (3) 詳細設計以降（請負工程）で何ができるかを考える

基本設計が完了すると、ユーザ企業は達成感や責任を果たした安堵感を覚える。ここから先は請負契約に基づきシステムベンダが“決められた時間（＝スケジュール）で”、“決められたもの（＝基本設計通りのシステム）”を作るフェーズであり、ユーザ企業は、数か月後のシステムベンダからの納品を待つのみである。新システム稼働に向けて様々な準備作業（マスタ整備、新業務フロー定義、受入テスト準備等々）を並行して行う必要があるものの、プログラム開発に直接関与することはない。プログラム品質に関する責任は、開発を担うシステムベンダが負うものであり、ユーザ企業にとっては不可侵な領域である。筆者も大筋、その通りだと思う。プログラム開発そのものに手を出ることができない以上、プログラム品質向上に向けてユーザ企業にできることには限界がある。しかし、だからこそ、できることは何でもやるべき、というのが筆者の持論である。

#### ・詳細設計工程においてもコミュニケーション機会を設ける

ほとんどのシステムベンダは、詳細設計以降、納品時まで自社作業が中心となり、ユーザ企業と顔を合わせる機会が激減する。一方、ユーザ企業も基本設計を終えて（ある意味、自分勝手に）システムの実現イメージを抱いているため、詳細なシステム仕様に対して鈍感になりがちだ。しかし、実際には、基本設計では詰め切れていない様々な仕様が存在する。いくつか例を挙げてみよう。

- ・各項目の初期表示値はなにか？
- ・入力時にカーソルはどう推移するか？
- ・エラーメッセージはどのように表示されるか？
- ・画面遷移はどうなる？ポップアップ画面はどう表示されるか？
- ・入力時のエラーチェックはどのように働くか？ 等々

実際にシステム導入に関わった方なら、「ああ、たしかにその検討が漏れていたな」という経験をお持ちではないだろうか。これらの積み重ねが、気づけば仕様確認漏れとして後工程で問題になる。問題が山積みになる前にきちんと解決出来る場（＝システムベンダとのコミュニケーション機会）を設けておくことが肝要だ。かつ、これはシステムベンダとの間で事前に合意しておくことが必要（システムベンダ側は詳細設計フェーズにおいて定期的なコミュニケーション機会を設けることを想定していないケースが多い）である。

#### ・開発工程において動作確認を行う

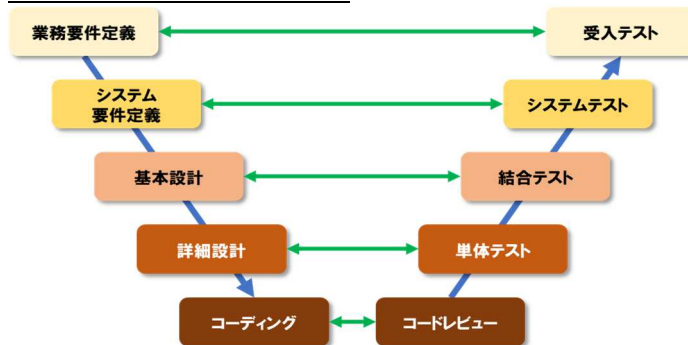
システムベンダに対して、プログラム開発途中に動作を確認したい、という要求をしてもなかなか応じてくれない可能性が高い。従って、これも事前にシステムベンダと合意しておくことが欠かせない。実際に動く画面を見ることで、（自分勝手な）システム動作イメージと現実の差異を認識できることに加え、上述したさまざまな詳細仕様の処理を体感することができる。認識のずれが発覚した時にプログラムの修正要求が叶うか否かは状況次第だろうが、少なくともここでの気づきを後続のプログラム開発に反映することは可能だ。

#### ・テストに深く関与する

システム開発工程とテスト工程の関係を表す際に用いられるV字モデルは広く知られ

た概念だが、通常、コードレビューからシステムテストまではシステムベンダが請負契約の中で実施することが多い。ユーザ企業はテスト結果報告書が納品された時点でテスト結果をレビューすることになる。

・システム開発におけるV字モデル



受入テスト（UAT）工程は、ユーザ企業がプログラムの検収と本番運用の可否を判断するために行う最後のテストであるが、ここで問題が発覚しても、正直なところ、取り得る選択肢はあまりない。睡眠時間を削ってひたすら改修とテストを繰り返し、祈りとともに本番運用に間に合わせるか、あるいは、本番運用開始時期を延期し、必要なレベルの前工程作業をやり直すか、、、いずれにしても幸せな結末を迎える可能性は低い。

