

# 情報システム開発の原点

南條 優

## The hinge of system development lifecycle

Masaru NANJO

### システム開発現場の混乱とその要因

納期の大幅な遅延、予算の大幅な超過、稼働直後の相次ぐトラブルなど、システム開発現場の混乱が多発している。これには様々な要因が考えられるが、その直接的な原因は要求定義の不備である。要求定義が不十分であれば、当然のことながらシステム仕様が確定しない。そこを曖昧にしたまま次の工程に入れば、これまた当然のことながら、相次ぐ仕様変更となり、予期せぬ工数の大幅増となる。しかも、その仕様変更がスムーズに行われればまだしも、途中で待ったが掛けられてシステム開発が中断したり、停滞したりするケースも多い。

その結果、システム稼働が大幅に遅れ、予算を大幅に超過するという深刻な事態になるが、仕様が確定しないまま、見切り発車をしてスケジュールを守ろうとすると、今度は稼働後に苦情やトラブルが頻発し、メンテナンス地獄に陥って、多数のメンテナンス要員をいつまでも残さざるを得なくなり、総費用はかえって増大する。

それでは、要求定義の不備はなぜ起きるかということ、そこには、上級SE不在という深刻な現実があり、システム開発環境の変化が

その背後にある。

私は過去に30年間に亘りシステム開発の現場にいて、プロジェクトリーダーの喜びも苦しみも幾度となく味わってきたが、私が現役を離れる前と後とでシステムの開発環境は大きく様変わりした。

システム開発環境の変化は次の三つの時代区分で見ることができる。

1. ユーザー企業のシステム部門が全工程を統括した、システム部門の時代
2. ユーザー企業のシステム部門と情報サービス産業が上流工程と下流工程を分担した、分業の時代
3. ユーザー企業がシステム開発を全面的に情報サービス産業に委託した、アウトソーシングの時代

昔は、大手のユーザー企業各社はそれぞれ独自のシステム開発部門を持ち、上流工程から下流工程まで一貫して開発作業を行ってきたが、やがて人材派遣が一般化し、上流工程はユーザー企業が担当し、下流工程は情報サービス産業各社に委託するという分業体制となった。更に、近年では、システム開発全体を外部委託するというアウトソーシングが主流になっている。

そこで問題になるのが、システム開発の上

流工程を担当する上級 SE (システムアナリスト、システムコンサルタント、システムプランナーなど) がいないという現実である。情報サービス産業各社が飛躍的に業績を伸ばしたのは分業の時代である。そのため、人材育成の主眼は下流工程を担当するプログラマに置かれており、上級 SE はほとんど育っていない。一方、ユーザー企業はリストラの一環としてシステム部門のスリム化を進め、派遣社員依存から全面委託に切り替えて、上級 SE の育成を止めてしまった。つまり、上級 SE が委託する側、される側の双方から消えてしまった。ここに問題の根源がある。

委託する側のユーザー企業は情報サービス産業側に上級 SE がいるものとして開発を委託し、委託される側の情報サービス産業はユーザー企業側に上級 SE がいるものとして開発を受託する。そのしわ寄せが要求定義に収束する。そういう図式である。

情報サービス産業側が自社のイメージアップに使ったシステムインテグレートという言葉も混乱を助長した。インテグレートという言葉は上流工程と下流工程を統合するという意味であり、上級 SE が存在して初めて成り立つ概念である。下流工程だけのシステムインテグレートなど、本来はありえないことである。

にもかかわらず、その矛盾がなぜ看過されてきたかという、そこに要求定義という言葉のマジックがあったからである。

### 要求定義という言葉の意味

ここでいう要求とはシステム要求のことであり、system requirement の訳である。本来は必要性とか必要条件と訳すべきであるが、それを要求と訳してしまったことが、悲劇のそもそもの発端である。それでも、システムに要求されることと解釈するならば、それでもよいのであるが、ユーザーが要求したこと

と、とんでもない誤解をしている SE があまりにも多い。要求定義書を客の注文書もしくは注文請け書だと思っている SE もいる。

システムエンジニアは医者に例えると分かりやすい。体調不良で医者に見てもらったとする。そこで医者から要求定義書の用紙を渡されて、どのように治療したいか詳しく記述しろと言われたらどうだろう。要求が間違っていた場合は、たとえ死んでも、要求した側の責任ですからねと言われればなお更である。

