

# 守りながら攻める —CIO と「プロジェクトマネジメント」の現在



**大賀憲 株式会社シグマックス P2(Program & Project) シェルパ マネージングディレクター**  
 専門商社、外資系コンサルティングファームを経て現在に至る。製造業・流通業、アミューズメント、総合商社などへのコンサルティングに従事。グループ経営、経営管理、リスク管理などの策定から業務改革、大規模システムのインプリメンテーションまで、一貫通貫のサービス提供を得意とする。現在はプログラム&プロジェクトマネジメントのサービスを責任者としてリードしている

ビジネス環境の変化やデジタル化に伴い、テクノロジーをエンジンに多くの企業が変革を迫られている。しかし全社横断的な取り組みとなるプロジェクトや、その集合体であるプログラムが増える一方、事業部門とIT部門の連携がうまく進まない日本企業も多い。

また、高いベンダー依存度、縦割りの組織や制度、リーダー人材(人財)の不足やスキルの空洞化といった問題を抱えながら、IT部門は、クラウドやモバイル、ソーシャルなど“第3のプラットフォーム”と言われる新たなテクノロジーを活用し、経

営や売りに貢献する難易度の高いプロジェクトに取り組みなければならない。

守りながら攻める。複雑化するプロジェクトマネジメントの要点を抑えるために必要なことは何か。連載第1回となる今回は数々の大規模プロジェクトのマネジメントに関わってきたシグマックスのマネージングディレクターである大賀憲氏にその基本的な概念や実現のために必要なことを聞いた。

——今、企業における「プロジェクトマネジメント」のニーズが高まっているとのことだが、その背景には何があるのか。

かつてのシステムには、「あるべき姿」というものがありました。1990年代後半から2000年代前半、日本では「グローバル化」というキーワードのもと、企業がこぞって全体最適や業務改革のためにBPR(ビジネスプロセス・リエンジニアリング)に取り組み、統合基幹業務システム(ERP)導入を始め

た時期がありましたが、これもひとつの「ベストプラクティス」であったといえます。

振り返ってみれば、2000年問題もあって猛烈な勢いで導入されたERPも、現行業務にあわせたカスタマイズを重ねていった結果、狙った効果を半分も享受できない日本企業が多かったというのが実態ではありますが。

「プロジェクトマネジメント」という意味では当時から、システム導入を発注したベンダー側のプロジェクトマネージャー(プロマネ)の力量に依存してプロジェクトが進行し、それによって成功不成功が決まるという状態がほとんどだったように思います。本来は、企業側にリーダーシップをとれる人財がきちんとプロマネとしてアサインメントされ、ベンダー側のプロマネとともにタッグを組んで成功に導く、というのが理想形ですが、経験者が企業にほとんど存在しないので現実的ではありませんでした。

しかし、それから10年たった今でも、一般的に縦割りで動く日本企業では、トップマネジメント肝入りの全社プロジェクトが展開される場合でさえ、エース級人財がプロジェクトに集められない、というケースも多い。当時も今も最初から戦力的に厳しいのは変わらないのです。困難を伴うプロジェクトに取り組んでも、結果的に期待したような成果は出ないし経営にも事業にもそれほど感謝されない、という状況は続いています。

システム開発プロジェクト内容そのものも色合いが変わっています。これまでは事業の課題を解決するために、企業の情報システム部門が提案依頼書(Request For Proposal:RFP)と予算を提示し、コンペティションを競り勝ったITベンダーやコンサルティング会社に発注。業務とシステムを設計し、それに順じて開発を進める、という流れでした。時間軸も2年から3年、長いものでは5年くらいかかりました。

一方、今は急速なデジタル化の進行に伴って、企業の価値創造に貢献するようなプロジェクトも求められています。テクノロジーが多様化、複雑化する一方で、市場の変化が速いので、プロジェクトの成果実現までのスピードも圧倒的な速さが必要です。しかし今IT部門はスキルの空洞化に苦しみ、ベンダー自身も従来型のITスキルからの脱却が難航しているのが現状です。

プロジェクトに求められるものが大きく変わっている今、関わる人や組織がマインドセットを変え、能力と視座を上げていくこと、ITと経営と事業部門が一体となり、システム開発のプロジェクトに臨む必要があります。

## ——ITと経営と事業部門の一体化が求められる背景とは。

グローバル化とデジタル化という2つの潮流があると思います。「グローバル化」という言葉が現在も経営戦略の話題に上りますが、これは何も日本以外の国も市場として見る、というだけの話ではありません。「デジタル化」が同時に進むことで、異業種も交えた戦いが始まっているという点がポイントです。

たとえば、かつてはA食品会社のライバルはB食品会社でしたが、今はプラットフォームを握るGoogleやAmazonの動きも視野に入れる必要があります。当然、海外進出においてもその市場性は国ごとにも違い、中国に至ってその違いは都市単位です。国内外、異業種の競合入り乱れる中で、未知への手がかりをつかんでいくという活動は、かつての「ベストプラクティスを導入すれば競争には負けない」という世界とは全く異なります。

「デジタル化」という意味では、テクノロジーの進化でクラウドやモバイルが台頭し、さらにセンサの発展によりビジネスにおけるデータの持つ価値が急速に高まっています。多様化も進んでいますから、アセンブリ(組み立て作業)は避けられません。スマホだけでもApple、Androidがあり、それぞれセキュリティの認証の仕方も日々進化していきます。

マルチテクノロジーを駆使して未知の市場で戦っていくためには、事業側とIT部門側が一体となって、自社のビジネスを変えていくことにスピード感を持って取り組まなければなりません。環境変化もテクノロジーの進化も10年前とは比べものにならないスピードで進む今、悠長に3年もかけてシステムを導入し、終わったらまた事業から次の課題を聞いてプロジェクトを起こす、というような感覚で物事を進められる時代ではないのです。

## ——企業の体制や制度も大きく影響すると思うが。

そうですね。効率性を重視し、縦割り型組織で動いている日本企業では、プロジェクトという単位で物事を組み立て、

評価していくという考え方を、組織・制度の設計思想に入れていません。予算も会計年度および四半期単位でコントロールされ、そこでパフォーマンスを評価します。そもそもプロジェクトとは、一定の期間において、ゴールに対してどれだけの投資をし、どれだけリターンがあったのか、という形で評価するものですが、そのように、組織や制度がそもそも作られていないのです。

たとえば新しい事業を立ちあげて新市場に乗り込むためのIT投資をしようとする時も、予算はIT予算として計上され、IT部門の活動の結果として投資対効果を評価するのが普通です。IT部門はプロフィットセンターではないので、結果として、この予算は投資ではなく「コスト」として管理されることになる。

価値を生み出して「儲けるために何をどうするべきか」という視点で行っているはずの活動なのに、「いかにコストを抑えるべきか」という点に焦点が集まるわけです。しかし本来のプロジェクトにはこの考え方はなじまないし、これでうまくいくわけがありません。

### ——プロジェクトの進め方そのものが変わると。

基幹系の大型システムの刷新、といったものであれば、やはり第2プラットフォームのテクノロジーを活用してウォーターフォール型で着実に、でも市場にあわせて成果実現のスピ

ードは限りなく上げる、というアプローチが必要になるでしょう。

一方、新規事業を立ち上げる、ビジネスモデルを変えるとといった価値創造型のプロジェクトでは、第3プラットフォームの新しいテクノロジーを活用した未知の世界でのチャレンジになりますから、ちょっとやってみて「違うな」と思ったら修正し、完成系に近づけるというサイクルを高速で回していく、アジャイル型も確実に増えています。しかし、いずれも、全社的取り組みとして事業部門とIT部門が一体となった取組みなしでは実現できないことは変わりません。

### ——そういった新しい取り組みをする企業は増えているか。

増えていますし、もっと増えなければならないと思います。アジャイル型プロジェクトは、基本的にとがったテクノロジーを持っているベンチャー企業など他の企業とコラボレーションしていくやり方です。やりたいことを簡単にドキュメント化してまずは作ってみる。それを動かしながら調整し、リリースしてみても顧客の反応を見てまた修正、というスピード感です。こちらはリーダーシップやビジョンが極めて大事な世界ですね。

一方ウォーターフォール型基幹系プロジェクトの進め方も変化が求められています。実際、ここは今当社としても売り上げ規模が一番大きい部分であり、言い換えれば企業からのニーズが最も大きい領域です。プロジェクトのタイプに応じてプロジェクトマネジメントの特性は異なります。

システムプロジェクト 2つのタイプ

	2 <sup>nd</sup> プラットフォーム 基幹系等	3 <sup>rd</sup> プラットフォーム モバイル・ソーシャル等
プロジェクトの狙い/評価軸	QCD(品質 価格 納期)	Value (価値創造)
バイモーダルIT*1 におけるモード	モード1 線形での確実な遂行	モード2 非線形な不確実への俊敏な対応
	効率性、安全性、正確性 重視	実現までのスピード重視
アプリケーションタイプ	SOR*2	SOE*3
方法論	ウォーターフォール	アジャイル
アプリケーションデリバリー	ITL	DevOps
プラットフォーム	オンプレミス	クラウド
リリース頻度	長い	短い

\*1:バイモーダルIT :Gartnerによる定義

\*2:Systems of Record :ビジネスで起きる事象を蓄積。正確性が重要。データ中心

\*3:Systems of Engagement :ビジネス上での関係性を強化するためのシステム。即時性。利用者中心

## ——ウォーターフォール型のプロジェクトマネジメント支援では、どこが評価されるのか。

われわれが最も大事にしているのはプロジェクトの上流部分をきっちりと抑えることです。これまでのプロジェクトでは多くの場合、IT ベンダーが要件定義という名前のもとに「お客様がやりたいことを決めましょう」というフェーズを設けて、言葉を選ばずにいうなら「おおざっぱに」要件を決めてしまう。そうすると極端な話、現行をあまり変えたくない人の意見や、現場で声の大きい人たちの要望をざっくり集めて、ホッチキスで留めたような要件定義になってしまうわけです。

大型プロジェクトの大半が「成功裡」と言えない状態で終了する、あるいは経営に評価されずに終わる理由のほとんどはここにある、とわれわれはかねてから考えています。よって、とにかく定めたゴールを達成して成果をきっちり出すことがプロジェクトの目的ですから、「Why?(目的)」を明確にすることを最初の焦点にします。

具体的には、「事業部門側の役員」、すなわちビジネスオーナーと、プロジェクトが取り組むべきことについてしっかりと話し合い、オーダーを明確にしてから、「What?(プロセス)」「How?(手段)」の設計に落とし込みます。ここでのポイントは、最高情報責任者(CIO)やIT部門、事業各部門との合議ではなく、ひとつひとつビジネスオーナーと合意形成を図ることです。

## ——各論に落とししていく時のポイントは。

システムの仕様に落とし込む段階では、今度はCIO側と並んで「実現手段は全社最適の目線で決める」というアプローチを取っています。自ずと「ベンダー1社だけではシステム構築不可能」という判断になる。結果的に、マルチベンダー、マルチソリューションを組み合わせることになります。われわれは製品やサービスを顧客目線に立って選定し、それらベンダーを束ねる顧客とワンチームで、最後の成果が出るまでプロジェクトマネジメントを推進していきます。

## ——発注者側に立つということか。

そういうことです。従来のいわゆる「うまくいかない」プロジェクトでトラブルが発覚し始めるのは開発のテストあたりから。予算が固定された中で開発が上手くいかなくなってくると、基本的に馬力がかかる設計開発の予算とワークロードを削っていく。予算内に収めることがゴールになってしまい、そもそもその目的を見失ったまま終わってしまう、そんなプロジェクトにならないように、最初から最後までを一気通貫でお客様と共に設計し、動かしていきます。