ならば、ユーザ企業はどこからどこまでテストに口を出し、関与すべきか。できるならば、単体テスト以降のすべてのテストに関与することが望ましい。

単体テストでは、テスト結果確認も勿論有効だが、システムベンダがどのようなプロセスでどのようなレベルのテストを行うか、これを事前に確認しておくことが欠かせない。

結合テスト（内部結合・外部結合）においては、計画書/仕様書作成段階で基本設計書の記載内容を検証できるテストケースがしっかり盛り込まれているかをチェックし、不足があれば遠慮なくテストパターンの追加を要求する。

システムテストは業務視点でテストシナリオを作成し、発生しうるあらゆる業務について網羅的に検証することに加え、周辺システムとの連携テスト、基幹システムであれば、月次、年次処理等についても漏れなく検証が必要だ。また、サーバ復旧、データリカバリ等のインフラ周りのテストについてもこのタイミングで実施しておかなければならない。

ここまでやる必要があるのか？という意見もあるかもしれない。が、筆者はさらに踏み込んだ取組が必要ではないか、と考えている。

### 3. システムベンダの役割に立ち入る

ユーザ企業ができる限りテスト工程に深く関与することで、品質向上に好影響をもたらすことは間違いない。システムベンダにユーザ企業側の品質へのこだわり、本気度が伝わることで（良い意味で）プレッシャーを与えることにもなるだろう。しかしながら、これら一連のテストは、あくまでも“システムベンダの社内ルールに基づく社内テスト（＝開発者視点でのテスト）”であり、ユーザ企業はテスト仕様書と結果のし

ビューは出来たとしてもテスト作業そのものに関与できるわけではない。ならば、このテストを相応のスキルを有する第三者（≠開発を請け負ったシステムベンダ）に委託することでプログラム品質管理に深く関与することはできないものか？

実は、こうしたニーズに応える形で、第三者視点でのソフトウェアテストサービスを提供する会社はすでに存在している。彼らは、システムベンダに左右されない独自のソフトウェア品質管理アプローチに基づきシステムの規模や複雑性に応じたテスト定義を行うだけでなく、実際に、テスト計画の策定からテスト実行まで担うことをセールスポイントにしている。

第三者テストを実施した事例をご紹介します。

外部テストは要件定義フェーズからプロジェクトに参画し、要件定義書、基本設計書の内容を詳細に把握したうえで、独自のテストアプローチに則り、結合テスト以降のすべてのテスト計画を策定するとともにテスト作業を担当した。これによりシステムベンダは結合テスト以降のテスト作業から解放され、障害対応に特化したことで早い段階から改修作業にリソースを投入でき、無事スケジュール通りのシステム稼働を実現したのである。

外部テストの起用はプログラム品質に対して“疑問”を持ち、“第三者の視点で客観的にテストを行う”ことに意義があるのは勿論だが、一方で、深刻なリソース不足が慢性化しているシステムベンダにとっても、自前でテスト要員を調達する必要がなくなり、プログラム開発に特化できるという点では決して悪い話ではない。実際、上記のケースも外部テスト費用の一部については、システムベンダが負担しており、システムベンダと折り合いさえつけば、相互で費用負担することも可能と思われる。

本来であれば、請負契約の一部であるテスト工程においてユーザ企業が要員調達にまで踏み込む、ということは、従来の思考では考えられないことであったと思う。しかし、冒頭触れたようなプログラム品質が問題となって稼働が遅れるというトラブルを未だ数多く見るにつけ、本提案が決して“過剰な取り組み”であるとは言い切れないと考えている。

#### 4. まとめ

筆者がシステム導入プロジェクトの現場に身を置いて20余年。これまでもユーザ企業の立場からプロジェクト成功に力を注いできたが、プロジェクト途中でシステムベンダの開発力自体を改善することはほぼ不可能である。我々にできることは、開発工程をシステムベンダ任せにしないこと、さらに開発成果として出来上がったプログラムの品質強化に向けた取り組みにとことんこだわることではないだろうか。確かに、「システムベンダに支払う費用だけでも大変なのに、このうえさらにテスト費用までこちらで負担しなければならないのか!？」という悲鳴がユーザ企業からあがるであろうことは容易に想像できる。さらに、請負契約範囲の再定義など、システムベンダとの間で整理しなければならないことも少なくない。しかしながら、それでも、それだけの価値があると筆者は考えている。

これから、システム導入プロジェクトに取り組む予定がある方は、一度、真剣に検討してみてもいかがだろうか。