レストランに例えてもよい。レストランに行ったらシェフが来て、要求定義書の書式を示しながら、どんな食材をどのように料理して食べたいか詳しく記述しろと言われたらどうだろう。まずくても、高くても、要求が間違っていたら要求側の責任である。これではたまったものではない。

医者は診断や治療のプロであり、患者は素人である。シェフは料理のプロであり、客は素人である。同様に、システムエンジニアは情報化のプロであり、エンドユーザーは素人である。ここのところを勘違いしてはいけない。

ただし、ユーザー企業側に医者つまり上級 SE がいて、情報サービス産業側が看護婦などの医療スタッフを派遣している場合は、話は別である。医者がカルテを示し、医療スタッフが治療を担当する。この場合はカルテが要求定義書である。

ここで、ユーザーニーズという言葉を考えてみたい。ユーザーニーズはユーザー要求ではないし、ユーザー希望でもない。ユーザーにとって必要なことという意味である。ここを間違えるとたいへんなことになる。

たとえば、糖尿病の人に何が食べたいかと聞けば、甘いものとかお酒とかいうであろうし、高血圧の人に食べ物の希望を聞けば、塩辛いものを列挙するだろう。しかし、彼らの本当のニーズは糖分控え目であり、減塩であ

る。

過激な表現をすれば、患者の希望を何でも聞き入れる医者が出たとすれば、その医者患者は全員が死亡するだろうし、社員の希望をなんでも実現しようとする SE がいたら、その会社は間違いなく倒産する。

## システム開発における要求定義の位置付け

要求定義の重要性を理解するためには、システム開発のライフサイクルにおいて、要求定義がどこにどのように位置付けられているかを明確にしなければならない。そのためには、まず、システム開発のライフサイクルのどこで何をするかを確認しておく必要がある。システム開発は基本的には次の六つの工程から成っている。

- システム調査
- システム分析
- システム計画
- システム設計
- システム構築
- システム導入

各工程はフェーズと呼ばれ、日本語では局面と訳されているが、この言葉にはかなり重要な意味が含まれている。つまり、各工程は単なるステップではなく、それぞれに異なった状況に置かれているということである。

フェーズは相と訳すこともできるが、昆虫の成長過程に例えると理解しやすいと思う。蝶や蛾などの昆虫は卵から孵化して幼虫になり、蛹になり、脱皮して成虫になる。姿形はもちろんであるが、名前が変わるものもある。水中にいたオタマジャクシは蛙となって陸に上がり、同じく水中生物のヤゴはトンボになって空を飛ぶ。その間に、もちろん、環境は激変する。これがフェーズというものの正体である。

システム開発の全工程を安直に二分すると、前半が上流工程で後半が下流工程という

ことになり、前半を担当するのがシステムエンジニアもしくは上級 SE であり、後半を担当するのがプログラマもしくは初中級 SE であるという図式になるが、現実はそのような単純なものではない。

前半工程と後半工程の区切りは要求定義である。その要求定義はシステム設計の一部であり、ここまでが上級 SE の仕事である。

ここでは、一応、上級 SE という言葉を使ったが、システムアナリスト、システムコンサルタント、システムプランナー、アプリケーションスペシャリストなどの総称であり、明確な定義はされていないが、プログラマが兼務することは、現実問題としては無理である。

実は、ここが最大の問題点なのであるが、要求定義は調査、分析、計画と進めてきたシステム開発上流工程の総決算であり、上流工程がきちんに行われていれば、要求定義はそれほど難しい仕事でも手間の掛かる仕事でもなくなるが、上流工程に手抜きや大きな誤りがあれば、そのしわ寄せがすべて要求定義に集まり、システム開発が難航する最大の原因になる。

## システム開発ライフサイクルの確認

前述したようにシステム開発には、システム調査、システム分析、システム計画、システム設計、システム構築、システム導入の六つのフェーズがあるが、各フェーズで何が行われるかを整理しておこう。

### システム調査

システム開発の最初はシステム調査であるが、これは情報収集のフェーズである。情報システムの成否の鍵は、それを担当する上級 SE が握っているが、良い SE と悪い SE の差はどこにあるかということ、豊富な予備知識を持っているかいないかである。