別のベンダーを使ってスタートした大型プロジェクトが途中でとん挫してしまった顧客から、「なんとかしてほしい」という要請を受けて仕切り直しに入る、というケースもありました。

## ——そうした変化の中で、プロジェクトを評価し、方針を決めるCIOは、これからどのような役割を果たせばよいか。

事業基盤を支える第2プラットフォーム型ITと、価値創造をけん引する第3プラットフォーム型IT。この両方を両手でコントロールしなければならない、というのがこれまでの役割と大きな違いでしょう。当面、企業のITはどちらかだけの世界にはならないわけですから、管理する領域とリーダーシップを発揮する領域を使い分け、明確なメッセージを社内に発信していくことが大事です。

日本企業を眺めてみると、デジタル化が企業の競争力に直結しているにも関わらず、多くの企業でCIOが最高執行責任者(COO)よりも一段低いポジションで自らよとして見えます。デジタル化の重要性を痛感している企業は、最高経営責任者(CEO)、COO、CIOが三位一体で動いていますね。そういう意味では、CEOはもちろんのこと、CIO自身も自ら意識を変えていく必要があります。

デジタルで事業を変える取組みを、事業部門とIT部門がそれぞれオーナーシップをもって手を取り進めていくなれば、両者のリーダーはそもそも同じレベルにいるべきですし、そういう主体性をCIO自身もって組織に働きかけることが経営にとってとても重要だと思います。

## ——そのために必要なことは。

経営戦略や事業戦略に関して持論やビジョンを持っていることです。マーケティングやサプライチェーンも含めて、10年後の自社の姿をテクノロジーの側からイメージできていること。未来を見据えてビジョンを持つためには、社外とのコミュニケーションも重要です。同業種のCIOの交歓会などに閉じることなく、他業種やITベンチャーの人達とも交流する。もはやテクノロジーの世界は「大企業だから偉い」という理屈は通らないわけで、いかに多様な情報を持っているかに勝敗がか

かっています。

トライ&エラー型の思考回路も、これからのCIOにとってはなくてはならないものです。アジャイルで開発をするということは、人の評価やプロジェクトの動かし方、指示の出し方、上手くいってないときのサポートの仕方も含めて、ウォーターフォール型とは発想を変えるということです。減点主義を捨てて加点主義で、コラボレーションの発想をもち、メンバーの動きを促す。ビジョンやリーダーシップが必要です。

これからのプロジェクト運営

	2 <sup>nd</sup> プラットフォーム 基幹系・複雑型	3 <sup>rd</sup> プラットフォーム 環境変化対応型
プログラム・プロジェクトの性質	定量的なゴール 明確なスコープ 確実な推進 明確なROI	変革の実現、曖昧 変動するROI 爆発的なReturnを狙う スピード重視
体制、運営指針	ピラミッド・階層化(軍隊式) 明確な役割分担	ネットワーク型 コラボレーション
オーナーの役割	マネジメント	リーダーシップ ビジョン
PMOの主要な役割	着実な遂行に向けたマネジメント (マルチ)ベンダーマネジメント	変革実現に向けたリーダーシップ ステークホルダーマネジメント 環境変化、トライアル結果に基づく、 軌道修正

——1人で全部請け負うのは現状難しいような気がするが。

CIOとCTOを分けている会社があるように、管理する人とリーダーシップを発揮する人で分担する方法はあるかもしれませんが、いずれにしてもシステムは全部つながる必要があります。コミュニケーションと目線の一致を、より意識する必要がありますね。

いずれにしても、情報システム部の社内業務を積み上げて部長になった人が、突然CIOという名前が変わったときに、これまでの延長線上でこれらの役割が全て果たせるようになるかといったら、それは難しいでしょう。次世代リーダーの

育成も含めて、企業が今からでも手を打たなければいけない領域だと思います。逆に次の世代の人たちにとってはチャンスです。

つまり、第2プラットフォームを過去のキャリアでがっちり経験し、今、第3プラットフォームでプロトタイピングをガンガン回し始めている人たちです。そこで成功していけば、若くてもCIOになるチャンスがあると思います。逆に、変わらないCIOは、その場を去らなければならない時がくるでしょう。

※この論文はZDNet Japan(2015年12月~2016年10月)に掲載されたコラムの再掲載となります。



# プロジェクトを「丸投げ」できない理由(前編)

## ——SIer に頼りすぎるリスク



執筆者: 小室貴史 株式会社シグマクシス P2(Program & Project) シェルパ ディレクター

外資系コンサルティングファームを経て2012年シグマクシスに入社。経営管理、管理会計分野を中心とした戦略構想立案、プロセス変革、組織変革のコンサルティングに強みを持つ。流通業、製造業、商社、SIベンダー等に対するパッケージシステム導入のプロジェクトマネージャー、大型システムプロジェクトのPMOリーダーの経験を多数有する。

連載第1回ではインタビューの形をとりながら、これからの最高情報責任者(CIO)や情報システム部門は、基幹系システムの改善とコスト削減といった従来の仕事をこなしながら、新しいビジネスの創出のためにデジタル化やクラウドやモバイル、ソーシャルなど“第3のプラットフォーム”テクノロジーの活用といったテーマに取り組まなければならないと述べた。グローバル化、デジタル化により、経営や事業から求められる要求スピードはかつてないほど高まり、守備範囲は圧倒的に広がっている。

基幹系システムの対応においては、一時期の統合基幹業務システム(ERP)導入ブームに類する大きなブームはない。一方、グローバル標準化による効率化・コスト削減、サービス指向アーキテクチャ(SOA)やAPI連携技術などによるアプ

リケーション間の機能連携、リアルタイム性向上、内部統制・セキュリティ強化など、新たな技術へのキャッチアップが継続的に必要だ。

一方で、市場の変化に対応した事業運営を実現するために、クラウドやモバイル、データを駆使した柔軟なシステムを構築・運用する必要がある。結果として、さまざまなプロジェクトが並行して立ち上がって常に動いているが、それらは果たして期待した成果を生んでいるのか。企業の多くは同じ悩みを抱えている。

今回は、企業がそんな環境下で取り組む「プロジェクト」の実態を、ユーザー企業とSIベンダーの関係を通して、改めて見直してみる。

### 人材不足はITベンダーにとっても大きな悩み

2000年頃の日本企業の間では、バブル崩壊後の全社コスト削減、PL経営からBS経営へのシフトの中で、情報システム部門のスリム化や子会社化、情報システム子会社へのア

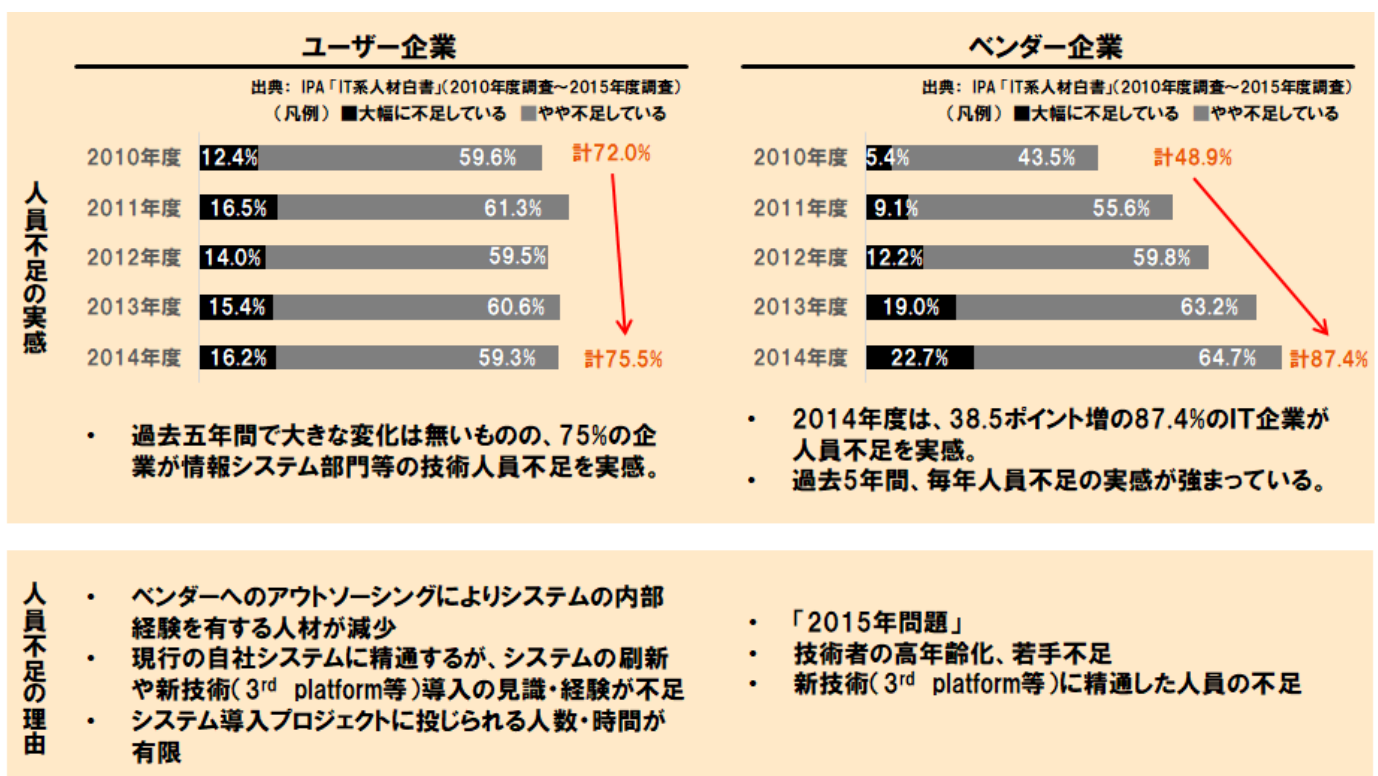
ウトソーシングの切り替えなどが流行した。この過程で、人員削減が進むとともに、業務プロセスに関する知識やプロジェクトマネジメントのほか、技術的な能力は子会社に集約され、本社に残っていた一部人材(人財)も高齢化により人数が減り続けている。

程度であったものが、2014 年には 90%近くまで急増している。背景には、「2015 年問題」に代表される技術者の高齢化、新卒、第二新卒といった若手に不人気業界としてみられた結果、新規の採用が出来ないこと、現状の人財の新しいテクノロジーへのスキル転換の遅れなどがある。

自社も頼みの綱の IT ベンダーも人財と知識の空洞化を起しているという現状に、今日本企業は置かれている。

情報処理推進機構 (IPA) がまとめている「IT 人材白書」によると、2010 年から 2015 年の調査の平均で 70%を超える企業が「IT 系人財が不足している」と回答している。日本企業は、この過程の中で、完全に IT ベンダーやシステム子会社への自社システムの設計や開発、運用に依存する体質へと見事に変わっていったのだ。

しかし、IT ベンダー側も状況は厳しい。同白書で「人財不足を認識している」と回答した比率は、2010 年には約 50%

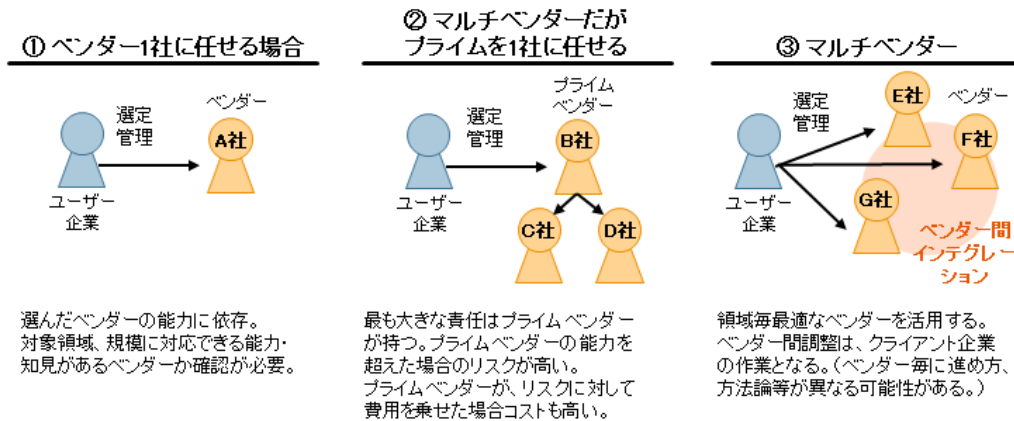


## プロジェクト管理は誰がやるのか

さて、そんな中でユーザー企業は各種システム構築プロジェクトを動かすことになるわけだが、プロジェクトの組み方は一般的に3つある。

- パターン(1): 1社のSIベンダーにすべて任せる
- パターン(2): 1社のプライムベンダーに複数SIベンダーの管理を任せる
- パターン(3): 複数SIベンダーを並列に組み合わせ自社が管理す





一見すると、パターン(1)、(2)は、自らが対応する相手が1社という意味でパターン(3)に比較するとリスクが低いように見えるが、SIベンダーも人財不足に苦しんでいる、という前提を置くと、実態は大きく変わる。

パターン(1)は、いわゆる「丸投げ」だ。管理責任はユーザー企業にあるものの、実際には要件を決めたらあとは任せる、というのが実態だ。しかし本来は、SIベンダーがプロジェクトの全ての領域に適切な人財を配置し、適切に動かしているか、ユーザー企業は自ら確認する必要がある。テクノロジーが多様化している今、「プロジェクトに必要な人財を1社で全部そろえられるようなSIベンダーはいない」という現実は、理解しておく必要がある。

パターン(2)の場合も、ユーザー企業が対面するのは1社だが、プライムベンダーの関係会社あるいは長年の取引先がその下につくケースが多い。要は、プライムベンダーが最も扱いやすい会社でプロジェクトが構成される。言い換えれば、本当にプロジェクトにとって適切なスキルをもった会社が配置されているかどうかは、ユーザー企業にはわからないというリスクを抱えることになる。

加えて、プライムベンダーは外部リソースを下に抱える分、責任負担とリスク負担が増えるため、価格やコンティンジェンシープラン(緊急時対応計画)などの値を高めに設定するのが普通だ。丸投げしたプライムベンダー以下がブラックボックス化した瞬間、プロジェクトコストも、リスクも高い危険なプロジェクトということになりかねない。

結局、どんなに自社がスキル不足であっても、プロジェクトを動かす以上、その品質と成果をコントロールすることから、ユーザー企業は逃れることはできないと考えた方がよい。

## システム構築プロジェクト、 マネジメントすべき4つの領域

翻って、ユーザー企業がシステム構築をプロジェクトマネジメント(PM)を通じて成功させるために、目を配らなければいけない領域はどこか。われわれは以下の4つだと考えている。

### <1> 全体管理

スコープ(目的と範囲)、時間、コスト、品質、人的資源、コミュニケーション、リスク、調達、統合管理の9つの観点のマネジメントを行う。

### <2> 改革/変革推進

プロジェクトに関連する変革(組織変革、業務改革など)、特にシステム導入と並行で実施すべき変革の推進や管理を実施する。

### <3> 利害関係者調整

プロジェクトオーナー、関連する業務部門・IT部門の管理者・社員、プロジェクトメンバー、社外関連メンバーなどを適時適切に巻き込み、協力的な関係性を維持する。

### <4> システム構築

システム構築に関わる作業で関与するベンダーの選定、スコープ調整、パフォーマンス(進捗、品質)の管理などを実施する。

通常、PM という(4)だけをイメージしがちだが、プロジェクトの目的は「システムを構築すること」ではなく、「作ったシステムを使って事業の価値を上げること」だと考えれば、この4つの視点なしでプロジェクトの成功はありえないと考える。

ここで先ほどの「プロジェクトの組み方」3つのパターンに戻ってこれを当てはめてみる。パターン(3)の、自ら複数のSIベンダーを管理するユーザー企業は、この4つを自社だけでマネジメントすることになる。当然、パターン(1)の「丸投げ」請負型SIベンダーも、パターン(2)のプライムベンダーも、いざプロジェクトが始まったら、ユーザー企業と並んでPMに責任を持つ。

実際、ユーザー企業も、契約書の条項や作業範囲記述書(Statement Of Work: SOW)で、SIベンダーのPMに対する

コミットメントレベルを高めようと工夫するのが普通だ。しかし、実態としては、ユーザー企業と同等レベルの立ち位置と視点でPMに責任を持つ(持てる)SIベンダーはほとんどいない。

なぜなら、彼らがPMにおいて責任範囲とするのは、自ずと「自社が構築を手掛けているシステムの全体管理」に限られるからだ。多少スコープを広げたとしても、自らが構築を手掛けるシステムに関わる改革/変革推進や利害関係者調整の部分まで。それも、自分の本丸である担当システムの構築にトラブルを抱えれば、当然自社にとっての優先度・重要度に合わせて資源を再配分してしまう。

具体的に、それぞれの領域で起こり得る状況について見てみよう。

※SIベンダーの立場で見る、プロジェクトマネジメント4つの領域

	SIベンダーの視点	その結果、ユーザー企業が抱えるリスク
1. 全体管理	<ul style="list-style-type: none"> <li>自らが担当するシステム構築・導入に影響がある範囲・内容中心で管理を行う。</li> </ul>	<ul style="list-style-type: none"> <li>SIベンダー視点での優先度、重要度判断が行われるため、部分最適化が進む。</li> </ul>
2. 改革/変革推進	<ul style="list-style-type: none"> <li>新業務・システムの影響がある改革/変革については、進捗確認レベルで実施。</li> <li>成果実現のために重要な改革/変革であっても、システムに関係ない内容については優先度を下げるケースも</li> </ul>	<ul style="list-style-type: none"> <li>戦略的変革/改革が進まない</li> <li>システム導入と改革/変革の足並みがそろわず、システム導入後の目指す成果が実現できない。もしくは、システム導入効果の刈り取りが遅れる。</li> </ul>
3. 利害関係者調査	<ul style="list-style-type: none"> <li>システムの業務要件定義に影響がある範囲の利害関係者調整を優先する。</li> </ul>	<ul style="list-style-type: none"> <li>利害関係者が多岐にわたる場合に、重要な利害関係者全てを納得させられない。</li> <li>システムプロジェクト外の利害関係者とのコミュニケーションが疎かになり、事後的に決定事項を覆しに来るなど大きな課題になる可能性がある。利害関係者とベンダーが対立関係になるケースも</li> </ul>
4. システム導入	<ul style="list-style-type: none"> <li>ここで品質低下などの問題が発生した場合、全ワークロードをこの領域の問題解決に集中する。</li> </ul>	<ul style="list-style-type: none"> <li>SIベンダー作業の管理に追われ、ユーザー企業側の管理が疎かになる。</li> <li>プロジェクト全体の成果実現に集中できなくなる。トラブルの現状をユーザー企業が認識しにくくなり、アクションが遅れてプロジェクト全体の遅延につながる。</li> </ul>

両社とも必死に取り組んでいるのに、このように互いの期待にギャップがあるままプロジェクトが進行し、気が付いた時には手がつけられない状況に陥ってしまう。こういうケースを経験した読者も少なくないだろう。

結局、自社のスキルが十分であろうがなかろうが、SIベンダーに盲目的に頼っている状況から脱却しなければ、プロジェクトの成功を実現することはできないのだ。ではどうすればよいのか？ 後編は、その具体的な解となるプログラムで

ある、プロジェクト・マネジメント・オフィス(PMO)について解説する。

(後編に続く)

※この論文は ZDNet Japan (2015 年 12 月～2016 年 10 月)に掲載されたコラムの再掲載となります。

# プロジェクトを「丸投げ」できない理由(後編)

## ——企業とSIの理想の関係



執筆者: 小室貴史 株式会社シグマクシス P2(Program & Project) シェルパ ディレクター

外資系コンサルティングファームを経て2012年シグマクシスに入社。経営管理、管理会計分野を中心とした戦略構想立案、プロセス変革、組織変革のコンサルティングに強みを持つ。流通業、製造業、商社、SIベンダー等に対するパッケージシステム導入のプロジェクトマネージャー、大型システムプロジェクトのPMOリーダーの経験を多数有する。

前編では、スキル不足に直面する企業とシステムインテグレーター(SI)の状況に触れた。そして、それでもなおプロジェクトを成功に導くためには、企業が自らの責任においてプロジェクトマネジメント(PM)を遂行する必要があることを述べた。

今回は、プロジェクトを推進していく上で重要な機能となる「プログラム・プロジェクト・マネジメント・オフィス(PMO)」について解説する。

### プロフェッショナルチームがプロジェクトを遂行

前編で述べたように、企業のみならずSIベンダー側も人材(人財)不足や最新スキルの習得に苦しんでいる現状では、特定のSIベンダー1社にプロジェクトを委託する、あるいはプライムベンダーに複数ベンダーをコントロールさせる形式は、リスクが大きい。

しかし複数のベンダーをユーザー企業自らがマネジメントしながらプロジェクトを遂行して成果につなげるのは難易度が高すぎる。ここで威力を発揮するのが企業内で、プロジェクトのマネジメント支援を専門に実施する部門「PMO」である。そして、ここでの重要なポイントは、このPMOが、ユーザー企業でもSIベンダーでもない「第三者」であることである。

SIベンダーの中には、プロジェクト間調整や全体管理を行うPMOを企業への提案のスコープに含めるところもあるが、そのPMOとここでいう第三者が担うPMOは、そもそもの位置づけがまったく異なる。前者は、SIのスコープ、主にシステム構築を中心とした作業をスムーズに動かすための周辺調整を目的とするものが多い。

つまり、自社の担当領域に影響があるところを中心に動くPMOだ。それに対して後者は、企業が駆動したプログラム(複数のプロジェクトの集合体)あるいはプロジェクトの成功のために、必要な打ち手を適時適切に行うことを目的とする。マネジメントする領域は、第1回で述べた、以下の4つの領域である。

## システム構築プロジェクト、マネジメントすべき4つの領域

### (1) 全体管理

スコープ(目的と範囲)、時間、コスト、品質、人的資源、コミュニケーション、リスク、調達、統合管理の9つの観点のマネジメントを担う。

### (2) 改革/変革推進

プロジェクトに関連する変革(組織変革、業務改革等)、特にシステム導入と並行で実施すべき変革の推進及び管理を担う。

### (3) 利害関係者調整

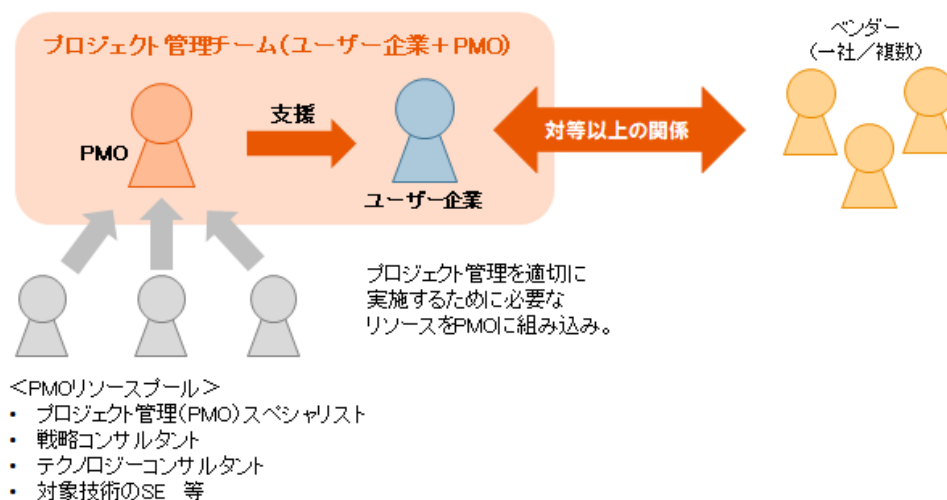
プロジェクトオーナー、関連する業務部門・IT部門の管理者・社員、プロジェクトメンバー、社外関連メンバー等を適時適切に巻き込み、協力的な関係性を維持する。