それではその予備知識をどこで手に入れるかということ、基本的にはSE 本人の過去の経験であるが、そこにはないものはシステム調査で補わなくてはならない。

システム調査では次の三つの作業を行う。

- 現状調査
- 動向調査
- 事例研究

情報には組織の内部で発生する内部情報と、外で発生する外部情報とがあるが、前者に対する調査を現状調査といい、後者に対する調査を動向調査という。

現状調査は現行のアプリケーションに関する調査であるが、現実をありのままに見るクールな目が必要である。もっと端的に言えば、タテマエに惑わされずにホンネの部分を知ることである。

動向調査ではシステムを取り巻く環境に目を転じ、世の中の動きや時代の流れを知る。もう少し具体的に言えば、情報関連の技術動向、自社商品の市場動向や顧客性向、ライバルの動き、業界事情、海外事情、法律や行政の動きなどである。

事例研究は一口で言えばお手本探しである。物を作る場合は何でもそうであるが、お手本があるかないかでは大違いである。ここでは同業他社の成功事例はもちろんであるが、新技術をいち早く取り入れた先進事例、ユニークな海外事例などを幅広く研究して、よいところ取りをする。

## システム分析

システム分析はシステム開発の六つのフェーズの中では一番理解されにくいフェーズであるが、一口で言えば、目的明確化のフェーズである。ここでは、今何が必要なのか、そのためには何をすればよいのか、それをすると本当に得なのか、何か重大な危険をはらんでいないかなどを次の四つの作業で明らかにする。

要求分析

構想計画

フィジビリティスタディ

費用効果分析

要求分析の要求とは、システムに要求されることという意味であって、利用部門の誰かが要求したことという意味ではない。

ニーズという言葉に置き換えた方がよいかもしれないが、ニーズには顕在ニーズと潜在ニーズがあり、この場合は潜在ニーズの方がはるかに重要である。

顕在ニーズは、本人が自覚しているニーズなので、システム調査で把握することも可能であるが、潜在ニーズは、本人が気付いていないニーズなので、分析をしてみないと分からない。

構想計画では選択可能ないくつかの案、つまり、代替案を立案し、比較検討して一番よいものを選ぶ。現行システムをそのまま継続することも一つの代替案である。

ここで一番重要なことは、ブレークスルー、つまり、現状打破の糸口をつかむということである。やはり、何か突破口がないと革新的なシステムは生まれない。その一つは斬新なアイデアであり、もう一つはIT(情報技術)であるが、そのヒントはシステム調査で得られているはずである。

フィジビリティスタディは可能性の検討と訳されているが、ここでは主にマイナス要因、つまり、成功を阻害する要因を調査し、実現性を検証する。環境などへの影響評価とリスク分析が主な内容である。

費用効果分析では費用を算定し、効果を予測して、採算性を検証する。

システム分析というと、何か難しい名前の技法を駆使してユーザーを煙に巻くという印象が強いが、ユーザーが理解できないような技法は使わない方がよい。

## システム計画

システム計画はその名の通り計画立案のフェーズである。システム開発は膨大な費用を伴うので、担当者レベルでは判断できず、全社的な意思決定を必要とする。そのためのフェーズがシステム計画であり、このフェーズの成果物はシステム提案書である。

ここでは次の三つの作業を行う。

基本設計

実行計画

システム提案

基本設計は広い意味ではシステム設計の一部であるが、システムの基本仕様が決まらなないと実行計画の作成は無理なので、システム設計フェーズに先行して、このフェーズの最初に行う。

基本設計の基本をアウトライン（輪郭）だと思っている人がいるが、そうではない。たとえば、人の顔を描く場合、初心者は顔の輪郭を先に描くが、これでは、その人物像はまるで見えてこない。プロなら最初に眼を描き、次に鼻を描く。これが、「目鼻を付ける」の語源である。写真のポートレートも同様で、目にピントが合っていないければ、その人物の魅力は伝わらない。

基本設計で最も重要なことは、そのシステムのセールスポイントを明確にすることである。そこから方向性が見えてこなければいけない。そのためにはシステム開発の目的、システムの効用、システムの特徴、他システムとの差別化などが明確でなければならず、それがあってのアウトラインである。