### (4) システム構築

システム構築に関わる作業で関与するベンダーの選定、スコープ調整、パフォーマンス(進捗、品質)の管理などを担う。

われわれが提唱、提供するPMOは、SIベンダーの中に紛れることなく、第三者としてユーザー企業のPMチームの一部としてプロジェクトに参加する。プロジェクトの4つの領域に目を配り具体的な打ち手を実行するためには、さまざまな知見が必要であるため、PMOは多様な能力を持ったメンバーによって構成される。メンバーは連携しあってフェーズごとに濃淡をつけながらPMO活動に参画し、プロジェクトのライフサイクル全体を通してユーザー企業のPMを支援していく。

PMOの立ち位置と構成メンバー



## PMOは「第三者」が良い理由

ここで改めて、なぜPMOが「第三者」である必要があるのか整理してみよう。

「鷹の眼」で見渡し、リスクの芽を顕在化する前に摘むことに専念できる

よいPMOとは、「仕切りが上手いプロジェクト事務局」のことではない。ほとんどの時間を、プロジェクトの失敗リスクを低減させるための目配りと細かい施策の実行に費やす泥臭い仕事だ。

対象となるプロジェクトにおいてどのような作成物(成果物)を、どの範囲、どの深さ(品質)でいつ作るべきかを適切

に判断する能力を備え、それを発揮し続けることが求められる。さらに業務およびシステム要件の定義内容を正しく評価し、その後の品質管理を着実に継続しなければならないタフな仕事である。

品質管理の必要性については、ことあるごとにユーザー企業も SI も唱えるが、問題が起きてから繰り返し広げられる社内レビューは「品質管理」とは言わない。特に、プロジェクトのステークホルダーが多くなるほど、「コトの後でのレビュー」は、各自が自社を守る場に転じるケースも多く、結果としてユーザー企業にとって正しい結果につながるかどうかという、大いに疑問が残る。

プロジェクトの成功のためには、顕在化する前のリスクの芽をつぶすことが何より大切だ。プロジェクトに参画する SI にとってもこれは決して悪いことではなく、うまく PMO と協力することで、プロジェクトのリスクを早期に把握し、コントロールすることも可能となる。特に、複数のベンダーが絡むプロジェクトでは、第三者がその立場にいてくれることによって、各社は自らが担当する領域の品質とスピード向上に専念することができる、というメリットもある。

### SI には言いにくいユーザー企業への提言が可能

プロジェクトは「ワンチーム」が理想だ。プロジェクトで協働するユーザー企業と SI が、友好的かつ対等の関係であることが望ましい。だが、頭では分かっている、それは「発注者」と「受注者」という関係性であるがゆえに非常に難しいことでもある。

どんなプロジェクトであっても、トラブルの起きないプロジェクトはないし、コントロール不能な外部環境の要因で予想外の展開になることもある。

大事なはそのリスクをいかに低減し、トラブルが発生したとしてもできるだけ早期に正しく解決することなのだが、例えば SI が作業遅延や品質問題を起こすと、発注者であるユーザー企業に対して SI が圧倒的に弱い立場に置かれるのが普通だ。

仮に、そのトラブルの原因がユーザー企業側にあったとしても、である。ここで問題の根っこを SI が指摘することは、わかっていたとしても心情的に難しく、このちょっとした齟齬の積み重ねがプロジェクトの終盤に大問題となって爆発することも少なくない。結果としてユーザー企業も SI も幸せにはなれない。

第三者の PMO であれば、双方の状況を的確に判断し、客観的な目線でユーザー企業と SI の間に入り、双方の役割分担と作業をコントロールすることができる。

繰り返しになるが、問題が顕在化してからレビューを繰り返してもプロジェクトで失われた時間とコストは取り返せないし、品質を取り戻すために要するエネルギーは莫大なものになる。リスクの芽を摘みながら着実にプロジェクトを進行させるためには、ただユーザー企業側に立つのではなく、企業と SI も含めた全体に目配りをできる存在が必要なのである。

### PMO の活用は、なるべく早期から

では、PMO はプロジェクトのどのタイミングから起動させればいいのか。われわれが PMO を担う場合には、遅くとも要件定義工程の後半か、設計工程の前半には参画している状態である必要がある、とクライアントには伝えている。その理由は、大きくは以下の 2 つだ。

#### 理由(1) 下流で顕在化するトラブルの多くは、上流で仕込まれている

プロジェクトにおける問題は、開発やテストなど下流工程で一気に噴出することが多いが、その原因の多くは要件定義や設計工程にある。検討スコープが漏れていた、要件定義が甘かった、ユーザー企業のキーマンの認識ギャップなどである。

よって、上流工程でそれらのリスクの芽が摘まれている必要がある。

**理由(2)プロジェクト体制や契約形態が固まってからの軌道修正は困難**

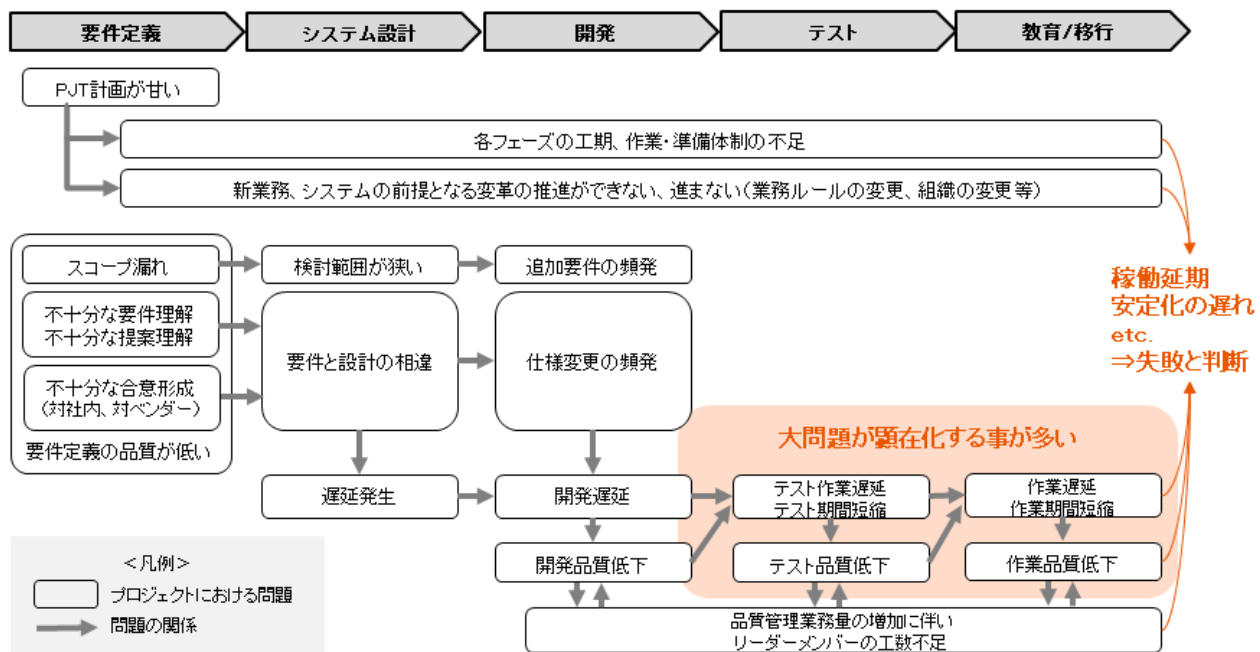
詳細設計・内部設計が始まり、開発工程まで進んでしまうと、SI側の体制変更や調整が難しくなり、顕在化した問題を解決するための労力が倍増する。たとえば、要件定義の甘さが原因で下流の問題が発覚したケースを想定して考えてみよう。

上流にさかのぼってのリカバリーも含めると、体制の強化やスケジュールの調整が必要となるわけだが、開発工程は、開発ボリュームや納期が契約上明記された請負契約になっていることが多く、単純にその変更だけで時間と工数が費やされていく。

また、ボリュームが大きいプロジェクトの開発では、投入人数がピークを迎える時期にあたり、参加している開発会社の数が最も多くなる。ここで計画と体制を調整することは、元締め側のSIとしても相当タフな作業になると同時に、リカバリーのための適材を見つけようにも空いていないという問題も起きやすい。

そもそも、プロジェクトの体制構築にあたっては、特定領域に強みを持つSIには、無理やり広い範囲の作業を担当してもらいよりも、得意作業に集中してもらった方がプロジェクトの成功につながる。PMOならばプロジェクトの本格稼働の前に、適材適所のアサインメントを、全体を見渡しながらか実施できる。SIから一部管理業務や作業を巻き取ることで、各社に自身の強みに集中してもらうことも可能だ。

システム構築プロジェクト、トラブルの芽は前工程にある



**理由(3)下流工程で「いざ！」という時に、PMOメンバー自ら問題解決要員に**

それでお下流工程で問題が起きてしまった場合、そのリカバリーができるのはどのような人財だろうか。新業務要件と新システムの設計要件の全体像を理解している人財か、新規に投入するなら、早急に要件をキャッチアップしてプロジェ

クトを最適化できる人財でなければ難しい。しかし、ユーザー企業、SI共に人財が潤沢ではない状況において、いずれも新規投入は困難だ。

こんな時、システム導入やプロジェクト管理の経験を有するPMOメンバーが要件定義工程に参画していれば、ユーザー企業側支援だけでなく、SI側の一部作業も支援できる。

前述の通り、PMO のメンバーは、システム導入の経験持っているメンバーで構成されているから、要件の深い理解があれば、いざというときには手を動かしてトラブル解決にも乗り出すことができるのだ。

## 「企業と SI の理想の関係」

PMO を有効活用することで、やみくもなベンダー依存から脱却しつつも、プロジェクトの成果を最大化できる可能性がある点を説明してきた。

PMO は、ユーザー企業と SI の関係性を良好に保ちながら、お互いの強みを生かし、最終的に双方にとってメリットを出してプロジェクトを完了させるための触媒のような存在だと筆者は考えている。

最後に、ユーザー企業と SI の関係性を考える時、長年 PMO としてさまざまな企業の各種大型プロジェクトに関わってきた筆者にとっての「理想的な状況」は、以下のようになる。

- 技術領域、役割において適切な SI が選定され、その役割がきちんと果たせる体制になっている。
- SI ベンダーの作業が適切に評価され、品質低下、作業遅延などの問題が顕在化する前にアクションがとられている。
- 各 SI ベンダーの強みが発揮できるよう、苦手と思われる作業はユーザー企業と PMO で巻き取ることができている。
- SI ベンダー間の役割分担に目を配り、抜け漏れを調整することでプロジェクト全体のリスクが低減されている。

次回以降は、筆者およびシグマックスが関わってきた各種プロジェクトのケースを紐解きながら、プロジェクトマネジメントの要点をより具体的に解説する。

※この論文は ZDNet Japan (2015 年 12 月～2016 年 10 月)に掲載されたコラムの再掲載となります。

# なぜシステム構築の現場では同じようなトラブルが繰り返されるのか(前編)



**執筆者:** 座間利行 株式会社シグマクシス P2(Program&Project) シェルパ ディレクター

外資系メーカー、外資系コンサルティングファーム、IT ベンダーのコンサルティング部門を経て現職に至る。主に、製造業(電機・電子部品、精密機器、医療機器、自動車)、およびエネルギー業界の企業に対し、業務改革、IT 戦略策定、システムの導入、保守まで一貫したサービス提供を得意とする。また、大規模システム導入プロジェクトにおけるプロジェクトマネージャーの経験を多数有する。顧客企業のビジネスゴール達成まで、変革支援をさせていただく事をポリシーとしている。

システム構築に関わる以上、誰もがトラブルは避けたい。「要件定義はユーザーを入れてしっかりやろう」「事業とIT部門のコミュニケーションが大事」といった話は、はるか昔から叫ばれてきた。コンサルタントはプロジェクトマネジメント方法論を振りかざし、識者は「システム構築のトラブルを回避するためにはこうするべし」というTIPSを唱え続けている。

それでもIT業界には「システム導入プロジェクトの6~7割が納期遅延、予算超過、品質問題などのトラブルを起こしている」という通説があり、感覚値でもその程度の割合で事故

が起きているのが現状である。きれいごとをいってもトラブルは回避できない。トラブルが起きる理由は実は水面下にあるからだ。

前回までに、最高情報責任者(CIO)を取り巻く環境の変化、ユーザー企業とSIベンダーが直面する現状、そして第三者PMOの重要性について解説してきた。今回は、PMOの必要性を改めて理解してもらうために、いったん視点を変えて、よくある「プロジェクトのトラブル」について、私の経験を踏まえながら追ってみたい。

## 初めてのトラブルプロジェクトレスキューは15年前

私がコンサルタントを職にして25年になるが、もともとはメーカーの情報システム部門SEとして社会人生活をスタートした。徐々に欲が出て、業務を知りたいとビジネスプロセスの中身に入り込み、それをシステム化して事業に貢献するようになった。

「こんなに喜ばれるなら、それを本業にしたい」と思い、当時まだ業界としてはマイナーだった外資コンサルティング業界に思いきって転じた。そこから、業務設計、システム設計、構築、導入という一貫通貫のプロジェクト経験を積んできたのだが、お客様の依頼で、レスキュー隊長としてトラブルプロジェクトに初めて飛び込んだのは、コンサルタントになって10年ほどたってからだ。



それは、社員のスキル評価会議で休日出勤していた時のこと。オフィスをうろうろしていたら、当時の上司に呼び止められ、「お客様が打ち合わせで来社される。同席して」と言われた。快諾して席につくと、そこに現れたのは、大手製造業 A 社の CIO だった。「今進行中のシステム構築プロジェクトがうまくいかない。使っているグループ系列の SI ベンダー X 社では立て直せない。なんとかしてほしい」と彼は言った。

そのプロジェクトには、当時私が所属していたコンサルティング会社も業務設計で参加していたが、SI ベンダー X 社下で契約。お客様の要請は、「全社視点でのプロジェクトの立て直しをしたい。もっと前面に立って仕切ってくれ」というものだった。「わかりました、立て直しは彼(筆者)がやりますから」と上司は一言答え、そこからプロジェクトレスキューとしてのキャリアが始まった。

ふたを開けてみると、プロジェクトの注力範囲(スコープ)も予算もオーバーし、プロジェクト人材(人財)のスキルはミスマッチ、SAP のプロジェクトなのに X 社には SAP の知識がなく、プロジェクトマネジメント経験も浅かった。既存システムの連携相手となる別のシステム子会社とのコミュニケーションもまったくかみ合っていない。X 社の SE は SE として必死にて頑張っているものの、そもそもプロジェクトとして成立する状態ではなかった。要はプロジェクトマネジメントが機能していなかったのだ。

急いで全体計画を見直してお客様に提示し、承諾をもらって PMO の役割で立て直しに入った。結局システムは無事稼働したが、稼働は半年遅れ、立て直しにかかった追加費用は最後、元請の X 社が責任をとって負担することになった。

もとはと言えば、お客様が SI 企業をしっかりと選定していれば、こんなことにはなっていなかったかもしれない。X 社も、無理に仕事を受けた結果、トラブルの末最後に費用を追加負担するという憂き目を見ることにはならなかっただろう。ではなぜこうなったかと言えば、もとをたどれば X 社の選定理由が、スキルの適性ではなく、「製造業 A 社のグループ会社

だから」だったことに起因する。似たような話は、今も耳にしないだろうか。

## コンペで負けた案件で、2 年後にレスキュー要請

今筆者が所属しているシグマックスでも数年前に印象的な案件があった。お客様 B 社の大型基幹システム再構築プロジェクト、シグマックスと SI ベンダー Y 社が最終コンペに残ったが、当社は「提案内容は良いが、なにしろ高すぎる」という理由で負けた。

正々堂々と戦って負けたのなら仕方ないと吹っ切って、何かあればまたお願いいたしますとその場から身を引いたのだが、既に稼働が始まっているはずの 2 年後に再び同じお客様が当社に現れた。「あのプロジェクトがうまくいっていない。なんとかできないか」——。切実な要請を受けて、筆者らはすぐに体制を整えて現場に飛びこんだ。

ふたを開けてみると、スケジュールは大幅遅延、設計フェーズが延々と続き、でもなぜそうなっているのか原因が誰にも究明できないという大変な事態に陥っていた。われわれは大至急調査分析とリプランを実施し、PMO として仕切り直しをマネジメントした結果、SI ベンダー Y 社と共に最終稼働までに持ち込んだ。

結果的にこのプロジェクトは、われわれがお客様に表彰される記念すべきプロジェクトとなったわけだが、お客様にとっては、当初の予定より 2 年の稼働遅れ、費用は当初の見積もりの 2 倍以上と時間もコストも大幅オーバーという負担を強いられる結果となった。

この件で興味深かったのは、これだけの大トラブルプロジェクトだったにも関わらず、PMO であとから救済に入った当社、もともとプロジェクトを仕切っていた SI ベンダー Y 社、そしてお客様 B 社は、最終的にとても良好な関係を築いてプロジェクトを完了できたということだ。原因調査時点、計画の見直し直後はぎくしゃくした関係だったものの、仕切り直した後は目標を共有し、適材適所でのアサインメントでそれぞれが

力を発揮し、ワンチームで取り組めたことが成功要因だったと思う。

ただし、そもそものトラブルの原因は、やはり「プロジェクトマネジメントの機能不全」だったことは、前述のケースとは変わらない。

## トラブルプロジェクトでは、 一体何が起きているのか。

ここで述べた 2 つのケース、1 つは 10 年ほど前、もう 1 つは 4 年ほど前の話だ。しかし同じような現象は相変わらずあちこちで起きて、われわれは PMO として走りまわっている。一体何が起きているのか。

過去の経験から共通にみられるのは、以下のような現象だ。

1. ビジネスゴールが不明確
2. スキルミスマッチなベンダー選定

3. 実質的なプロジェクトリーダーの不在
4. 事業部門と IT 部門のコミュニケーション不足
5. 要件定義における事業のキーマン不在
6. 終わらない設計フェーズ
7. 止まらないスケジュールと予算の超過

などなど……。こうしてみると「こんなことしていてプロジェクトがトラブらないわけがない」理由ばかりなのだが、「こうならないようにしましょう」といっても本質的な解決にはならないのは周知のとおりだ。

次回からは、「わかっているのに、なぜこうなるのか」を具体的なケースから掘り下げて、そこに横たわる本質的な問題について考えてみたい。

※この論文は ZDNet Japan (2015 年 12 月～2016 年 10 月)に掲載されたコラムの再掲載となります。

# なぜシステム構築の現場では同じような トラブルが繰り返されるのか(後編)



**執筆者:** 座間利行 株式会社シグマクシス P2(Program&Project) シェルパ ディレクター

外資系メーカー、外資系コンサルティングファーム、IT ベンダーのコンサルティング部門を経て現職に至る。主に、製造業（電機・電子部品、精密機器、医療機器、自動車）、およびエネルギー業界の企業に対し、業務改革、IT 戦略策定、システムの導入、保守まで一貫したサービス提供を得意とする。また、大規模システム導入プロジェクトにおけるプロジェクトマネージャーの経験を多数有する。顧客企業のビジネスゴール達成まで、変革支援をさせていただく事をポリシーとしている。

## 「ビジネスゴールが不明確なままプロジェクトが スタートしてしまう」理由

### IT 部門と IT ベンダーでとりあえず構想策定してしまう

前編では、筆者のプロジェクト立て直しの 2 つのケースを挙げながら、トラブルプロジェクトの傾向についてみてきた。ここでは、その経験の中から、よく起きる現象とそこに横たわる本質的な問題について考えてみたい。

プロジェクトのステップの王道は、構想策定→要件定義→設計→開発→テスト→導入だ。プロジェクトに関わる人間は、このステップを当然のこととして踏みながら進行する。だが、肅々と手順を重ねながらも、「成果物は作っているものの、本当にこの要件は定義すべき要件なのか?」「要件があふれすぎ。なんのためのプロジェクトなんだか……」というもやもやとした思いが渦巻くことはないだろうか。

この「もやもや」こそが、後半戦でのプロジェクト大爆発の予兆なのだが、原因は、前工程である構想策定でビジネス

ゴールが明確化されていない、そしてそれを再確認しながら要件定義を進められていないことにある。ではなぜビジネスゴールの設定がされないまま進行してしまうのか。

大型システムプロジェクトは、大きく「IT 部門主導型」と「ユーザー部門主導型」に分けられる。IT 部門主導型の場合、多くはシステムの老朽化対策、ハードやソフトの保守期限切れ対応や事業継続計画(BCP)対策、といったことが目的で、ユーザー部門主導型の場合は、さらに「現行業務改善型」と「業務改革実行型」に分かれる。構想策定でビジネスゴールが不明確なまま走ってしまうのは、多くの場合、IT 部門主導型と、ユーザー部門主導&現行業務改善型のタイプだ。

まず、IT 部門主導型は、放っておくと「しかたなくやるもの」「インフラコスト」として扱われるため、ユーザー部門はそもそも興味を持たないし、経営からも「できるだけ手間もコストもかけずにやれ」と言われる。ユーザー部門主導&現行業務改善型も、全社視点で構想を立てられるキーマンの確保がままならないのが普通だ。

現行業務に引きずられることなく、全社最適化の視点で新しいシステムのあり方を描けるレベルの事業部の人材(人財)は少ないうえに、いたとしても売れっ子だし自分の職務で忙しい。そもそも事業部門と IT 部門は互いに距離があり、コラボレーションしようというモチベーションが低いという問題もある。

何より、「それでも大事な仕事だからやってくれ」とトップマネジメントや役員が社内のキーマンを任命するほど、「システムプロジェクト」は社内で重要視されてもいない。

人財がいなくなれば、外部の経験豊富なコンサルタントの力を借りたいと思うが、企画構想フェーズは実行稟議を通す前の段階なので IT 部門は予算がとれない。となると、ここで頼れるのは、自社商品やサービス営業を念頭においた IT ベンダーたちの提案だ。結果的に、IT 部門の技術的視点と IT ベンダーの営業的提案を組み合わせる最初のフェーズの成果物、「基本構想」がなんとか作り上げられる。