実行計画は人、もの、お金、時間に分けて考えるとよい。

人計画は基本的には要員計画であるが、ここでは導入推進組織、開発工数見積り、要員調達手段などを立案する。

もの計画は基本的には設備計画であるが、情報機器の機種選定や付帯設備工事、ソフトウェアの選定などを行う。

お金計画は基本的には費用計画であるが、予算編成の前提となる費用見積りのほか、費用対効果の対比を行って採算性を検証する。

時間計画とは、要するに日程計画のことであるが、ここでは、システム開発スケジュールとシステム導入（もしくは移行）スケジュールの双方を立案する。

システム提案は計画全般に関して経営トップやエンドユーザー（利用部門）の承認つまりゴーサインを得るための手続きである。ここをパスしなければシステム設計以降のフェーズには入れない。

まず、基本設計と実行計画に基づいてシステム提案書の作成を行い、関係各部門への根回しの後で、プレゼンテーション（提案説明）を行う。最後は承認手続きであるが、委員会審議、役員会審議、稟議書など、いくつかの関門がある。

## システム設計

システム設計はシステム開発の四番目のフェーズであるが、システム部門にとってはここから本番であり、前の三つのフェーズはその準備段階であるとも見られる。システム開発を情報サービス産業にアウトソーシングする場合は、ここから先を外注することが多い。しかし、これが悲劇の発端なのであるが、ここまでは前半でここから後半という分け方には非常に大きな落とし穴がある。実はこのフェーズの中に、システム開発の前半と後半を分ける非常に大きな分水嶺がある。それが要求定義である。

システム設計は次の三つの作業から成り立っている。

要求定義

外部設計

内部設計

要求定義はシステム開発の前提となるシステム要件を明確にすることであり、本来は要件定義とすべきであるが、要求定義という



言葉が一般化しており、ここでは慣例に従った。

システム要件を厳密に定義するならば、どのような資源（情報や技法など）を用いて、どのような制約条件（時間や予算や法律など）の下で、どのような目的を達成するためには、どのような機能や性能や操作性が必要かということである。要するにシステムの必要条件のことであるが、その、system requirement という英語を誰かがシステム要求と訳してしまったことから、システム開発現場での混乱が始まったと私は思っている。

システム要求とはシステムが要求していることという意味であって、断じて、ユーザーから要求されたことという意味ではない。比較的意味の近い言葉にニーズがあるが、これとても誤解を生じやすい。もう少し言葉を絞って潜在ニーズとすれば、ほぼそれに近い。

問題は、その潜在ニーズをどうやって把握するかということであるが、アンケートを取るとか、インタビューをするだけでは無理である。つまり、調査だけでは駄目で、医者ならば診断、SE ならば分析が必要になる。したがって、システム分析というフェーズを省いてシステム設計に入れば、必ず、この問題が露呈する。

要求定義が固まれば次は外部設計である。情報システムの最大の特徴は、それがマンマシンシステムつまり人間機械系であることであるが、そのため、人間側と機械側という二つの接合面を持つ。前者がユーザーインターフェースであり、後者がマシンインターフェースである。それをコンピュータから見れば、ユーザーインターフェースは外側なので、その部分の設計を外部設計といい、マシンインターフェースは内側なので、その部分の設計を内部設計という。要するに、ユーザーから見える範囲を設計することが外部設計であり、ユーザーから見えない部分、つま

り、ハードウェアや基本ソフトに左右される部分の設計が内部設計である。

## システム構築

システム構築は工業製品で言えば製造工程であり、工数としては六つのフェーズの中でここが最大である。にもかかわらず、フェーズの呼称が確立していない。製造工程なのでシステム製造と呼びたい気もするが、工業製品ではなく知的財産なので、制作のほうが正しいかもしれない。ここでは、多く使われている方をとるという原則に従い、システム構築とした。