こうして、せつかくの時間と予算の投資をする機会にも関わらず、次世代の事業に資するビジネスゴールが一切議論されぬまま、要件定義フェーズにコマが進められていくのだ。これでは、本質的な要件定義ができるわけがない。

## 「要件定義がなかなか終わらない」理由

キーマン不在で膨れ上がる要件の中、プロジェクトメンバーはピンポン状態に

さて、要件定義とは、エンドユーザーの言い分をくまなく吸い上げることではない。新たなシステムを作るにあたっての基本構想(ビジネスゴール)にのっとり、業務プロセスの何を捨て、何を新しくし、何を变えるのかを決めていく、いう極めて重要なフェーズだ。ここでは、システム化の対象となる業務のプロセスオーナーがガッチリと関与し、新プロセスの意思決定をしていくことが求められる。

業務に詳しく、ビジネスゴールとの関連性を意識しながら意志決定ができる人間がいないと、当然要件定義は迷走する。

例えば、「在庫削減プロジェクト」をイメージしてみよう。調達・製造業務に携わる人は、在庫を減らしたい。しかし営業部門は、在庫不足はリスクなのでできるだけ多く抱えたい。全社視点に立つなら、「需要予測のプロセスも並行して考えよう」という話になるのが正しい姿だが、意志決定者が不在だと声の大きい方のユーザーが勝ってしまう。結果として、プロジェクトメンバーは「あちらの部門ではこう言われ」「こちらの部門ではああ言われ」をピンポン玉のように繰り返し、延々と要件が固まらない。

プロジェクトとは、錯綜する数々の利害関係者を調整しながら、ビジネスゴールを達成する取り組みだと考えている。よって、こういう局面ではまずそれぞれの言い分を聞き、ビジネスゴールと照らして着地すべき合意点を見つける。それぞれにメリットデメリットがあっても、最後は全社にとってそれが正しい選択であるということをはっきり理解してもらうようにコミュニケーションし、合意形成しながら進めていくというのがあるべき姿だ。

翻って、企業内でそういうことができる人財は、業務に精通しているだけではなく、コミュニケーションやファシリテーションに長けて、意思決定のリーダーシップを備えている、ということになるが、例によって、そういうキーマンはシステムプロジェクトにアサインされないのが常だ。そして、「ピンドメ役」不在のまま、要件定義フェーズはずるずると遅延していく。

## 「システム設計がいつまでたっても終わらない」理由

忍び寄る「刺身」の誘惑

さて、こうなると、そろそろプロジェクトチームのお尻には火がついてくる。要件定義は終わっていないがスケジュール上

はもう設計にはいっていただければならないからだ。ここで多くのプロジェクトはこう考える。「設計フェーズをやりながら要件定義をして遅れを取り戻そう」——。しかし、これは絶対に手を出してはいけない「禁断の木の実」だ。

そもそも要件定義が終わらない理由は、「決められなかった」ことにある。引き続き二転三転する要件定義に加えて設計を並行して動かすと、要件に振り回されて設計手直しの無限ループが始まる。こうなると、SI ベンダーを含めたプロジェクト全体が精神的肉体的に疲弊し、モチベーションが下がり、「とにかく早くプロジェクトを終わらせる」ということだけを目指すようになる。生産性は落ち、品質は低下するという負のスパイラルに突入し、終わりの見えない設計フェーズが延々と続くことになる。

本来、タスクの矢羽がきれいにシーケンシャルに並んだスケジュールでプロジェクトを進行させるべきところを、フェーズを折り重ねて時間を縮めようとするアプローチのため、われわれはこれを「刺身状態」と呼ぶ。刺身の切り身が一部重なりあって皿に並んでいるイメージだ。ちなみに前編で触れた、われわれがお客様のレスキュー要請を受けて立て直しに入った 4 年前のプロジェクトも、まさに設計フェーズが「刺身状態」で負のスパイラルに陥っていた。

思い切って一度前フェーズに立ち戻り、スケジュールを刺身からシーケンシャルな矢羽に組み立てなおして再起動する、という取組みの結果、最後は稼働に成功したものの、このプロジェクトは 2 年の遅延、お客様は 2 倍のコストを負担することになった。「刺身」には絶対に手を出してはいけない。これは鉄則である。

## 「このままでは無理だとわかっている、仕切り直しに踏み切れない」理由

### 減点主義の組織では自ら黄色信号を出すことは難しい

通常、設計フェーズで「刺身」が登場すると、システム構築経験者なら誰もが「これはやばい」と思う。このまま進めてうまくいくはずがない、どこかで仕切り直しをしないと大変なこ

とになる、と誰もが心の中では思っている。だが、プロジェクトリーダーは、プロジェクトオーナーに「何とかあります」と言い続け、不安を見せるメンバーには、「根性でやれ」「ベンダーをたたいてもやらせろ」と強引な要求をつきつける。

プロジェクトリーダー個人としては決して悪気があるわけではないのに、なぜそうになってしまうかと言えば、プロジェクトを発足させるときに「X 年、XX 億円で完了します」ということをコミットして稟議を通しているからだ。それができないということは、シンプルに組織の評価で「バツがつく」ということを意味する。

本来は、どんな稟議でスタートしようとプロジェクトは成功させなければ意味がないし、不測事態に見舞われないプロジェクトも存在しないのだが、「トラブルがあったら助ける」というモードが企業内にはない限り、プロジェクトリーダー自ら「トラブルが起きています、助けてほしい」と声を上げることはできない、というのが普通の感覚だ。

ちなみに、SI ベンダーはこういう場合どう対応するのか。多くのケースでは、まずスケジュールが遅延し始めた当初は、「なんとか頑張ります」と答える。そして行き詰ってくると、「サービスインの期日は死守します、でも追加のフィーと人員追加の許可を」と要望を出す。しかし、人員を追加しても、急ぎよ集めたメンバーのスキル・品質はバラバラで、要件のキャッチアップも思うようには進まない。

そもそも「刺身」が出現した時点で、SI ベンダー社内では、「あのプロジェクトはトラブっている」と認識され、「あのプロジェクトへのアサインメントは危ない」という機運が広まっているのが普通だ。そんな状態でアサインされた新たなメンバーの心身のモチベーションが高いはずはなく、人数が増えてもスピードは上がらず、さらなる「フィーと人員追加」が繰り返される。こうしてプロジェクトのコンティンジェンシー予算はどんどん切り崩され、遂にプロジェクトは暗礁に乗り上げる。

ある意味、ユーザー企業側よりもプロジェクト経験のある SI ベンダーのほうがトラブルへの感度は高いはずだが、それでも何も言わずに踏んばろうとするのはなぜか。

それは、自らトラブルのサインを声高に出すことは、自分たちを雇ってくれているプロジェクトリーダーを背中から刺す行為に等しいからだ。ビジネスの取り引きがある以上、それは仁義に反する行為という感覚がある。こうして、誰もがうしろめたさを抱えながら、いつか座礁するとわかっている船で航海を続けることになる。

## 改めて「プロジェクトを成功させること」の重要性を考える

ここまで、筆者の経験を振り返りながら、典型的なトラブルプロジェクトの原因についてみてきたが、結局大事なポイントは2点、「プロジェクトのゴールを明確にして、それを達成することに集中すること」、そして「プロジェクトの成功に向けて必要なことはすべてやること」だ。おかしいと思うことは指摘し、問題だと思うことはきちんとつぶし、「もやもや」を残さず、着実に前進し続ける。うまくいかなければやり直す勇気も持つ。

最終的に予算と納期を守り、狙ったビジネスゴールを達成できれば、プロジェクトは成功だ。本来、ユーザー企業側もSIベンダー側も「やるからにはプロジェクトを成功させたい」と思っているのだから、その気持ちに素直にのっかってやりきるということだ。

マネジメントがシステムプロジェクトの重要性を理解できない、社内外にスキルが足りない、プロジェクトマネジメントの能力が不足している、事業部門とIT部門とのコミュニケーションがうまくいかないなど、制約となる条件は多々あるにせよ、デジタル化が進む今、システムプロジェクトの遂行なしで事業運営ができないのは厳然たる事実だ。

本連載の第2回、第3回で述べた通り、この現状を打破すべく客観性を持った第三者PMOも活用して、ユーザー企業もSIベンダーもともにプロジェクトの成功にまい進できるチーム作りにぜひ取り組んでいただきたい。

筆者が経験した泥臭く生々しい事例はまだまだまだたくさんあるが、本稿が読者の皆さまの何らかの気づきになれば幸いだ。今プロジェクト進行中の皆さま、これからスタートする皆さまが、前向きでクリエイティブなプロジェクトを経験できますように、とお祈りすると共に、解決できなさそうなトラブルの予感がしたら、第三者の力を借りるという選択肢を思い出しで欲しい。

※この論文はZDNet Japan(2015年12月~2016年10月)に掲載されたコラムの再掲載となります。

# 発注者の力量が試される ——アジャイル開発の拡大とプロジェクトテスト変革



執筆者： 小室貴史 株式会社シグマクシス P2(Program & Project) シェルパ ディレクター

外資系コンサルティングファームを経て2012年シグマクシスに入社。経営管理、管理会計分野を中心とした戦略構想立案、プロセス変革、組織変革のコンサルティングに強みを持つ。流通業、製造業、商社、SIベンダー等に対するパッケージシステム導入のプロジェクトマネージャー、大型システムプロジェクトのPMOリーダーの経験を多数有する。

第4回、第5回では、「頭ではわかっているのにプロジェクトがトラブってしまうのはなぜか」というテーマで、プロジェクトの現場で起きている現象の背景を紐解き、その原因について考えてみた。そして、ユーザー企業、SIベンダーそれぞれの事情が折り重なってトラブルが避けられないポイントについては、企業内で、プロジェクトのマネジメント支援を専門に

実施する第三者の部門「PMO」の力も借りて乗り切り、プロジェクトの成功に向けて集中できる体制を作ることの大切さについても触れた。

今回は、より具体的に「テスト」のアプローチを例に、新しいプロジェクトの進め方を考えてみる。

## 「刺身」の出現と同時に簡単に吹っ飛ぶ 「テスト設計と準備」

多くの読者は、ウォーターフォール型開発の品質保証における「Vモデル」という手法をご存知だと思う。提案時にテストの成功要因としてVモデルの活用を強調するSIベンダーも多い。しかしながら、実際にプロジェクトを走らせてみると、テスト設計や準備作業の遅延がたびたび発生しているのが現状だ。

システムテストや受入テストの設計、準備の作業を検討できる立場の人間は誰か、というと、SIベンダー側で業務領域の設計に責任を負うリーダー（業務リーダー）である。普通に

考えれば、業務リーダーは、システム設計を終えて開発工程（コーディング／単体テスト）に入ってしまうと、工数に余裕ができて余裕をもってテスト準備に入れると思われがちだ。

しかし実際には、業務運用の詳細定義や、クレンジングやシステム対応検討を含めた移行・教育の準備といった膨大な作業があり、特にユーザー企業側メンバーを巻き込んでの動きに追われている。

ここに、「上流で仕込まれたトラブルの種」が花開き、システム設計の品質問題、開発者のスキル問題で単体テストにトラブルが勃発したりすると、それだけでなくギリギリで動いている業務リーダーのワークロード（作業負荷）は逼迫する。

SI ベンダーにとって、担当領域の遅延は言い訳無用なので、後ろのテストの準備があろうがなかろうが、業務リーダーは開発領域の火消しを最優先に飛び回ることになる。

そして、各フェーズのスケジュールが折り重なった「刺身状態」が出現すると、最後の山場となるはずのテストに向けた設計や準備といった、下工程で最も重要なステップがなし崩されたまま、プロジェクトは最終局面に突入することになる。

## 日本人はなぜ「テストツール」を使いたがらないのか

テストツール 例	内容
負荷テストツール	昔からよく使われている、多ユーザー同時アクセス状態や高付加状態を作り出すツール
テスト自動化ツール	決まったシナリオやデータでのテストを自動で繰り返し行えるツール(データをファイル等から読み込みループ処理をすることもできる)
ソースコード自動解析	ソースコードを自動で解析し、バグやリスク要素を自動で洗い出すツール
スタブ機能提供ツール	スタブ(複数機能の連携テストにおいて、相手側機能が出来上がっていない際に準備する相手側の代替機能)を容易に準備できるツール

人間がやらなくてもよいテストを代わりに精度高くやってくれる「テストツール」。日本のプロジェクトではなぜ積極的に使われないのかというと、多くのプロジェクトが、「作業はなんでも人間がやる」ということを前提に設計されているという背景がある。

例えば、インドではエンジニアでも単価が安い人と高い人がいる。高い人には、領域は限られても難易度の高いテストを、安い人には簡単でも大量の処理をこなすテストを割り振らなければ採算が合わないから、最初からテストをできるだけ要素分解して設計するのが普通だ。よって、良いテストツールが登場すれば、「人間にやらせた方がいいもの以外」はテストツールに任せるということが簡単にできてしまう。

しかし日本では、そんなにエンジニアに価格の幅がない。そして SI ベンダーは「人月と工数」で稼ぐビジネスモデルだから、できるだけ多くの人をプロジェクトにアサインしようとす

こうならないために「上工程の問題は上工程で潰そう」、ということは、連載の中で一貫して述べてきた通りだが、本稿では下工程側から見たときの 1 つの問題解決策として、W モデルとテストツールの併用をお勧めしたい。このアプローチで、テストフェーズの効率化、そして遅延リスクと品質リスクの大幅な低減が可能だ、と筆者は考える。

既にご存じの通り、テストツールにはいろいろな種類があり、うまく活用するとテストの品質向上や工数削減の効果を期待できる。だが、負荷テスト用のツールは昔からよく使用されているものの、テスト自動化ツールやスタブ提供機能などは、日本ではまだ活用事例は少ない。

る。SE の人数をコントロールすることなく、みんなでテストをするわけだから、「要素分解しなければならない」というモチベーションがプロジェクト全編通して働かないのだ。だから、いざテストツールを手にしても、何をツールに投げて、何を人間の作業に残すかという切り分けが簡単にできない。

ましてや下工程が逼迫した状況において、「決してテスト遅延は許されない」という立場に追い込まれていたら、「無駄な工数がかかる」「作業遅延のリスクがある」という理由を挙げて、SI ベンダーは使い慣れないツールの使用を避けたがる。だが、そもそもツールを使う理由それこそが、プロジェクトの品質向上、遅延リスクの低減なのであるから、これでは完全に本末転倒だ。

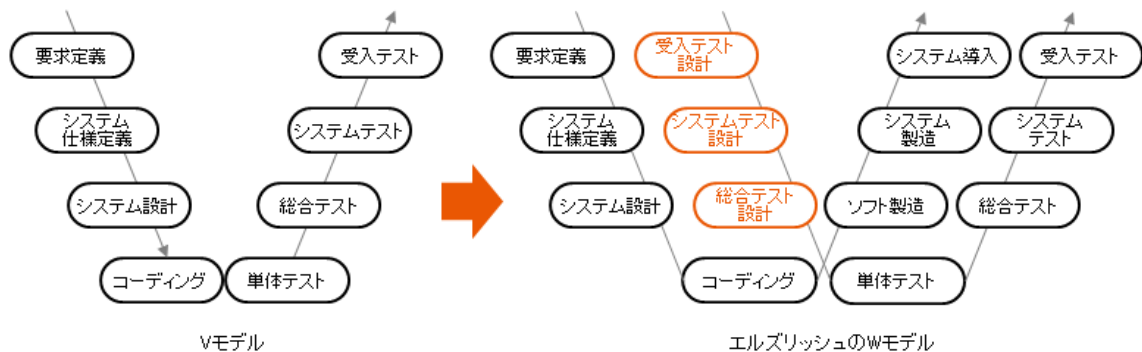
## W モデルとの組み合わせで テストツールは効果倍増に

そこで筆者が提唱するのが、W モデルとテストツールの併



用である。W モデルは、V モデルの改良版といってもよい。V モデルと W モデルの違いは、テスト設計作業をいつどのよう

にやるべきかを明確に示している点にある。



VモデルとWモデルの違い

W モデルでは、要件定義の後にすぐ受入テストの設計を、システム使用定義の後にすぐシステムテストの設計をすることになる。テスト設計を早期に実施することでテストの準備がより早く進み、テストの難易度、必要期間や工数がより詳細にわかるようになるのだ。

要素分解することで、どのテストにどのツールを使い、何を人間の手で検証するかを定義し、詳しいテスト計画を作成することができる。さらに、具体的なテストのデザインを並行して進めることで、設計の不備を検知し、下流でのトラブルの種をここで潰しておくこともできる。

実際のところ、テスト設計の適任者である業務リーダーは、開発フェーズに入っても前述の通り全工程において余裕がないから、Wモデルでテスト設計するには、それができるリーダークラスの人財を上工程に追加しなければならなくなる。

しかし、設計をきちんとやった上で適切なテストツール正しく使いこなせば、下工程のテストフェーズにかかる工数とコストは確実に削減できる。そして当然テストの精度もあがる。Wモデルとテストツールの「併用」は、プロジェクトの下流工程の遅延リスク削減のひとつの鍵なのだ。

ちなみに、テストツールは1つのプロジェクトだけの効果を見て費用対効果を見るべきではないと筆者は考えている。複数回実行するテストの2回目以降の人的ミス低減と工数減にも効果がある、という意味では、例えば組織変更対応な

どの運用保守において、その効果を評価できる。ソースコード自動解析ツールは、多くのプロジェクトで活用することで品質の安定化が見込めるし、スタブ機能(完成済みプログラム動作検証のための代替プログラム)提供も複数プロジェクトで効果を出すだろう。

当然、アジャイル型開発が進めば進むほど、スピードと精度が求められるようになるわけで、テストツールにお任せする領域は必然的に増えてくるだろう。

Wモデルとテストツールの活用は、単純に開発の効率化の実現にとどまらず、日本のようにエンジニアの価格が高く、かつユーザー企業の求める品質レベルが高い国においては、プロジェクトの進め方そのものに影響を与える新しいアプローチだ。人間の労力だけに頼るプロジェクト運営から脱却し、機動性と正確性の高いプロジェクトの実現に、私たちもチャレンジしていきたいと考えている。

## 日本人がインドのオフショアをうまく使えない理由

最後にオフショア活用についても触れておきたい。欧米の企業は、インドなどの低価格かつ高スキルのオフショアをうまく活用している。金融系の大規模システム導入においては、テストの工程のみをインドのリソースとテストツールを活用する別のSIベンダーに外だしする、というケースもみられるようになっている。

日本人と同スキル、同レベルの人財を半額以下の単価で調達できるのだが、日本ではその活用が進まない。その大きな理由に「言語の壁」がよく上げられるが、筆者はそれ以前に「品質管理」の考え方の違いが大きいと考えている。日本でインドの開発者を活用する場合、日本語でプログラム設計書を詳細に記述し、それを英語に翻訳する。また、コミュニケーションのギャップを解消するのに、ブリッジ SE を置く。

欧米では、要件定義は自国内で実施するものの、設計の重要な部分をインド側に任せるケースが多い。彼らは、いきなり精度の高い詳細設計を行うのではなく、いったんインド側に作らせた実物をみながら、何度かのやり取りで精度を上げていく、というプロトタイプングのアプローチを許容するのだ。

結局、使う側がプロジェクトの成功、品質向上、そしてリスク低減に主眼をおいて、「何を自分でやり、何を外に出すか」「どれを人間でやり、どれをツールに投げるか」といった要素分解ができていなければ、インド人開発者であろうがテストツールであろうが、うまく使いこなすことはできない。

アジャイル型開発の必要性が高まり、AI の進化に伴ってソフトウェアの高度化が進む今、日本企業は、システム構築のあり方を根っこから見直す時期に来ているのではないだろうか。

※この論文は ZDNet Japan (2015 年 12 月～2016 年 10 月)に掲載されたコラムの再掲載となります。

# 研究開発のプロジェクトマネジメント ——技術の「組み合わせ」とイノベーション



執筆者： 小室貴史 株式会社シグマクシス P2(Program & Project) シェルパ ディレクター

外資系コンサルティングファームを経て2012年シグマクシスに入社。経営管理、管理会計分野を中心とした戦略構想立案、プロセス変革、組織変革のコンサルティングに強みを持つ。流通業、製造業、商社、SIベンダー等に対するパッケージシステム導入のプロジェクトマネージャー、大型システムプロジェクトのPMOリーダーの経験を多数有する。

ITの世界で「プロジェクトマネジメント」と言えば、それはシステム構築のプロジェクトマネジメント(PM)を意味する。しかし、もともとPMは、建設業界やプラント業界で生まれ、磨かれ、それがIT業界に持ち込まれたものだ。いずれにおいても、「ゴールと期限があり、複数のステークホルダーが関わりながら、1つのものを作り上げる」という活動であることは変わりなく、その規模が大きくなればなるほどPMの重要性は

増す。

そしてその適用範囲は、IT業界からさらに広がり、他の分野にも適用され始めている。最終回の今回は、最近当社が関わっている研究開発(R&D)領域のPMについて触れ、本連載を締めくくりたい。

## 研究職プロフェッショナルを マネジメントできるのか

当社がシステム構築プロジェクト以外のPM支援を開始したきっかけは、当社ウェブサイトのフォームに投げ込まれたお問い合わせだった。当社のサービスについて話を聞きたいというA社の依頼を受けて、筆者と数人のメンバーで先方にお伺いした時、いろいろ話を聞くうちに浮かび上がってきたのが、「産学連携型研究開発をやっているが、対象組織も多く、研究者がメンバーということで、マネジメントに不安を感じている」という現状だった。

A社が手掛けているのは、複数大学、複数企業によるコンソーシアムを形成し、分科会的に分かれた各組織が担当領域の研究開発を進め、革新的な製品を開発する取り組みだ。当然投資額も大きく数年間の大規模プロジェクトなわけだが、年次で成果を上げなければ、製品製品開発計画が見直され、投資も縮小または打ち切られてしまう可能性もある。

多くの関係者を巻き込みながら着実に進捗し、最終ゴールを達成しなければならないという活動は、大型システム構築とまったく同じなのではないか。そう思った私たちは、これ

までの経験がここでも生かせるのではないかと考え、PM 支援を開始した。

入ってみると、ステークホルダーの状況は極めて複雑だった。何しろ、コンソーシアムに入るモチベーションが各社各大学バラバラだったのである。「新しい技術開発に取り組みたい！」と積極的なところもあれば、「補助金をもらえるなら参加する」という受け身なところもある。何しろプロジェクトメンバーの全員が、研究者たちだ。「コントロールされることを好まない(というか、「されてたまるか」)」という自主自立の研究者気質を前提に、すべてが動いている。