改めて説明することもないが、ここには次の三つの工程がある。

プログラミング

システムテスト

ドキュメンテーション

プログラミングはコーディングや結合編集、単体テストなどから成るが、システム開発ツールを開発するとか、作業を標準化するなどの作業も全体の効率を大きく左右する。

システムテストは完成検査のことであるが、結合テスト、シリアルランテスト、ストレステスト（過負荷テスト）、ライブテスト（総合テスト）、非常時テストなどから構成される。ストレステストはデータやアクセスが短期に集中した場合のテスト、ライブテストはシステムに詳しくない一般ユーザーを参加させてのテスト、非常時テストは防災や防犯の対策を確認するテストである。

ドキュメンテーションは文書化と訳されているが、ここでは、システム運用のためのシステムドキュメント、プログラムメンテナンスのためのプログラムドキュメント、オペレーションのためのマニュアルなどを整備する。

## システム導入

システム導入はシステム開発の最終段階で

ある。多くの解説書はシステム構築をシステム開発の最終章としているが、SEの仕事はそれで終りではない。新しいシステムが稼動し、初期の目標が達成されたことが確認されるまでは開発担当SEの仕事である。

システム導入は次の三つの工程から成っている。

- 導入準備
- システム移行
- フォローアップ

アプリケーションプログラムが用意されても、もちろん、それだけでは導入の準備万端が整ったとはいえない。やるべきことは他にもたくさんある。それらを一つ一つ固めていくことが導入準備である。

たとえば、初期データ入力、データコード設定、コンピュータの搬入・据付・調整、通信接続、用品・備品調達、実施訓練、防災・防犯対策などである。

システム移行とは現行システムから新システムへの業務の引継ぎであるが、ここではデータ移行や帳票の切替え、スケジュール調整などを行う。

何事も始めが肝心である。システム稼動直後は、担当者が不慣れなこともあり、予期せぬトラブルが生じやすい。そこを上手に乗り切らなければ画竜点睛を欠くことになる。

ここでの主な作業は初期指導、初期トラブル対応、マニュアル整備、システム性能評価である。

## 提案型 SE とは何か

かつて、提案型 SE という言葉が流行したことがある。SEはユーザーの言うとおりに何でもやりますという御用聞き SE では駄目で、自ら改善案や解決策を提案すべきだという指摘である。まさに其の通りであるが、そう簡単にできることではない。無知な提案や無責任な提案は百害あって一利なしである。

昔の自分を振り返ってみても、私は利用部門に対して、絶えず、様々な提案をし続けてきたし、それなりの成果も挙げてきたが、昨今の SE と比較すると、かなりのアドバンテージがあった。まず、そもそも自分の会社のことなので、トップの経営方針や会社が置かれている環境は分かっているし、業界知識や商品知識もある。それに加えて、システム調査やシステム分析も継続的に行っていたので、相当な予備知識を持っていた。その予備知識を情報化に結びつければ、エンドユーザーよりも広い目でシステムを見ることができ、その分、優位にたてる。

それに対して、アプリケーションに関する予備知識を何も持たない業界 SE が、コンピュータ関連知識だけを振りかざして、改善案や解決策を提案しようとしても、それは無理である。ましてや、上級 SE としての経験の浅い SE にそれを求めるのは酷というものであろう。

それでは業界側としてはどうすればよいか。上流工程を下流工程から切り離して請負い、きちんとしたシステム調査、システム分析、システム計画を実施し、それが承認されてから、下流工程を請け負う。もちろん、これが正攻法であり、そうすべしというのが正論であるが、上級 SE がユーザー企業と情報サービス産業の双方から絶滅危惧種に指定されている現状では空しい意見である。

そこで、その代替案としての私の提案はこうである。情報サービス産業各社は上級 SE の残存部隊や準上級 SE を結集して、まず、自分の得意な分野のビジネスモデルのプロトタイプを作る。プロトタイプという言葉は最近ではちょっとした流行語になっており、試行錯誤の代名詞として使われているが、プログラムレベルのプロトタイプではなく、システム設計それも基本設計や外部設計レベルのプロトタイプである。そこには先進企業の成功事例が色濃く反映されていなければならな

い。

それをユーザーにきちんと示して要求定義を行う。各社の個性の部分は要求定義で補う。最初のビジネスモデルがしっかりしていれば、要求定義をきちんと行うことで上流工

程の代替になる。もちろん、次善の策ではあるが、現状よりはましであり、システム開発現場の混乱を大幅に減らすことにはなると思う。