A 社のリーダーからは、「自社でプロジェクトをマネージすべきだとは思いますが、研究者たちに進捗管理や、課題管理を自らすモチベーションはないし、なにより研究開発に没頭させたい」と言われていた。であれば、私たち自身が第三者として研究開発プロジェクトのプログラム・マネジメント・オフィス (PMO) をやろう、と腹をくくって取り組んだ。

## R&D にウォーターフォールアプローチは全く通用しない

現場でまず私たちが直面したのは、「作業を計画通りに積み上げていっても成果が上がるとは限らない」という、これまでのわれわれの常識を覆す事実だった。所定の成果を出すまでの期限に向けて、プランを作り、工程ごとにマイルストーンを切り、スケジュール、品質、課題とステークホルダーを管理し、プロジェクトを推進するという基本的な考え方は、通常のプロジェクトとは全く変わらない。が、R&D のプロセスの基本は「トライ&エラー」である。

たとえば、次の工程に進むために、ある物質の耐熱温度を 100 度から 200 度に上げ、かつコストダウンを実現したいといった研究開発テーマがある。そのテーマをクリアするための仮説検証の回数は一体何回になるかわからない。実験して失敗し、また仮説を変更して実験して失敗し、そして成功するといった繰り返しが R&D だ。

通常、システム構築プロジェクトでは、「やり直し」が発生しないように工程をウォーターフォールアプローチで設計し、タスクを月、週、日といった期間で切り、必要なリソースを割り当て、成果物を定義し、進捗をチェックするというやり方が王道だが、その定石が全く通用しない。要は、R&D においては、工数と成果に厳密な相関関係がないのだ。

そうなると、進捗のマイクロマネジメントには意味がなくなってくる。ましてや実験に取り組んでいるのは管理されることが苦手な研究者たちだ。自由に泳いで研究取り組んでもらいながら、スケジュール上のクリティカルパスを外さない時間軸で成果を出すことを管理するにはどうするか？ それが私たちのテーマとなった。

そこで、通常日次で行っている進捗確認のサイクルを週次あるいは隔週にしたり、レポートで可視化する内容の粒度をあえて落としたり、と適切な落としどころを探り当てるために、私たち自身も PM のトライ&エラーを続けた。

結果たどりついたのは、「研究テーマごとにこれ以上仮説検証を繰り返しても前にいけない」というタイミングを見極め、「そろそろ別の仮説設定に戻ってははどうでしょう」ということをわれわれが促すやり方だった。

要は、目標達成に向けての進行状況をコントロールすることが、R&D における PM のキモだ、と私たちは気が付いたのだ。これを、「進捗管理」に対して「目標管理」と呼んでいる。目標管理は「研究開発ステップ」の達成度と、「目標達成見込み」の達成度を掛け合わせて、「目標達成度」を管理するものだと考えてもらえればよい。

研究開発プロジェクトの目標管理  
 各研究開発テーマ毎に、研究開発ステップ及び目標達成見込みを把握し、  
 目標達成度で進捗を管理することにより、目標達成の確度を高めた。

研究開発ステップ定義

N O	研究開発ステップ			達成度①
	要素技術研究	モジュール開発	プロセス開発	
1	コンセプト検討	コンセプト検討	コンセプト検討	20%
2	検証	設計	立上げ・改造	40%
3	試作	試作	検証	60%
4	実証	評価	ライン立上げ・ 実証	80%
5	完了	出荷	完了	100%

目標達成見込み定義

N O	目標達成見込み		達成度②
	目標変更なし	目標達成見込み	
1	目標変更なし	A:実現可能	100%
2	目標据え置き だが、課題解決 必要	B:打ち手あり、実現可能性高	75%
3		C:打ち手あり、実現可能性中	50%
4		D:打ち手あり、実現可能性低	25%
5	目標変更必要	E:実現不可能	0%

目標達成度定義

N O	研究開発 テーマ	目標	研究開発ステップ					目標達成見込み ステータス	目標達成度 (達成度①×達成度②)
			コンセプト検討	検証	試作	実証	完了		
1		コストXX%削減	★					A	20%
2		開発リード タイムXX%削減		★				B	30%
3		XX技術の開発			★			C	30%
4						★		D	0%

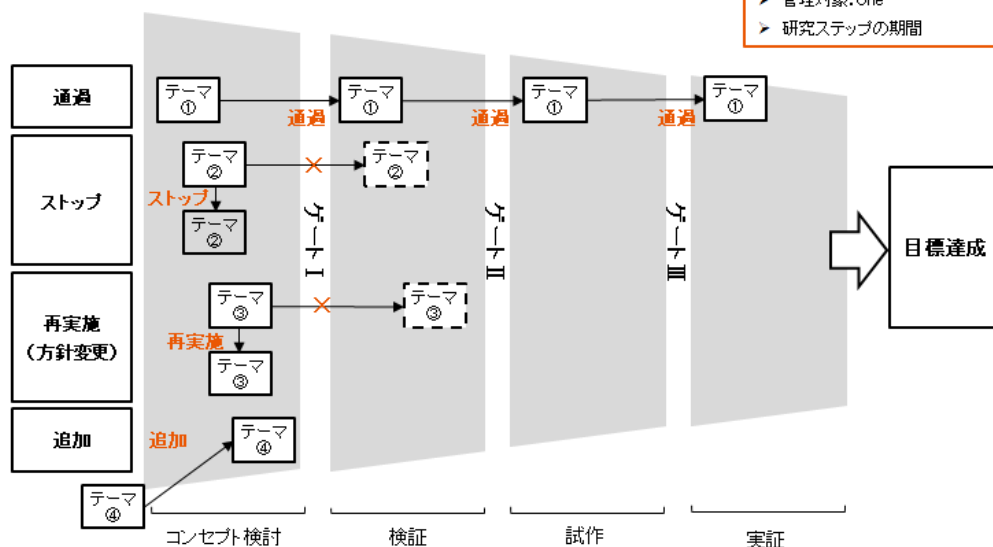
また、従来から存在する管理手法であるが、ステージゲート方式を活用して研究開発全体の進捗状況把握と確認ポイント  
 における意思決定も行うようになった。

目標管理

スマートゲート方式  
 各研究テーマが次の研究ステップに進むタイミングで、目標の達成状況を客観的に評価した上で、テーマの通  
 過/ストップ、再実施、追加の仕組み(ゲート\*)を設ける

\*ゲート通過基準(案):全テーマの「実現確度×効果」の合計で判断(Σ=実現確度×効果)

- 確認ポイント
- 導入の可否
  - 管理対象:One
  - 研究ステップの期間



ここで大事なのは、研究者 1 人ひとりの個性を把握し、何をやっているのか、何につまずき、何に悩んでいるのかを知り、必要に応じてステークホルダー間の調整をわれわれが担い、各所に散らばる「仮説」の数を少しずつ収束させながら、期限内に技術的なゴールを達成させるという動き方だ。

報告内容は極力シンプルにし、研究者に負担を掛けないように配慮することで、研究者の能力とモチベーションをフルに発揮してもらう。私たち自身、プロジェクトメンバーとして中に入り込んで、客観的な目線でプロジェクトの収束をサポートする、という日々が続いたわけだが、これらの PM が成功した背景には、研究者を束ねる A 社プロジェクトオーナーの強いリーダーシップがあったことは言うまでもない。

ちなみに、この目標管理のアプローチは、ウェブアプリ開発などで用いられるアジャイル開発の PM にも適用できる。開発する側もあえて請負開発とはせず、ユーザー部門をうまく巻き込み、ベータ版を開発して所定の回数機能確認してもらい、仕様を固めてゆく工数まで含めた支援契約で開発すればよい。ユーザーの満足度を上げ、かつ開発コスト上振れとスケジュールの遅延を起こさない意味で、新しい方法論として確立されつつあると言えよう。

## アートや段ボールで、シャイな研究者たちのコラボを促進

R&D の PM でもう一つ大きなチャレンジだったのが、コラボレーションの促進だ。研究開発プロセスの効率を上げる、あるいはアウトプットの価値を高めるためには、多種多様な研究者のコラボレーションが不可欠だ。特に、グローバルの過酷な開発競争を勝ち抜くために産学協同のコンソーシアムを作っているわけであり、各組織の英知を結集しなければそもそも取組みの意味がない。

しかし、モチベーションひとつとっても、企業の研究者は新製品開発やコスト削減、品質向上といった、ビジネス価値に貢献する技術や素材の発見に向いているのに対し、大学の

研究者は物理現象のメカニズムの解明のような、原理原則の新発見にエネルギーを注ぐ。

このように企業と大学で研究に対する目的の違いがあるのに加え、そもそも高い専門性を持っている研究者は、シャイでコミュニケーションが苦手という性質を持っている。いかに彼らを 1 つのベクトルに向かわせ、自らコラボレーションするように促すか、という点に数々の工夫が必要だった。

1 人ひとりの専門性とモチベーションの方向を知り、シナジーが出やすい単位でプロジェクトを組むというアサインメントをする、目標やコンセプトの共有を徹底する、という点にエネルギーを注いだ。

個々の研究者がタコつぼ化しないようにするには、共同作業の面白さやそこから生まれる新しい発想を体験してもらうのが一番、ということで、ダンボールや粘土など容易に形作ることができる素材を使って短時間でコンセプトを形にして擦り合わせをする、「絵を描く」という行為を通じて研究開発への想いを共有しベクトルを合わせていく、というワークショップも行った。

クリエイティビティが求められる研究者が組織や専門性の壁を越えるためには、言語よりもむしろアートや工作による表現の方が自由で向いているともいえる。いかに「共感」の場を用意できるかがポイントであるように思う。

アートを活用した研究者のワークショップ

1日の流れ	ワークショップの目的・内容
9:00	EGAKU
12:00	
13:00	TSUKURU
17:30	

**目指す方向性の確認、一体感醸成**  
 アートの創作と鑑賞を通じ、チームが想いを共有し、目指す方向性を確認するとともに、新たな視点を生み出すオープンな場を創る

**開発コンセプトの作業・共有**  
 組織を超えたチームで、研究開発テーマを段ボールなどを使ってプロトタイプングし、意見交換を行う

コンセプトを具体化し見える形で表現することで、認識のズレを明らかにし、修正点の把握・改善を繰り返す  
 →ハッカソン型コンセプト統合

Vision Forest & 段ボールプロトタイプング

## プロジェクトマネジメントの手法を活用して 「組み合わせ」で価値を生む R&D を

イノベーションの重要性が叫ばれる中、研究開発の世界もグローバル競争にさらされている。日本はそもそも「擦り合わせ」のアプローチを得意としてきたが、グローバルのデファクトスタンダードを握っているのは、Apple や Tesla に代表される、技術の「組み合わせ」で価値を生むプレイヤーたちだ。日本企業が競争力を上げていくためには、いかにコラボレーションを活発化し、トライ&エラーの質と効率を上げて、高い価値を創造し続けられるかが勝負となる。

ウォーターフォール型のプロジェクトマネジメントで培った経験とノウハウに、目標管理型のアプローチを組みあわせ、コラボレーション活性化のプログラムを並行で動かしていくこ

とで、日本の強みを生かした新しい R&D のあり方が確立できるのではないか、と筆者は考えている。

かつてのシステム構築では、第三者の PMO を使うという概念が全くない時代があった。しかし今や大型プロジェクトの多くは PM のプロフェッショナルを外部から入れている。近い将来、R&D の領域においても、同じようなことが起こる可能性は高い。そのためにも、私たち自身、多様な PM 経験の蓄積を目指すと同時に、できるだけ多くの企業にそのノウハウの移植を行い、日本企業の競争力向上を支援していきたいと考えている。

※この論文は ZDNet Japan (2015 年 12 月～2016 年 10 月)に掲載されたコラムの再掲載となります。